3 *Participation in a SOA Ecosystem* view

No man is an island

No man is an island entire of itself; every man
is a piece of the continent, a part of the main;
if a clod be washed away by the sea, Europe
is the less, as well as if a promontory were, as
well as any manner of thy friends or of thine
own were; any man's death diminishes me,
because I am involved in mankind.
And therefore never send to know for whom
the bell tolls; it tolls for thee.
John Donne

The OASIS SOA Reference Model defines Service Oriented Architecture as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" and services as "the mechanism by which needs and capabilities are brought together". The central focus of SOA is "the task or business function – getting something done."

Together, these ideas describe an environment in which business functions (realised in the form of services) address business needs. Service implementations utilize capabilities to produce specific (real world) effects that fulfill those business needs. Both ~~T~~those using the services, and the capabilities themselves, may be distributed across ownership domains, with different policies and conditions of use in force. The role of a service in the SOA context is to enable effective business solutions in a distributed environment. SOA is thus a paradigm that guides the identification, design, ~~and~~ implementation (i,e. organization), and utilization of such services.

The *Participation in a SOA Ecosystem* view in the SOA-RAF focuses on the constraints and context in which people[7] conduct business using a SOA-based system. By business we mean any shared activity entered into whose **objective** is to satisfy particular **needs** of each person. The OASIS SOA RM defines SOA as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains." To put it another way, to effectively employ the SOA paradigm, the architecture must take into account the fact and implications of different ownership domains, and how best to organize and utilize capabilities that are distributed across those different ownership domains. These are the main architectural issues that the Participating in a SOA Ecosystem view tries to address.

The subsections below expand on the completely abstract reference model by identifying more fully and with more specificity what challenges need to be addressed in order to successfully accomplish SOA. Although this section does not provide a specific

[7] 'People' and 'person' must be understood as both human actors and 'legal persons', such as companies, who have rights and responsibilities similar to 'natural persons' (humans)

recipe, it does identify the important things that need to be thought about and resolved within an ecosystem context.

The people actively participating in a SOA-based system, together with others who may potentially benefit from the services delivered by the system, together constitute the **stakeholders**. The stakeholders, the system and the environment (or context) within which they all operate, taken together forms the **SOA ecosystem**. That ecosystem may reflect the SOA-based activities within a particular enterprise or of a wider network of one or more enterprises and individuals.Although a SOA-based system is essentailly an IT concern, it is nonetheless a system engineered deliberately to be able to function in a SOA ecosystem. In this context, a service is the mechanism that brings a SOA-based system capability together with stakeholder needs in the wider ecosystem. This is explored in more detail in Section 3.2.2 below.

Furthermore, this *Participation in a SOA Ecosystem* view helps us understand the importance of execution context – the set of technical and business elements that allow interaction to occur in, and thus business to be conducted using, a SOA-based system.

This section describes how a SOA-based system behaves when participants may be in different organizations, with different rules and expectations, and assumes that the primary motivation for participants to interact with each other is to achieve **objectives** – to get things done.

The dominant mode of communication within a SOA ecosystem is electronic, supported by IT resources and artifacts. The stakeholders are nonetheless people: since there is inherent indirection involved when people and systems interact using electronic means, we lay the foundations for how *communication* can be used to represent and enable action. However, it is important to understand that these communications are usually a means to an end and not the primary interest of the participants of the ecosystem.

Several interdependent concerns are important in our view of a SOA-ecosystem. The ecosystem includes stakeholders who are participants in the development, deployment and governance and use of a system and its services; or who may not participate but are nonetheless are affected by the system. **Actors** – whether stakeholder **participants** or delegates who act only on behalf of participants (without themselves having any stake in the ecosystem) – are engaged in **actions** which have an impact on the real world and whose meaning and intent are determined by implied or agreed-to semantics.

The main models in this view are:

- the **Social Structure in a SOA Ecosystem Model** introduces the key elements that underlie the relationships between participants and that must be considered as pre-conditions in order to effectively bring needs and capabilities together across ownership boundaries;
- the **Action in a SOA Ecosystem Model** introduces the key concepts involved in service actions, and shows how ~~ownership, risk and transactions are key concepts in the SOA ecosystem~~joint action and real-world effect are what is being aimed for in a SOA ecosystem..

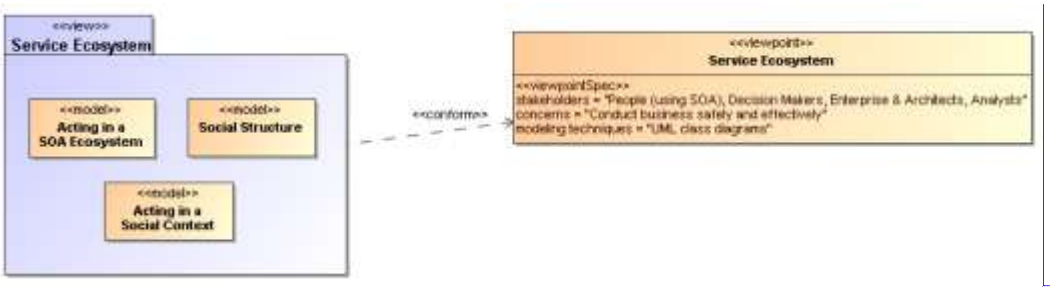


Figure 3 Model elements described in the Participation in a SOA Ecosystem *view*

3.1 Social Structure in a SOA Ecosystem Model

The actions undertaken by participants in a SOA ecosystem are performed in a *social context* that defines the relationships between the participants. That context is the social structure. In order to achieve success in SOA, the overall social structure in which the SOA effort is to be undertaken must be taken into consideration. ~~, as o~~Ownership boundaries and their implications can only be understood and addressed within the context of the larger social structure within which they exist and the nature of the relationships between the different participants in that structure.

The primary function of the Social Structure Model is to explain the relationships between an individual participant and the social context of that participant. The model also ~~underlines the importance of~~helps in defining and understanding the implications of crossing ownership boundaries. It is, for example, the foundation for understanding security, governance and management in the SOA ecosystem.

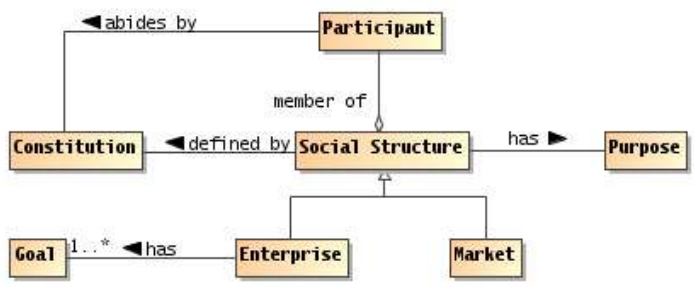


Figure 4 Social Structure

Social Structure

> A social structure[8] is a nexus of relationships amongst participants brought together for a specific purpose. (Social structures are sometimes referred to as social institutions.)

A social structure represents a collection of participants, but a collection that is brought together for a purpose. There may be a large number of different kinds of relationships

[8] Social structures are sometimes referred to as social institutions.

Comment [c6]: Update figure to only have the two models above

Comment [PFB7]: Update model and view names

Comment [PFB8]: Diagram needs reworking.
Remove 'enterprise' and 'market' and have an association between Social Structure and Goal; and another, 'Constitution defines Goal' (1..*)

between participants in a social structure. The organizing principle for these relationships is the social structure's purpose.

A social structure may have any number of participants, and a given participant can be a member of multiple social structures. Thus, there may be interaction among social structures, sometimes resulting in disagreements when the premises of the social structures do not align.

A social structure has a purpose – the overarching reason for which it exists. All social structures are established with implied or explicitly defined purpose. The purpose is usually reflected in specific goals laid down in the social structure's constitution or other 'charter'.

A social structure can take different forms. For example, an enterprise is a common kind of social structure that embodies a form of hierarchic organization; an online chat room represents a social structure of peers that is very loose. A market represents a social structure of buyers and sellers. The legal frameworks of entire countries and regions also count as social structures.

The RAF is concerned primarily with social structures that reflect relationships amongst **participants** in SOA ecosystems, notably:

- the enterprise social structure which is composed internally of many participants but that has sufficient cohesiveness to be considered as a potential stakeholder in its own right; and
- the peer group which governs relationship between participants within an ecosystem..

Enterprise

> An enterprise is a social structure with an identifiable ~~head~~leadership structure, and that has internally established goals that reflect a defined purpose. It can act as a participant within other social structures, including other enterprises and is represented by members of its leadership structure.

Peer Group

> A peer group is a social structure with~~out an identifiable head~~no discernable leadership structure, that may or may not have internally established goals, but is identiable as the locus of interaction between participants with individual goals and who are considered peers of one another.

Many interactions between participants take place within ~~an enterprise~~ social structures. Depending on the scale and internal structure of an~~the~~ enterprise social structure, these interactions may or may not~~can~~ cross ownership boundaries (an enterprise can itself be composed of sub-enterprises). However, interactions between participants within a peer social structure ~~are~~ inherently ~~a~~cross ownership boundaries.

The nature and extent of the interactions that take place will reflect, often implicitly, degrees of trust between participants and the very specific circumstances of each participant at the time, and over the course, of the interactions. It is in the nature of an SOA ecosystem that these relationships are rendered more explicit and are formalized and form a central part of what the SOA-RM refers to as "Execution Context".

Social structures involved in a particular interaction are not always explicitly identified. For example, when a customer buys a book over the Internet, the social structure that

determines the validity of the transaction is often the legal framework of the region associated with the book vendor. Such legal jurisdiction qualification is typically buried in the fine print of the service description.

Constitution

> A constitution is a set of rules, written or unwritten, that spell out the purpose, goals, scope, and functioning of a social structure.

Every social structure functions according to rules by which participants interact with each other within the structure. In some cases, this is based on an explicit agreement, in other cases participants behave as though they agree to the constitution without a formal agreement. In still other cases, participants abide by the rules with some degree of reluctance – this is an issue raised later on when we discuss governance in SOA-based systems. In all cases, the constitution may change over time, in those cases of implicit agreement the change can occur quickly.

3.1.1 Participants, Actors and Delegates

Social structures have stakeholders, some of whom may be enterprises. They interact within the broad ecosystem. Actors operate within a system. The concept of Participant is particularly important as it reflects the hybrid role of both a Stakeholder (in the ecosystem), primarily concerned with expressing needs and seeing those needs fulfilled; and an Actor (in the System), directly involved with system-level activity. This hybrid role of Participant thus provides a bridge between the ecosystem and the system.

An actor can be either a **participant** (and thus also a stakeholder) – with a stake in the ecosystem; or a **delegate** (a human actor with no stake in the ecosystem or an automated agent), acting on behalf of a participant.

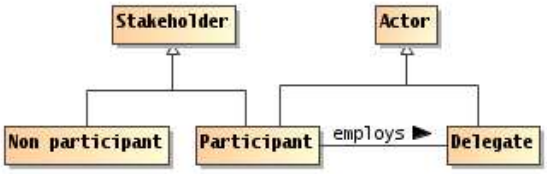


Figure 5 Actors, Participants and Delegates

Stakeholder

> A stakeholder in the SOA ecosystem is a person with an interest – a 'stake' – in the ecosystem.

Note: Not all stakeholders necessarily participate in the SOA ecosystem; indeed, the interest of non-participant stakeholders may be in realizing the benefits of a well-functioning ecosystem and not suffering unwanted consequences. They can not all or always be identified in advance but due account is often taken of such stakeholder types, including potential customers, beneficiaries, affected third parties, as well as potential "negative stakeholders" who might deliberately seek a negative impact on the ecosystem (such as hackers or criminals).

Actor

An actor is a participant or delegate capable of action within a SOA-based system.

Participant

A participant is a person[9] who is both a stakeholder in the SOA ecosystem and an actor in the SOA-based system.

Delegate

A delegate is an actor that is acting on behalf of a participant.

A delegate can be a person or an automated or semi-automated agent.

Many stakeholders and actors operate in a SOA ecosystem, including software agents that permit people to offer, and interact with, services; delegates that represent the interests of other participants; or security agents charged with managing the security of the ecosystem. Note that automated agents are always delegates, in that they act on behalf of a stakeholder.

In the different models of the RAF, actor is used when it is not important whether the entity is a delegate or a participant. If the actor is acting on behalf of a stakeholder, then we use delegate. This underlines the importance of ~~automation~~ delegation in SOA-based systems, whether the ~~automation~~ delegation is of work procedures carried out by human agents who have no stake in the ecosystem but act on behalf of a participant who does; or whether the ~~automation~~ delegation is performed by technology (automation). If the actor is also a stakeholder in the ecosystem, then we use participant.

In order for a delegate to act on behalf of another person, they must be able to act and have the authority to do so.

3.1.2 Roles in Social Structures

Social structures are abstractions: a social structure cannot directly perform actions – only people or automated processes following the instructions of people can actually do things. However, an actor may act on behalf of a social structure and certainly acts within a social structure depending on the roles that the actor assumes and the nature of the relationships betweent the concerned parties or stakeholders.

Comment [PFB9]: There is a modeling and inheritance error in the original wording – it would imply that a participant (as an inherited sub-class) can be an automated agent.

[9] Again, this can be a 'natural' or 'legal' person

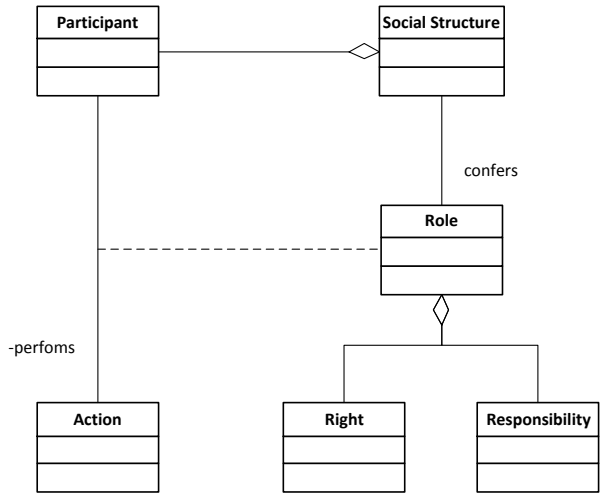


Figure 6 Role in Social Structures

Role

A role is a type of relationship between a participant and the actions that participant may performs (or is allowed to perform)~~ed~~ within a social structure.

A role is not immutable and is often time-bound. A participant can have one or more roles concurrently and may change them over time and in different contexts, even over the course of a particular interaction. One participant with appropriate authority in the social structure may formally *designate a role* for another participant, with associated rights and responsibilities, and that authority may even qualify a period during which the designated~~is~~ role may be valid.

Conversely, someone who exhibits qualification and skill may *assume a* role without any formal designation. For example, an office administrator who has demonstrated facility with personal computers may be known as (and thus assumed to role of) the 'goto' person for people who need help with their computers.

Although many roles are clearly identified, with appropriate names and definitions of responsibilities, it is also entirely possible to separately bestow rights, bestow or assume responsibilities and so on, often in a temporary fashion. For example, when a company president delegates certain responsibilities on another person, this does not imply that the other person has become company president. Likewise, a company president may bestow on someone else her role during a period of time that she is on vacation or otherwise unreachable, with the understanding that she will re-assume the role when she returns from vacation.

Authority

Authority is the right or responsibility to act on behalf of an organization or another person.

Right

A right is a predetermined permission conferred upon an actor that allows them to perform some action or assume a role in relation to the social structure.

Rights can be constrained. For example, sellers might have a general right to refuse service to potential customers but this right could be constrained so as to be exercised only when ~~based on~~ certain criteria are met.

Responsibility

> A responsibility is a predetermined obligation on a participant to perform some action or to adopt a stance or role in relation to other actors.

Responsibility implies human agency, which is why only participants, as opposed to all actors (who can be non-human agents) are concerned. even if the consequences of such responsibility can impact other (human and non-human) actors.

Rights, authorities, responsibilities and roles form the foundation for the security model as well as contributing to the governance model in the 'Ownership in a SOA Ecosystem' View of the RAF. Rights and responsibilities are similar in structure to permissions and obligations; except that rights and responsibilities are associated with participants as opposed to permissions and obligations which are associated with actions.

People will assume and perform roles according to their actual or perceived rights and responsibilities, with or without explicit authority. In the context of a SOA ecosystem, human abilities and skills are relevant as they equip individuals with knowledge, information and tools that may be necessary to have meaningful and productive interactions with a view to achieving a desired outcome. For example, a person who needs a particular book, and has both the right and responsibility of purchasing the book from a given bookseller, will not have that need met from the online delegate of that bookstore if he does not know how to use a web browser. Equally, just because someone does have the requisite knowledge or skills does not entitle them *per se* to interact with a specific system.

3.1.2.1 Service Roles

As in roles generically, ~~It is inherent to the SOA paradigm that~~ a participant can play one or more of those roles inherent to the SOA paradigm in the SOA ecosystem, including as a service consumer, a service provider, a mediator, and so on, depending on the context. A participant may be playing a role of a service provider in one relationship while simultaneously playing the role of a consumer in another. Roles inherent to the SOA paradigm include Consumer, Provider, and Mediator.

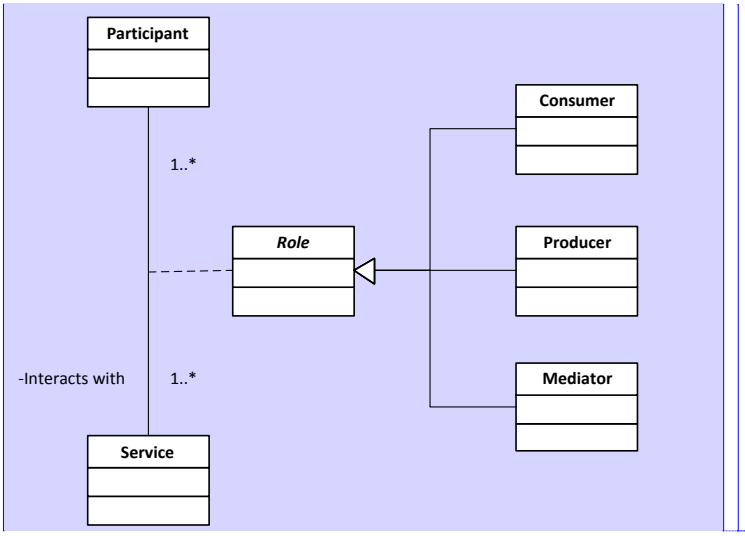


Figure 7 Participant Roles in a Service

Provider

A provider is a role assumed by a participant who is offering a service.

Consumer

A consumer is a role assumed by a participant who is interacting with a service in order to fulfill a need.

Mediator

A mediator is a role assumed by a participant to facilitate interaction and connectivity in the offering and use of services.

It is a common understanding that service interactions are typically initiated by service consumers, although this is not necessarily true in all situations. Additionally, as with service providers, several stakeholders may be involved in a service interaction supporting ~~the~~ a given consumer.

The roles of service provider and service consumer are often seen as symmetrical, which is also not entirely correct. A consumer tends to express a 'Need' in non-formal terms: "I want to buy that book". The type of 'Need' that a service is intended to fulfill has to be formalized and encapsulated by designers and developers as a 'Requirement'. This Requirement should then be reflected in the target service, as a 'Capability'. The Capability, when accessed via a service, delivers a 'Real World Effect' that *brings to bear* the capability and satisfies a specific occurrence of the need that has been defined. This is explored in more detail later on. ~~in section 3.2.1~~Service mediation by a participant can take many forms and may invoke and use other services in order to fulfill such mediation. For example, it might use a service registry in order to identify possible service partners; or, in our book-buying example, it might provide a price comparison service, suggest alternative suppliers, different language editions or delivery options.

Comment [c10]: Replace Producer with Provider

Comment [c11]: Do we need to add the role of "owner"?
[Peter]: Indeed possible, does it help? Maybe…

Comment [PFB12]: Can we call it an "archetype" or is this too confusing?

Comment [PFB13]: I think it is an important distinction between the service 'intent' (based on a supposed need 'archetype') and the actual, instantiated use, with very different and diverse customers…

3.1.3 Resource and Ownership

3.1.3.1 Resource

A resource is generally understood as an asset: it has value to someone. Key to this concept in a SOA ecosystem is that a resource needs to be identifiable.

Resource

A resource is any identifiable entity that has value to a stakeholder.

A resource may be identifiable by different methods but within a SOA ecosystem a resource must have at least one well-formed identifier that may be unambiguously resolved to the intended resource.

Codified (but not *implied*) contracts, policies, obligations, and permissions are all examples of resources as are capabilities , services, service descriptions, and SOA-based systems. ~~A service can be perceived also as a resource as it delivers a particular result to a consumer – the consumer sees it as a resource.~~ An *implied* policy, contract, obligation or permission would not be a resource, even though it may have value to a stakeholder, because it is not an identifiable entity.

Comment [c14]: Are non-automated capabilities (i.e., those that are performed by humans) considered resources?
[Peter]: Yes, I'd say so..

Identifier

An identifier is any sequence of characters that may be unambiguously resolved to identifying a particular resource.

Comment [c15]: This troubles me a bit. Can an identifier be something other than a sequence of characters. E.g., a picture, a graph, a string, a line of breadcrumbs? The "sequence of characters" just "feels" a bit to concrete and limited
[Peter]: Hmmm, if it is to be "unambiguously resolved to identifying a resource" can it be anything other than a stream of bits, however it may be seen by the user?

Identifiers typically require a context in order to establish the connection with the resource. In a SOA ecosystem, it is good practice to use globally unique identifiers; for example globally unique IRIs.

A given resource may have multiple identifiers, with different value for different contexts.

The ability to identify a resource is important in interactions to determine such things as rights and authorizations, to understand what functions are being performed and what the results mean, and to ensure repeatability or characterize differences with future interactions. The specific subset of individual characteristics that are necessary and sufficient in order to unambiguously identify a resource depends on the ecosystem and/or specific interactions within a system. However, in order to enable visibility and interaction in a SOA ecosystem, those resources that are important to a given SOA system must be *unambiguously* identifiable at any moment and in any interaction, many of which may not be predictable given the operation of systems across ownership boundaries. The way to achieve this is by using identifiers.

3.1.3.2 Ownership

Ownership is defined as a relationship between a stakeholder and a resource, where some ~~person~~ stakeholder (in a role as **owner**) has certain claims with respect to the resource.

Typically, the ownership relationship is one of control: the owner of a **resource** can control some aspect of the resource.

Ownership

> Ownership is a particular set of claims, expressed as rights and responsibilities, that a stakeholder has in relation to a resource; It may include the right to transfer that ownership, or some subset of rights and responsibilities, to another entity.

To own a resource implies taking responsibility for creating, maintaining and, if it is to be available to others, provisioning the resource. More than one stakeholder may own different rights or responsibilities associated with a given ~~resource~~service, such as one stakeholder having the ~~right~~ responsibility to deploy a capability as a service, another owning the rights to the profits that result from charging consumers for using the service~~capability~~, and yet another owning the right~~s~~ to use the service.

A stakeholder who owns a resource may delegate some or all of these rights and responsibilities to others, but typically retains the responsibility to see that the delegated rights and responsibilities are exercised as intended. There may also be joint ownership of a resource, where the rights and responsibilities~~y~~ are~~is~~ shared.

A crucial property that distinguishes ownership from a more limited ***right to use*** is the right to transfer rights and responsibilities totally and irrevocably to another ~~person~~stakeholder. When a stakeholder uses a resource but does not own the resource, that stakeholder may not transfer the right to use the resource to a third stakeholder. The owner of the resource maintains the rights and responsibilities of being able to authorize other stakeholders to use the owned resource.

Ownership is defined in relation to the social structure relative to which the given rights and responsibilities are exercised. In particular, there may be constraints on how ownership may be transferred. For example, a government may not permit a corporation to transfer assets to a subsidiary in a different jurisdiction.

Ownership Boundary

> An ownership boundary is the extent of ownership asserted by a stakeholder over a set of resources and for which rights and responsibilities are claimed and (usually) recognized by other stakeholders.

In a SOA ecosystem, providers and consumers of services may be, or may be acting on behalf of, different owners, and thus the interaction between the provider and the consumer of a given service will necessarily cross an ownership boundary. It is important to identify these ownership boundaries in a SOA ecosystem, as successfully crossing them requires the elements identified in the following sections be addressed. Addressing the elements identified in the following sections is referred to in the OASIS SOA RM as establishing the execution context.

3.1.4 Assessing Trust and Risk

The assessments of trust and risk are based on evidence available to the *trusting* participant. In general, participants will seek evidence directly from the *trusted* actor (e.g., via documentation provided via the service description) as well as evidence of the reputation of the trusted actor (e.g., third-party annotations such as consumer feedback).

Comment [PFB16]: Moved this from the end of the section 3.1 – fits better here.

Comment [PFB17]: Should this be participant or actor? If we need to include human delegates, should we more precisely state 'human actor' to include this?

Trust is based on the confidence that the trusting participant has accurately and sufficiently gathered and assessed evidence to the degree appropriate for the situation being assessed.

Assessment of trust is rarely binary. An actor is not completely trusted or untrusted. There is typically some degree of uncertainty in the accuracy or completeness of the evidence or the assessment. Similarly, there may be uncertainty in the amount and potential consequences of risk.

The relevance of trust to interaction depends on the assessment of risk. If there is little or no perceived risk, then the degree of trust may not be relevant in assessing possible actions. For example, most people consider there to be an acceptable level of risk to privacy when using search engines, and submit queries without any sense of trust being considered.

As perceived risk increases, the issue of trust becomes more of a consideration. For interactions with a high degree of risk, the trusting participant will typically require stronger or additional evidence when evaluating the balance between risk and trust. An example of high-risk is where a consumer's business is dependent on the provider's service meeting certain availability and security requirements. If the service fails to meet those requirements, the service consumer will go out of business. In this example, the consumer will look for evidence that the likelihood of the service not meeting the performance and security requirements is extremely low.

~~3.1.4~~3.1.5 Policies and Contracts

As noted in the Reference Model, a **policy** represents some commitment and/or constraint promulgated and enforced by a stakeholder and that stakeholder alone. A **contract**, on the other hand, represents an agreement by two or more participants. Enforcement of contracts may or may not be the responsibility of the parties to the agreement but is usually performed by a stakeholder in the ecosystem (public authority, legal system, etc.).

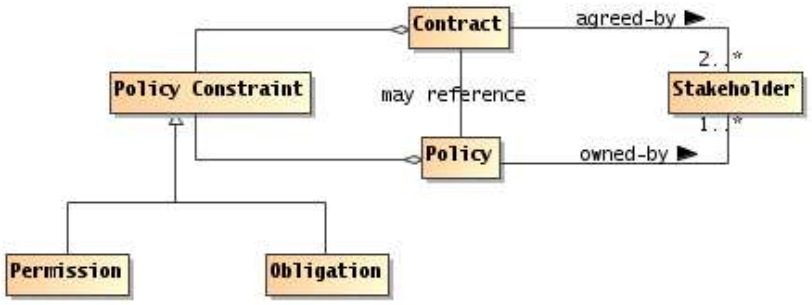


Figure 8 Policies and Contracts

Policy

> A policy is an assertion made by a stakeholder which the stakeholder commits to uphold and, if possible and necessary, enforce through stated constraints.

Policies can often be said to be about something – they have an object. For example, there may be policies about the use of a service. Policies have an **owner** – the

stakeholder who asserts and takes responsibility for the policy. Note that the policy owner may or may not be the owner of the object of the policy. Thirdly, policies represent constraints – some measurable limitation on the state or behavior of the object of the policy, or of the behavior of the stakeholders of the policy.

Contract

> A contract represents an agreement made by two or more participants (the contracting parties) on a set of promises (or contractual terms) together with a set of constraints that govern their behavior and/or state in fulfilling those promises.

A service provider's policy may become a service provider/consumer contract when a service consumer agrees to the provider's policy. That agreement may be formal, or may be informal. If a consumer's policy and a providers policy are mutually exclusive, then some form of negotiation or mediation to resolve the mutual exclusion before the service consumer/provider interaction can occur.

Both policies and contracts imply a desire to see constraints respected and enforced. Policies are owned by individual (or aggregate) stakeholders, and contracts are owned by the parties to the contract; these stakeholders are responsible for ensuring that any constraints in the policy or contract are enforced – although, of course, the actual enforcement may be delegated to a different mechanism. A contract does not necessarily oblige the contracting parties to act (for example to use a service) but it does constraint how they act if and when action covered by the contract occurs (for example, when a service is invoked and used).

Two important types of constraint that are relevant to a SOA ecosystem are permission and Obligation.

Permission

> A permission is a constraint that identifies **actions** that an actor is (or is not) allowed to perform and/or the **states** the actor is (or is not) permitted to be in.

Note that permissions are distinct from ability and from authority. Authority refers to the legitimate nature of an action as performed by an actor on behalf of a social structure and ability refers to whether an actor has the capacity to perform the action, whereas permission does not always involve acting on behalf of anyone, nor does it imply or require the capacity to perform the action.

Obligation

> An obligation is a constraint that prescribes the actions that an actor must (or must not) perform and/or the states the actor must (or must not) be in.

An example of obligations is the case where the service consumer and provider have entered into an agreement to provide and consume a service such that the consumer is obligated to pay for the service and the provider is obligated to provide the service – based on the terms of the contract.

An obligation can also be a requirement to to *maintain* a given state. This may range from a requirement to maintain a minimum balance on an account to a requirement that a service provider 'remember' that a particular service consumer is logged in.

Both permissions and obligations can be identified ahead of time, but only Permissions can be validated a priori: before the intended action or before entering the constrained state. Obligations can only be validated a posteriori through some form of auditing or verification process.

3.1.6 Communication

Communication

> A communication is a process of reaching mutual understanding, in which participants not only exchange information as messages but also create and share meaning..

A communication involves one or more actors playing the role of **sender** and at least one other actor playing the role of **recipient**; all actors must perform their part in order for the communication to occur.

A given communication may involve any number of **recipients**. In some situations, the sender may not be aware of the recipient. However, without both a sender and a recipient there is no communication. A given communication does not necessarily involve interaction between the actors; it can be a simple one-way transmission requiring no further action by the recipient. However, interaction does, necessarily, involve communication.

A communication involves a message, which an actor receiving must be able to correctly interpret. The extent of that correct interpretation depends on the role of the actor and the purpose of the communication.

A communication is not effective unless the recipient can correctly interpret the message. However, interpretation can itself be characterized in terms of semantic engagement: the proper understanding of a message in a given context.

We can characterize the necessary modes of interpretation in terms of a shared understanding of a common vocabulary and of the purpose of the communication. More formally, we can say that a communication has a combination of message and purpose.

Interactions between service consumers and providers do not need to resemble human speech. Machine-machine communication is typically highly stylized in form, it may have particular forms and it may involve particular terms not found in everyday human ~~interaction~~communication.

~~3.1.5~~3.1.7 Semantics and Semantic Engagement

A SOA ecosystem is a space in which actors need to share understanding[10] as well as sharing actions. Indeed, such shared understanding is a pre-requisite to a joint action being carried out as intended. It is vital to a trusted and effective ecosystem. Semantics are therefore pervasive throughout SOA ecosystems and important in communicative actions described above, as well as a driver for policies and other aspects of the ecosystem.

[10] We use a mechanical, Turing test-based approach to understanding here: if an actor behaves as though it understands an utterance then we assume that it does understand it.

In order to arrive at shared understanding, an actor must effectively process and understand assertions in a manner appropriate to the particular context. An assertion, in general, is a measurable and explicit statement made by an actor. In a SOA ecosystem, in particular, assertions are concerned with the 'what' and the 'why' of the state of the ecosystem and its actors.

Understanding and interpreting those assertions allows other actors to know what may be expected of them in any particular joint action. An actor can potentially 'understand' an assertion in a number of ways, but it is specifically the process of arriving at a *shared* understanding that is important in the ecosystem. This process is semantic engagement by the actor with the SOA ecosystem. It can be instantaneous or progressively achieved. It is important that there is a level of engagement appropriate to the particular context.

Semantic Engagement

> Semantic engagement is the process by which an actor engages with a set of assertions based on that actor's interpretation and understanding of those assertions.

Different actors have differing capabilities and requirements for understanding assertions. This is true for both human and non-human actors. For example, a purchase order process does not require that a message forwarding agent 'understand~~s~~' the purchase order, but a processing agent does need to '~~know'~~ understand' the purchase order in order to know what to with the order once received.

The impact of any assertion can only be fully understood in terms of specific social contexts; contexts that necessarily include the actors that are involved. For example, a policy statement that governs the actions relating to a particular resource may have a different impact or purpose for the participant that owns the resource than for the actor that is trying to access it: the former understands the purpose of the policy as a statement of enforcement; and the latter understands it as a statement of constraint.

~~3.1.6~~3.1.8 Trust and Risk

For an interaction to occur each actor must be able and **willing** to participate.

Willingness

> Willingness is the internal commitment of a human actor to carry out its part of an interaction.

Willingness to interact is not the same as a willingness to perform requested actions, however. For example, a service provider that rejects all attempts to perform a particular action may still be fully willing and engaged in interacting with the consumer. Important considerations in establishing willingness are both **trust** and **risk**.

Trust

> Trust is a private assessment or internal perception of one participant that another participant will perform actions in accordance with an assertion regarding a desired real world effect.

Risk

> Risk is a private assessment or internal perception ~~that~~ of the likelihood that certain undesirable real world effects ~~may~~ will result from actions taken, or that the RWE might not meet certain criteria (e.g., performance), and the consequences or implications of such.

Trust is involved in all interactions – it is necessary for *all* the actors (consumers, providers, mediators) involved in a given interaction to trust each other at least to the extent required for continuance of the interaction. The degree and nature of that trust is likely to be different for each actor, most especially when those actors are in different ownership boundaries.

An actor perceiving risk may take actions to mitigate that risk. At one extreme this will result in a refusal to interact. Alternately, it may involve adding protection – for example by using encrypted communication and/or anonymization – to reduce the perception of risk. Often, standard procedures are put in place to increase trust and to mitigate risk.

Comment [PFB18]: Not true of an orchestration/choreography. In the orch/cho, only the actors in a particular joint action need to trust each other, there does not have to be a transitive trust among all actors in that case.

With reference now to 'interaction' and not service activity, is this issue resolved?

3.2 Action in a SOA Ecosystem Model

Participants cannot always achieve desired results leveraging resources in their own ownership domain; thus generating a need for which they look for and leverage services provided by other participants, using resources beyond their ownership and control; They identify service providers with which they think they can interact to achieve their objective; They thus engage in joint action with those other actors (service providers) in order to bring about the desired outcome; the SOA ecosystem provides the environment to make this happen.

An action model is put forth a-priori by the service provider, and is effectively a promise by the service provider that the actions identified in the action model and invoked consistent with the process model will result in the described real world effect. Action model is basically a description of the actions that the service is willing to do on behalf of another. They should be associated with a real-world effect. The potential service consumer is interested in accessing or acquiring the real-world effect, and the action model identifies the actions that the service consumer will have to be a party to in order to access or generate the real-world effect.

When the consumer "invokes" a service, a joint action is started as identified in the action model, consistent with the temporal sequence as defined by the process model, and where the consumer and the provider are the two parties of the joint action. Additionally, the consumer can be assured that the identified real-world effects will be accomplished through evidence provided via the service description.

Since the service provider does not know about all potential service consumers, the service provider may also describe what additional constraints are necessary in order for the service consumer to invoke particular actions, and thus participate in the joint action. These additional constraints, along with others that might not be listed, are preconditions for the joint action to occur and/or continue (as per the process model), and are referred to in the SOA RM as execution context. Execution context goes all the way from human beings involved in aligning policies, semantics, network connectivity and communication protocols, to the automated negotiation of security protocols and end-points as the individual actions proceed through the process model.

Also, it is important to note that both actions and RWE are 'fractal' in nature, in the sense that they can often be broken down into more and more granularity depending on how they are examined and what level of detail is important.

All of these things are important to getting to the core of participants' interest in a SOA ecosystem: the ability to leverage resources or capabilities to achieve a desired outcome, and in particular where those resources or capabilities do not belong to them or are beyond their direct control. i.e., that are outside of their ownership boundary.

In order to use such resources, participants must be able to identify their own needs in the form of requirements, identify and compose into a business solution those resources or capabilities that will meet their needs, and engage in **joint action** – the coordinated set of actions that participants pursue in order to achieve measurable results in furtherance of their goals.

In order to act, in a way that is appropriate and consistent both to their own goals, objectives and policies, and those of others, participants must also communicate with each other.

A key aspect of joint action revolves around the **trust** that both parties must exhibit in order to participate in the joint action. The willingness to act and a mutual understanding of both the information exchanged and the expected results is the particular focus of Sections 3.1.86 and 3.1.7.

3.2.1 Needs, Requirements and Capabilities

Participants in a SOA ecosystem often need other participants to *do* something, leveraging a capability that they do not themselves possess. For example, a customer requiring a book may call upon a service provider to deliver the book. Likewise, the service provider needs the customer to pay for it.

There is a reason that participants are engaged in this activity: different participants have different **needs** and have or apply different **capabilities** for satisfying them.These are core to the concept of a service. The SOA-RM defines a service as "the mechanism by which needs and capabilities are brought together". This idea of services being a mechanism "between" needs and capabilities was introduced in order to emphasize capability as the notional or existing business functionality that would address a well-defined need. Service is therefore the *implementation* of such business functionality *such that it is accessible* through a well-defined interface. A capability that is isolated, or by itself (i.e., not accessible to potential consumers) is emphatically not a service.

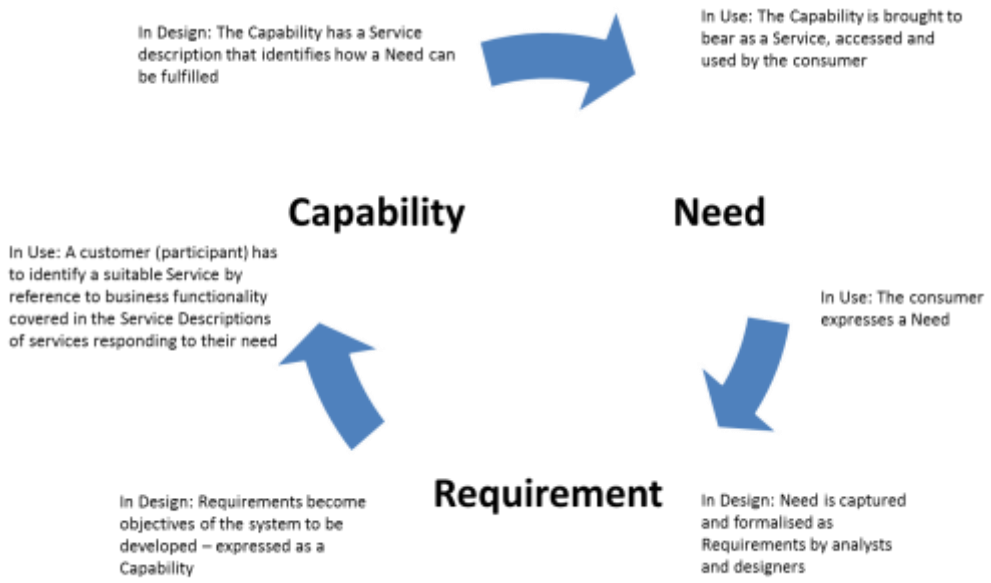
Business functionality

> Business functionality is a defined set of business-aligned tasks that provide recognizable business value to ~~stakeholder~~ 'consumer' stakeholders and possibly others in the SOA ecosystem.

Comment [PFB19]: This is added to cover also delegate

Comment [PFB20]: Moved earlier in this section (was originally starting after sub-section on Action & Joint Action), together with the new (sketched) diagram proposed of Need > Requirement > Capability

The idea of a service in a SOA ecosystem combines business functionality with implementation, including the artifacts needed and made available as IT resources. From the perspective of software developers, a SOA service enables the use of capabilities in an IT context. For the consumer, the service (combining business functionality and implementation) ~~produces~~ generates intended real world effects. The consumer is not concerned with the underlying artifacts which make that delivery



possible.

Figure 9 - Relationship between Need, Requirement and Capability

In a SOA context, the consumer (as a stakeholder) expresses a need ("I want to buy a book") and looks to an appropriate service to fulfill that need and assesses issues such as the trustworthiness, intent and willingness of a particular provider. This ecosystem communication continues up to the point when the consumer is ready to act. The consumer (as an actor now) will then interact with a provider by invoking a service (for example, ordering the book using an online bookseller) and engaging in relevant actions (validating the purchase, submitting billing and delivery details) within the system ~~– the joint action –~~ with a view to achieving the desired Real World Effect (having the book delivered).

Need

A need is a general statement expressed by a stakeholder and may be formalized as one or more **requirements** that must be fulfilled in order to achieve a stated goal.

Requirement

A requirement is a formal statement of a desired result (a real world effect) that, if achieved, will satisfy a need.

This requirement can then be used to create a capability that in turn can be brought to bear to satisfy that need. Both the requirement and the capability to fulfill it are expressed in terms of desired real world effect.

Capability

A capability is an ability to achieve a real world effect.

The Reference Model makes a distinction between a capability (as a potential to generate a real world effect) and the ability of bringing that capability to bear (via a realized service) as the realization of the real world effect.

3.2.2 Services Reflecting Business

The SOA paradigm often emphasizes the prescribed interface through which service interaction is accomplished. While this enables predictable integration in the sense of traditional software development, the prescribed interface alone does not guarantee that services will be composable into business solutions.

Business solution

A **business solution** is a set of defined interactions that combine implemented or notional business functionality in order to address a set of business needs.

Composability

Composability is the ability to combine individual services, each providing defined business functionality, so as to provide more complex business solutions.

Composability is important because many of the benefits of a SOA approach assume multiple uses for services, and multiple use requires that the service deliver a business function that is reusable in multiple business solutions.

To achieve composability, capabilities must be identified that serve as building blocks for business solutions. In a SOA ecosystem, these building blocks are captured as services representing well-defined business functions, operating under well-defined policies and other constraints, and generating well-defined real world effects. These service building blocks should be relatively stable so as not to force repeated changes in the compositions that utilize them, but should also embody SOA attributes that readily support creating compositions that can be varied to reflect changing circumstances.

The SOA paradigm emphasizes both composition of services and opacity of how a given service is implemented. With respect to opacity, the SOA-RM states that the service could carry out its described functionality through one or more automated and/or manual processes that in turn could invoke other available services.

Any composition can itself be made available as a service and the details of the business functionality, conditions of use, and effects are among the information documented in its service description.

For services to be useful as composable building blocks in the SOA ecosystem, the services should, whenever possible, deliver capability that is applicable to multiple needs. Simply providing a Web Service interface for an existing IT artifact does not, in

general, create opportunities for sharing business functions. Furthermore, the use of tools to auto-generate service software interfaces will not guarantee services than can effectively be used within compositions if the underlying code represents programming constructs rather than business functions. In such cases, services that tightly reflect the software details will be as brittle to change as the underlying code and will not exhibit the ~~ill~~un-defined but intuitive characteristic of loose coupling.

3.2.3 Action, Communication and Joint Action

In general terms, entities act in order to achieve their goals. However, the form of action that is of most interest within a SOA ecosystem is that involving interaction across ownership boundaries (between more than one actor) – **joint action.**

Comment [PFB21]: Reword

3.2.3.1 Action and Actors

Action

An action is the application of intent to cause an effect.

The aspect of action that distinguishes it from mere force or accident is that someone *intends* that the action achieves a desired objective or effect. This definition of action is very general. In the case of SOA, we are mostly concerned with actions that take place within a system and have specific effects on the SOA ecosystem – what we call **Real World Effects**. The actual real world effect of an action, however, may go beyond the intended effect.

Objectives refer to real world effects that participants believe are achievable by a specific action or set of actions that deliver appropriate changes in shared state. In contrast, a goal is not expressed in terms of specific action but rather in terms of desired end state.

For example, someone may wish to have enough light to read a book. In order to satisfy that goal, the reader walks over to flip a light switch. The *objective* is to change the state of the light bulb, by turning on the lamp, whereas the *goal* is to be able to read. The *real world effect* is more light being available to enable the person to read.

While an effect is any measurable change resulting from an action, a SOA ecosystem is concerned more specifically with real world effects.

Real World Effect

A real world effect is a measurable change to the shared state of pertinent entities, relevant to and experienced by specific stakeholders of an ecosystem.

This implies measurable change in the overall state of the SOA ecosystem. In practice, however, it is specific state changes of certain entities that are relevant to particular participants that constitute the real world effect as experienced by those participants.

3.2.3.2 Communication and Joint Actions

In this Reference Architecture Foundation, we are concerned with two levels of activity: as communication and as participants engaged in joint actions to use and offer services.

In order for multiple actors to participate in a joint action, they must each act according to their role within the joint action. This is achieved through communication and messaging.

Communication – the formulation, transmission, receipt and interpretation of messages – is the foundation of all joint actions within the SOA ecosystem, given the inherent separation – often across ownership boundaries – of actors in the system.

Communication between actors requires that they play the roles of 'sender' or 'receiver' of messages as appropriate to a particular action – although it is not necessarily required that they both be active simultaneously.

An actor sends a message in order to communicate with other actors. The communication itself is often not intended as part of the desired real world effect but rather includes messages that seek to establish, manage, monitor, report on, and guide the joint action throughout its execution.

Like communication, joint action usually involves different actors. However, joint action – resulting from the deliberate actions undertaken by different actors – *intentionally* impacts shared state within the system leading to real world effects.

Joint Action

Joint action is the coordinated set of actions involving the efforts of two or more actors to achieve an effect.

Note that the effect of a joint action is *not* always equivalent to one or more effects of the individual actions of the participating actors, i.e., it may be more than the sum of the parts.

Different viewpoints lead to either communication or joint action as being considered most important. For example, from the viewpoint of ecosystem ~~governance~~security, the integrity of the communications may be dominant; from the viewpoint of ecosystem ~~security~~governance, the integrity of the joint action may be dominant.

3.2.4 State, Shared State and Real-World Effect

State

State is the condition of an entity at a particular time.

Comment [PFB22]: Entity can be an actor, a resource, a light bulb..?

State is characterized by a set of facts that is true of the entity. In principle, the total state of an entity (or the world as a whole) is unbounded. In practice, we are concerned only with a subset of the State of an entity that is measurable and useful in a given context.

For example, the total state of a lightbulb includes the temperature of the filament of the bulb. It also includes a great deal of other state – the composition of the glass, the dirt that is on the bulb's surface and so on. However, an actor may be primarily interested in whether the bulb is 'on' or 'off' and not on the amount of dirt accumulated. That actor's characterization of the state of the bulb reduces to the fact: 'bulb is now on'.

In a SOA ecosystem, there is a distinction between the set of facts about an entity that only that entity can access – the so-called Private State – and the set of facts that may be accessible to other actors in the SOA-based system – the public or Shared State.

Private State

The private state is that part of of an entity's state that is knowable by, and accessible to, only that entity.

Shared State

Shared state is that part of an entity's state that is knowable by, and may be accessible to, other actors.

Note that shared state does not imply that the state *is* accessible to *all* actors. It simply refers to that subset of state that *may* be accessed by *other* actors. Generally this will be the case when actors need to participate in joint actions.

It is the aggregation of the shared states of pertinent entities that constitutes the desired effect of a joint action. Thus the change to this shared state is what is experienced in the wider ecosystem as a real world effect

3.3 Architectural Implications

3.3.1 Social structures

A SOA ecosystem's participants are organized into various forms of social structure. Not all social structures are hierarchical: a SOA ecosystem should be able to incorporate peer-to-peer forms of organization as well as hierarchic structures. In addition, it should be possible to identify and manage any constitutional agreements that define the social structures present in a SOA ecosystem.

- Different social structures have different rules of engagement
  - Techniques for expressing constitutions are important
- social structures have roles and members
  - Techniques for identifying, managing members of social structures
  - Techniques for describing roles and role adoption
- social structures may be complex
  - Child social structures' constitutions depend on their parent constitutions
- Social structures overlap and interact
  - A given actor may be member of multiple social structures
  - Social structures may be associated with different jurisdictions
  - Social structures may involved in disputes with one another
    - Requiring conflict resolution
  - Social structures inform and limit the "kinds" of governance that can be effectively deployed

3.3.2 Resource and Ownership

Communication about and between, visibility into, and leveraging of resources requires the unambiguous identification of those resources. Ensuring unambiguous identities implies

- Mechanism for assigning and guaranteeing uniqueness of globally unique identifiers
- Identifying the extent of the enterprise over which the identifier needs to be understandable and unique

- Mechanism and framework for ensuring the long-livedness of identifiers (i.e., they cannot just change arbitrarily)

~~3.3.2~~3.3.3 Policies and Contracts

- Policies are constraints
  - It is necessary to be able to express required policies
  - It is necessary to be able to enforce the constraints
  - It is necessary to manage potentially large numbers of policies
- Policies have owners
  - The right to establish policies is an aspect of the social structure.
- Policies may not be consistent with one another
  - Policy conflict resolution techniques
- Agreements are constraints agreed to
  - Contracts often need to be enforced by mechanisms of the social structure

~~3.3.3~~3.3.4 Communications as a Means of Mediating Action

Using message exchange for mediating action implies

- Ensuring correct identification of the structure of messages:
  - Identifying the syntax of the message;
  - Identifying the vocabularies used in the communication
  - Identifying the higher-level structure such as the illocutionary form of the communication
- A principal objective of communication is to mediate action
  - Messages convey actions and events
  - Receiving a message is an action, but is not the same action as the action conveyed by the message
  - Actions are associated with objectives of the actors involved
    - Explicit representation of objectives may facilitate automated processing of messages
  - An actor agreeing to adopt an objective becomes responsible for that objective

~~3.3.4~~3.3.5 Semantics

Semantics is pervasive in a SOA ecosystem. There are many forms of utterance that are relevant to the ecosystem: apart from communicated content there are policy statements, goals, purposes, descriptions, and agreements which are all forms of utterance.

The operation of the SOA ecosystem is significantly enhanced if

- A careful distinction is made between public semantics and private semantics. In particular, it MUST be possible for actors to process content such as communications, descriptions and policies solely on the basis of the public semantics of those utterances.
- A well founded semantics ensures that any assertions that are essential to the operator of the ecosystem (such as policy statements, and descriptions) have carefully chosen written expressions and associated decision procedures.

- The role of vocabularies as a focal point for multiple actors to be able to understand each other is critical. While no two actors can fully share their interpretation of elements of vocabularies, ensuring that they do understand the public meaning of vocabularies' elements is essential.

~~3.3.5~~3.3.6 Trust and Risk

In traditional systems, the balance between trust and risk is achieved by severely restricting interactions and by controlling the participants of a system.

It is important that actors are able to explicitly reason about both trust and risk in order to effectively participate in a SOA ecosystem. The more open and public the SOA ecosystem is, the more important it is for actors to be able to reason about their participation.

3.3.7 Needs, Requirements and Capabilities

In the process of capturing needs as requirements, and the subsequent requirements decomposition and allocation processes need to be informed by capabilities that already exist.

- Architecture needs to
  - Take into account existing capabilities available as services

~~3.3.6~~3.3.8 The Importance of Action

Participants participate in a SOA ecosystem in order to get their needs met. This involves action; both individual actions and joint actions.

Any architectural realization of a SOA ecosystem should address:

- How actions are modeled:
  - Identifying the performer or agent of the action;
  - the target of the action; and the
  - verb of the action.

Any explicit models of joint action should take into account

- The choreography that defines the joint action.
- The potential for multiple joint actions to be layered on top of each other