IT University of Göteborg

CHALMERS | GÖTEBORG UNIVERSITY

An Integrated Model for SOA Governance

An Enterprise Perspective

KINGKARN KANCHANAVIPU

CHALMERS   UNIVERSITY OF GOTHENBURG

IT University of Göteborg
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2008

Abstract

The concept of Service Oriented Architecture (SOA) expresses a new idea for organizing a service-based business environment. However, without wise governance, SOA fails. Therefore, this study focuses on the elucidation of the following issues: *What are the critical roles of SOA Governance that promotes the attractiveness of a service-based business environment as well as what factors can inhibit the role of SOA Governance?* The primary objective is based on the believe that an integrated model of governance can improve the understanding of people, i.e. stakeholders, in their efforts to establish a comprehensive architectural pattern that coordinate any effort aiming to make service-based business environment attractive.

This work elucidates the roles of SOA Governance. Accordingly, the primary and most significant role of such governance is the establishment and management of a negotiated and accepted SOA. The equivalent crucial second role of such governance is to use the established architecture and coordinate every related effort that promotes the attractiveness of an architected reality. In this sense, the service oriented architecture follows the wisdom of such governance. In the same sense, any effort that change the service-based business environment of business follows the premises of the established service oriented architecture. Therefore, we can conclude that the wisdom of SOA Governance promotes the attractiveness of a service-based architected environment

Furthermore, the work indicates that there are three main factors that inhibit the role of wise governance. Firstly, a fuzzy, inconsistent, incomplete, ambiguous terminology upon which the concept of SOA and SOA Governance are described, designed, evaluated, etc. Secondly, conflicts of interests and contradictory core ideas, such as alignment, agility, reusability, efficiency, etc. provided by different disciplines, i.e. Software Engineering, IT Management, Enterprise Architecture, etc., that inhibit the choice of a comprehensive architectural style for a service-based business environment. Lastly, the plethora of interesting but otherwise isolated and incomplete models of both SOA and SOA Governance create a sense of uncertainty, and therefore, create the need for endless process of acquisition of information. In the face of these critical issues that inhibit the role of wise governance, our study has developed an integrated model of SOA Governance aiming to clarify the relationship between governance, architecture, and service-based business environment. The model has been tested empirically with acceptable and fruitful results.

The above conclusions may be seen as a result of an adequate approach of inquiry consisting of three main stages. Firstly, a theory that shapes every part of this study. Secondly, the creation of a better and more integrated model (framework) for SOA Governance derived from the distillation of large volume theoretical ideas and models. These theoretical ideas concern both the characteristics of SOA and the wisdom of SOA Governance. Lastly, a fruitful comparative understanding between theoretical and empirical views of SOA environments with respect to identified issues of both SOA and SOA Governance. By this way, we have tested both the validity and reliability of the proposed model. In any case, we do not say the proposed model is completed both theoretically and empirically. What we say is that it is a promising idea to develop further the result of our effort.

Supervisor: Thanos Magoulas

Acknowledgements

I would like to express my appreciation to those who encouraged me to complete this master thesis. I am deeply indebted to my supervisor, Thanos Magoulas, for his great support. Not only overseeing my work but also inspiring me with different perspectives. Several interesting issues had been continuously discussed throughout the period we worked together. I am very grateful to work with such a wonderful and knowledgeable person. This master thesis would not have been accomplished without his great support.

Many thanks to Maria Bergenstjerna and Joakim Svärdström for giving me great ideas and helping me strike through initiating the thesis. Further thanks to Barbro Larsson for her time and effort during the interview. The information I extracted from the interview has become a great part in this work. Thanks to all my friends back in Thailand for their concerns. Even though they are not here with me, their supports always instill the encouragement in me. Also to my friends in Sweden who keep me company, just when I needed it.

Lastly, I would like to dedicate this master thesis to my parents who always support and encourage me in every way throughout my life.

Table of Contents

1.   Introduction

This section provides an introductory understanding of SOA and SOA Governance, through the use of real examples of service-based environment. On this basis, we define the purpose and delineation of the study as well as we outline the whole process of inquiry. Lastly, we outline the different parts that together form the thesis and its supporting arguments.

1.1   Background

Nowadays our business world is an ever changing world. Most of the businesses are customer-centric where customers are the main focus of their businesses. In order to serve customers' needs, the organization must be able to deliver their services in quick response. Thus, business responsiveness is the main requisite from the customer point of view. These organizations usually have goals either to rapidly deliver better quality solution to meet environmental changes or to maintain the current quality but deliver it quicker and cheaper, [BAKE2006]. Accordingly, Service Oriented Architecture (SOA)[1] is seen as the next generation of enterprise architecture that will enable the organizations to use IT and integrate their businesses across the company. SOA provides a new idea of organizing the service capabilities[2] of a business or public enterprise in order to respond quickly to ever changing customers' demands.

Service Oriented Architecture, (SOA) can be defined in terms of relationships between (1) A domain of Service Consumers, (2) A domain of Service Providers and, (3) A domain of Service Brokers[3]. These three domains form together a so-called Service-based Business Environment.
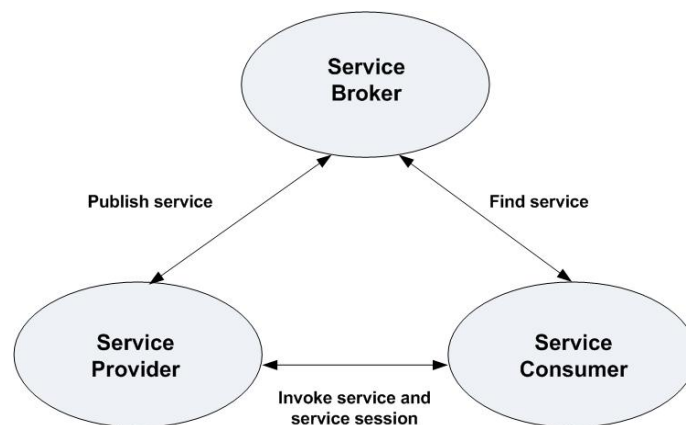
Figure 1: Service-based Business Environment

---

[1] Usually the concept of architecture is used in situation that cannot be treated by methodology because the instability and variability of both requirements and wants.

[2] Service capabilities can be given in terms of human resources, IT resources, etc.

[3] A service broker is neither a consumer nor a provider but a third part that is necessary where a service or business process is composed of several more elementary services that belong to different owners. In this sense, a broker provides information of what services are provided by whom.

However, the same concept can be exemplified - at least - through three different kinds of environments:

- The Google-environment, where Google as a Provider of informational services is contacted by consumers of informational services all over the world. However, this environment is lacking a so-called "Service Broker" because either the collaboration with others Service Providers has not been established or because the collaboration between several owned autonomous National Service Providers has not been defined.

- The Amazon-environment is also lacking a Service Broker despite the fact that this company employs several owned Service Providers all over the world. However, this environment is characterized by informational, transactional and communicative services.

- Lastly, the environment of GP, (Göteborgs Posten), a large newspaper company in Sweden may be treated as a Service Broker because it helps both Service Providers and Service Consumers to establish business linkages. Again this environment is characterized by informational, transactional and communicative services.

According to the above views, SOA environment is defined in terms of service consumer, service provider, and/or service broker. In the same sense, the most usual services in such environment are given in terms of informational services, communicative services, and transactional services.

In summary, a working definition of what SOA is can be stated as follows: A Service Oriented Architecture organizes a business environment (even public or social) where human and technical efforts and/or capabilities are employed in order to satisfy the informational, functional, communicative, etc. requirements, wants, expectations of people with or without the assistance of so-called Service Broker[4].

However, an architected business environment in accordance with the principles of SOA, fails without a wise[5] or sound SOA Governance. Accordingly, the motivation underlying this study can be explained by the facts indicating that 60%-80% of all SOA developmental efforts fail[6]. The reasons behind these failures may be many. Therefore, a

---

[4] In most cases, a Service Broker is a necessary part of a SOA environment indicating the situation where there is more than one Service Providers involved.

[5] In this case, the attribute of wise is derived from practical wisdom defined in terms of balance between rational, emotional and ethical aspects involved in any decision. The rational and emotional aspects (cultural) have been the ground upon which the decision should be treated as meaningful or meaningless, [CHEC1985]. Furthermore, the case of ethical aspect means a mutual respect for the interest of whole and the interest of parts simultaneously. Accordingly, every such decision is political decisions because it is defined, negotiated, and established by the acceptance of the stakeholders. In summary, every wise decision aligns rational, emotional, ethical aspects. However, rational decision (decision based on the best available knowledge) is necessary but cannot be successful if they are not aligned by the emotional and ethical dimension, [HEDB1980] and [ACKO2003].

[6] See Appendix F

secondary interest in this work is the need to know what factors promote or inhibit SOA Governance in their effort to establish and manage an attractive SOA-based Business Environment.

The focus of this study covers the foundational issues of SOA Governance, namely; (1) who has the right to make decisions and change the concerned environment, (2) what kind of issues are typical decisions within such environment, and lastly, (3) how these decisions are made as well as how people evaluate the effects of the concerned decisions.

1.2   Purpose of the study

The main purpose of this study is to create an integrated model of governance that should promote the understanding of people i.e. stakeholders that are engaged in the establishment of a comprehensive, understandable, and meaningful architectural style[7]. The main purpose of this architectural style, i.e. a business driven SOA, is to coordinate and organize everything that can promote the attractiveness of a service-based business environment. The desirability of environmental attractiveness concerns the needs, wants, expectations, etc. of all stakeholders, i.e. consumers, providers, owners, etc.

The process of integration refers to the construction of a more complete, consistent and meaningful model of Governance out of (1) the isolated and dominated models of SOA, and (2) the fragmented and dominated models of SOA-Governance.

We believe that we can absorb a lot of misunderstanding through an integrated model that covers both the aspects of continuous operations of a service-based business environment and the aspects of wise governance aiming to promote the attractiveness of such environment.

1.3   The problem statement of the study

In accordance with the above purpose for improving our understanding, the main question of this study can be stated as follow:

> *What are the critical roles of SOA Governance that promotes the attractiveness of a service-based business environment as well as what factors can inhibit the role of SOA Governance?*

However, in order to provide a fruitful answer, we can decompose the above problem statement and provide the basis for an explanatory theory that promotes the understanding of the following issues:

- What kind of governance model promotes best the soundness of service oriented architecture?

---

[7] Any style expresses similarities rather than differences. This means that two environments express the same architected pattern such as the SOA configuration with or without service broker [See Figure 18].

- What kind of architectural style[8] promotes the organization and coordination of any effort toward an attractiveness of business environment?

Furthermore, there is another issue that belongs to our investigation and clarification of the SOA Governance, namely, what factor inhibits the crucial roles of SOA Governance.

1.4   Delineation of the study

Our study is delineated to focus on the following significant aspects of SOA and therefore SOA Governance at the business level.

Firstly, the delineation covers the ever changing nature of alignment between business requirements and business effort (human and IT efforts). The logic of business, the logic of technical systems and the logic of humanity follows different clocks. Therefore, the expected alignment is a case of evolution rather than revolution[9].

Secondly, this study concerns business services rather than programming (software), communicative services rather than web services, services in general rather than self-services.

Thirdly, the study focuses on heterogeneous business environment involving more than one service providers as the owner involved in a particular business process. Therefore, every form of collaboration between enterprises, such as partnership, contractual arrangement or alliance is relevant to this study.

Lastly, the empirical part of the study is based on the case study. This is the consequence of decision about information acquisition related to this thesis.

1.5   Outline of the inquiry process

The approach of inquiry of this study consists of three main stages. Firstly, the creation of a conceptual framework (the theory underlying in this study) derived from the primary problem statement of the study. Secondly, the creation of a better and more integrated model (framework) of governance derived from the distillation of large volume theoretical ideas and models concerning both the characteristics of SOA and the wisdom of SOA Governance. Lastly, a fruitful comparative understanding between theoretical and empirical views of SOA environment with respect to various issues of both SOA and SOA Governance.

1.6   Outline of the report structure

The rest of the paper is divided into five chapters. In Chapter 2, we describe the approach and method of our inquiry. Chapter 3 deals with current model and theory of both SOA and SOA Governance. In Chapter 4, we distill the above theoretical view in order to integrate the model of SOA and the model of SOA Governance into a unified whole, furthermore, we use this model to outline the issues that should be used to verify empirically the reliability of the proposed model. Chapter 5, we present the empirical view from the

---

[8] Behind any architectural style are one or more core ideas. In the case of SOA, such ideas are efficiency, agility, alignment, etc.

[9] We will describe this aspect later on.

reality of collaborative effort of academic libraries in Sweden. Accordingly, these efforts are coordinated through a service broker called LIBRIS. Lastly, in Chapter 6, we provide a comparative analysis and discussion between the theoretical and empirical views of governance, by this way, we derive both partial and final conclusion.

Finally, an interesting and fruitful part of this thesis is the content of the appendix. In this section we have tried to provide the contradictory nature of both the concepts of SOA and the concepts of SOA Governance. Thus this part of the thesis covers with the following issues: (1) What is SOA, (2) What is SOA Governance, (3) What are the differences between SOA governance and IT governance, (4) Reusability vs. Agility , as well as (5) Why SOA fails.

2. Methodology

In this section we describe the whole methodology (logic of inquiry employed in this thesis). Accordingly, the approach that has been followed in the inquiring of understanding the issues of governance is both normative (theory driven) and descriptive (experience driven). However, the whole process of inquiry can be described in the activities below, namely:

- Establishing the foundation underlying the proposed model
- Model delineation and scoping
- Model construction
- Model verification
- Derivation of partial and final conclusions through comparison
- Judging the quality of the proposed model of SOA Governance (Validity, Reliability and Attractiveness)

2.1 Establishing the foundation underlying the proposed model

The main concern of this thesis is to develop a model for governance that promotes the choice of a comprehensive architectural pattern as well as explains how the proposed architectural pattern promotes the comprehensibility, understandability and attractiveness of service-based business environment.

In this way, this theory is designed to provide a sound answer to the problem statement of this work. Firstly, out of a management and governance perspective we need an architecture that promotes the comprehensibility of the architected environment, without comprehensibility, there is no chance for manageability and governance. Secondly, we need an architecture that promotes the mutual understanding of how the architected environment is shaped and continuously adapted to the ever changing views of the governance. Lastly, we need an attractive architecture that promotes the interest of all parties that is expected to be represented by governance.
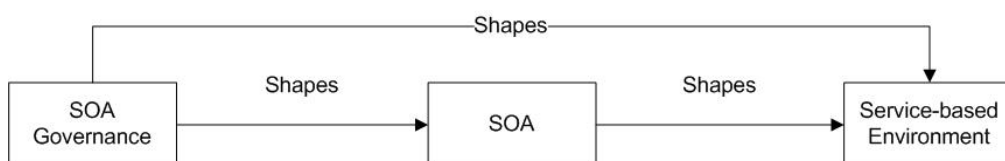
Figure 2: Towards a sound theory of SOA Governance

As can be seen in Figure 2 above, the logical nature of our inquiry can be expressed in the following way:

- SOA Governance shapes (creates) Service Oriented Architecture
- Service Oriented Architecture shapes (creates) Service-based Business Environment
----
- SOA Governance shapes (creates) Service-based Business Environment

Figure 2 indicates the following concepts:

Firstly, SOA Governance shapes the service oriented architecture. In the other word, determine a particular architectural pattern that is in agreement with a core idea defined by the governance. In this case, such idea can be given in terms of alignment, agility, reusability, etc.

Secondly, the selected architectural pattern i.e. SOA shapes the behavior of service-based business environment. However, this pattern remains stable so long the expectations of consumers do not change. This is like tango where a dancer is expected to dance just tango. However, in SOA environment, it is expected that the providers must be able to dance different dances depending on the nature of consumers' ever changing demands. Thus, tango is not enough to respond to the demands of a dynamic and heterogeneous environment. Accordingly, the rhythm reflects the form (architectural pattern) that determines the identity of the dance, independently of the dancers, despite of their professionalism. In the same sense, it is the dancers (the providers) that can be involved in many other dances. This is the nature of agility (following the spontaneous and unknown rhythms) in a dynamic and heterogeneous social reality.

Lastly, upon these two grounds, we can derive that the governance of SOA shapes and develop both capabilities and assets in order to respond to the demands of the service-based business environment.

2.2   Model delineation and scoping

The whole process of inquiry has been supported by the following theories and models; (1) The model of research proposed by Jönsson and Hedberg, [HEDB1978] (2) The Framework for understanding the Enterprise Morphology, -FEM model[10]-, [SVÄR2006] and (3) The model of management that has been proposed by Thompson, [THOM1967].

Firstly, following the ideas of Jönsson and Hedberg, [HEDB1978] in the inquiry of managerial and organizational issues we create first a reference model of the concerned enterprise and then we verify the model empirically.

Secondly, the FEM Model, [SVÄR2006] was used in order to provide the frame upon which the Model of Governance was constructed as well as to "localize" the scope covered by our model. By this way, we have created a logical theory that makes a clear distinction about the role of SOA and SOA concept as well as a clear distinction between the issues of architecture and architectural patterns covered by the proposed model from the issues of infrastructure that has been excluded from the model. Therefore, we have focused on the understanding level of SOA and SOA Governance rather on the technical issues of implementation. Similar idea has been reported by Bloomberg, [BLOO2006]. However, this consulting firm has proposed another interpretation of the same relationships. Thus, the use of FEM model has supported the study in the following issues (1) The focus on the level of understanding and common sense, (2) The integrated view of management, i.e. the governance rather the en-dimensional view of actions, (3) The distillation of relevant aspects of service quality, and (4) The clarification of the scope of a service oriented architecture

---

[10] See Appendix A

with respect to the constituted sub-architectures such as process related, structural, infological, socio-cultural, and contextual.
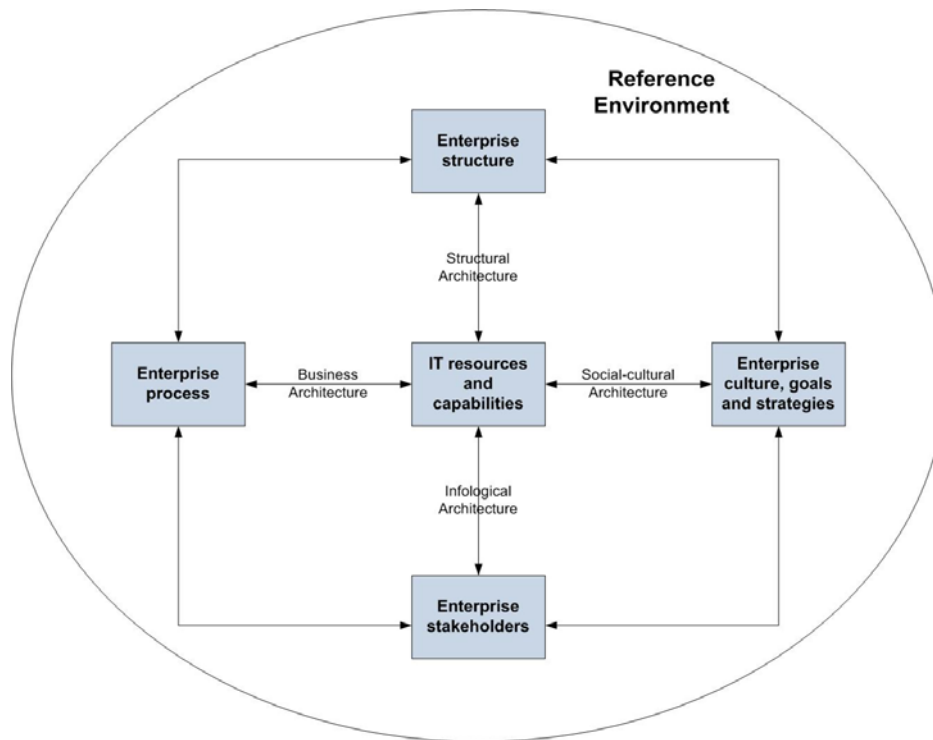
Figure 3: Enterprise architecture according to FEM model, [SVÄR2006]

The last supportive model was the idea of Thompson, [THOM1967], dealing with the nature of decisions in the context of organizations and institutions. The concept of governance is based on two different but well integrated dimensions. The first dimension concerns the ends of actions and the related uncertainty when we try to determine that issue. In the same sense, the second dimension deals with the issues of the course of actions required to achieve the concerned expectations, goals, purposes and the like.

Nature of governance	Clear means of action	Unclear means of action
Unclear ends of action	Negotiating and compromising	Inspiring
Clear ends of actions	Planning	Judging

Table 1: The nature of governance

The model above illustrates the nature of governance in terms of planning, judging, negotiating and compromising and inspiring. These four derived views demonstrate the expected behavior of governance.

In summary, whereas the concept of management refers to the course of actions, the concept of governance is derived from the integrated effects of both dimensions.

2.3 Model construction

The normative part of the inquiry was related with the efforts to create an integrated model of SOA Governance in the context of a SOA architected business environment,

Secondly, a proposal on a way to integrate SOA models with the SOA governance models is introduced and by this way provides a better platform for understanding the issues of operations and the issues related with the governance of these operations.

Thirdly, the integrated model of governance is used in order to develop the significant queries of inquiry. By this way, the theoretical views of SOA governance are distilled and at the same time the acquisition of empirical information about the same topics is prepared. Thus, this treatment we establish the grounds for comparability between the theoretical and empirical views of SOA governance.

2.4 Model verification

The purpose underlying the descriptive part of inquiry was to create systematically the empirical views of the study and by this way provide the material for the verification of our proposed model. This verification has been performed in form of a case study dealing with the collaborative environment of academic libraries in Sweden. The effort of these libraries is coordinated by LIBRIS. This environment involve more than 40 collaborative academic libraries in Sweden in order to satisfy the ever changing requirements of all stakeholders, i.e. (1) students, (2) lecturers, (3) researchers, (4) employees, (5) the state of Sweden as the ultimate owner of the libraries, and (6) the various communities that exist outside from the academic boundaries of such libraries, etc. The empirical materials were collected by two consecutive interviews of three hours each. The first interview was focused to the collection of information aiming to answer the pre-designed questions of the study The aim of the second interview was to verify the validity of the first interview as well as to correct and complete the empirical views accordingly.

2.5 Derivation of partial and final conclusions through comparison

The comparative part of the study outlines the similarities and the differences between theory and experience.

Both theoretical and empirical views of the study are systematically organized and compared in order to draw the partial conclusions of concerned issues. As the study demonstrates, there is a good agreement between the theoretical and the empirical views of the proposed model.

In the same way the partial conclusions have became the sound grounds upon which the final conclusions of this inquiry is based. Accordingly, we belief that we have provide an answer to the current issue of governance that promotes satisfactory the understanding or at least remove much of the existing misunderstanding of how SOA and SOA Governance are related to each other. The final result of this study is given in terms of a model that focuses towards a better understanding of governance. We hope that the model can support any effort of further research aiming to better clarify the complex nature of SOA Governance.

2.6   Judging the quality of the proposed model of SOA Governance

The quality of the proposed model follows the considerations of Jönsson and Hedberg, [HEDB1978]. In this sense, the issues of model validity are derived from the distillation of existing models of both SOA and SOA Governance. Furthermore the issues of model reliability are derived from the comparison between the theoretical and empirical views of the proposed model.

In any case, the proposed model was expected to provide a fruitful answer to the stated problem and simultaneously satisfy the requisites of validity, i.e. the expected harmony and consistency between this proposed model and the existing theories of SOA and SOA Governance. In the same way it must satisfy the requisites of reliability, i.e. the expected harmony/consistency between this proposed model and the real world of today. We believe that both criteria of quality have been satisfied in some degree.

The attractiveness of the model can be determined in the following terms:

Firstly, the proposed model concerns and focus on the issues of SOA and SOA Governance the architectural and therefore understanding level. By this way we limit the terminological confusion that is so characteristic in the formative age of any new idea.

Secondly, the model covers the known core ideas underlying the architecture, i.e. alignment, agility, reusability, efficiency, as well as a combination of these core ideas.

Thirdly, the model has been very fruitful in the management of this study.

However, in the current study the model has been tested with respect to the requisites and issues of agility.

The logic to be followed by our inquiry is illustrated in Figure 4 below. Accordingly, this figure demonstrates the relationship between Governance, Architecture and Business Environment.

In summary, this thesis is that SOA Governance shapes a Service Oriented Environment. This means that the governance is directly responsible for (1) The foundation upon which SOA is based i.e. the establishment of policies purposed, principles, criteria of evaluation, constraints, etc. (2) In the same sense, the governance is indirectly[11] responsible for the delineation, definition and associated efforts expected by SOA and which are related to the processes, practices, procedures, etc. involved in the design, development, deployment, operation, and evaluation of such environment according to the foundation described above. These two aspects together represent the grounds upon which our thesis is based. Thus, firstly, the thesis is represented by the relationship between SOA Governance and Service-based environment. Secondly, the first argument that supports the above thesis is given by the relationship between SOA Governance and SOA. Lastly, the second argument that supports the thesis is given by the relationship between SOA and Service-based environment.

---

[11] Usually the governance defines the foundation for coordinated efforts. However, the efforts are implemented by other people or organization rather than the governance itself.
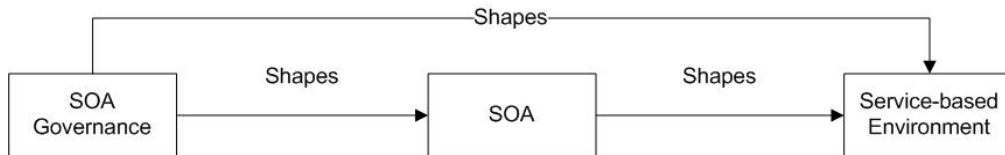
Figure 4: Outline of a sound theory of SOA Governance

2.7   A last word

From the beginning, the study was planned to have a more sophisticated process for the empirical part of the study, [See Appendix 9.1 and 9.2]. However, at that moment it was clear that any such effort should lead to a fuzzy empirical build of reality because the formative age of both SOA and SOA Governance concepts. Some states that within the time horizon of 5 years, the SOA concept should be just identical with the EA concept. Therefore, the first plan of information collection was abandoned. The decision to use the academic environment of Swedish libraries was ease because this environment is an attractive SOA environment.

3.   Theoretical Views of SOA and SOA Governance

In this section we describe enterprise oriented concepts of SOA and SOA Governance through the assistant of FEM model. In this sense, this chapter deals with the characteristic of SOA, the foundation of SOA Governance, as well as the factors that inhibit SOA Governance. This section begins with a definition of the concepts of architecture.

3.1   Architecture concept in general and enterprise architecture in particular

According to the OneLook dictionary [ONEL2008], the concept of architecture has several definitions.

Quick definitions (***architecture***)

- ***noun***:   the profession of designing buildings and environments with consideration for their esthetic effect

- ***noun***:   an architectural product or work

- ***noun***:   the discipline dealing with the principles of design and construction and ornamentation of fine buildings (Example: *"Architecture and eloquence are mixed arts whose end is sometimes beauty and sometimes use"*)

- ***noun***:   (computer science) the structure and organization of a computer's hardware or system software (Example: *"The architecture of a computer's system software"*)

Table 2: The list of "quick definitions" of architecture provided by the OneLook dictionary

Accordingly, the first definition of architecture refers to the design of composite objects like buildings or the design i.e. organization, of environments of any kind. What make sense in this definition is the esthetic expected effects of architecture.

The second definition of the concept, the architecture refers to the product of architectural work. Accordingly, this definition clarifies the vagueness of the first definition because the product of architectural work is something real such as a building rather than just the design or model of a building.

The third definition of architecture deals with the discipline of architecture that from the days of Vitruvius[12] [MORG1914] until our days the aim of this discipline is to apply wise architectural principles and thereby produce useful, stable and beautiful artefacts like houses or well organized and attractive social environment like cities.

The last definition of architecture, concerns the design, organisation, construction, operations etc. of systems such as (1) computer systems, (2) software systems, (3)

---

[12] Marcus Vitruvius Pollio (born c. 80–70 BC, died after c. 15 BC) was a Roman writer, architect and engineer. Most inferences about his life were extracted from his only surviving work De Architectura,

communication networks or any other part that belong to an information technological infrastructure.

Accordingly, IEEE 1471[13] [WIKI2008] is the first formal standard for software architecture or system architecture. It focuses on the description of an architecture as the concrete artefact representing the abstraction that is software architecture or system architecture.

IEEE 1471's contributions lie in the following (in this list, items in *italics* are terms defined by and used in the standard):

- It provides *definitions* and a meta-model for the description of *architecture*

- It states that an *architecture* exists to respond to specific *stakeholder concerns* about the software/system being described

- It asserts that *architecture description*s are inherently multi-view, no single *view* captures all stakeholder concerns about an architecture

- It separates the notion of *view* from *viewpoint,* where a *viewpoint* identifies the set of *concerns* and the *representations*/*modelling techniques*, etc used to describe the *architecture* to address those *concerns*.

- It establishes that a *conforming architecture description* has a 1-to-1 correspondence between its *viewpoints* and its *views*.

- It provides for capturing *rationale* and inconsistencies/unresolved issues between the *views* within a single *architecture description*

Despite these interesting characteristics of systems architectures it remains unclear if these systems belong to the control of a computer (operating system) or to the control of a social or business enterprise. The definitions below aim to provide some fruitful reflections about the meaning of enterprise architecture.

> **Enterprise Architecture** is the organizing logic for business processes and IT infrastructure reflecting the integration and standardization requirements of the firm's operating model, (Wikipedia)
>
> **Enterprise architecture** is the explicit description and documentation of the current and desired relationships among business and management processes and information systems (www.army.mil/escc/erp/aetg_terms.htm)
>
> **Enterprise architecture** is a comprehensive framework used to manage and align an organization's Information Technology (IT) assets, people, operations, and projects with its operational characteristics. In other words, the enterprise architecture defines how information and technology will support the business operations and provide benefit for the business. It

---

[13] IEEE 1471 is the short name for a standard formally known as ANSI/IEEE 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems.* The Recommended Practice is one type of IEEE standard whose adoption and interpretation are the responsibility of the using organization. This standard was published in 2007 as ISO/IEC 42010:2007, *Systems and Software Engineering-Architectural description* [IEEE1471].

illustrates the organization's core mission, each component critical to performing that mission, and how each of these components is interrelated. These components include: (1) Guiding principles (2) Organization structure (3) Business processes (4) People or stakeholders, (5) Applications, data, and infrastructure (6) Technologies upon which networks, applications and systems are built (National Institutes of Health, NIH)

Table 3: Definition of Enterprise Architecture

In this work we define the concept of enterprise architecture in terms or relationships between information systems and the essential constitutional parts that together form a public or a business enterprise. These parts are given in FEM model in terms of (1) Business processes, (2) mission, vision, goals, values, etc. (3) the structure of decision rights and responsibilities, and (4) the stakeholders [SVÄR2006].

Furthermore, the concerned relationships may be real or planned, stable or agile, formal or natural, rational or emotional, informational or decisional, transactional or relational, etc.

3.2   Service and Service Oriented Architecture

Service Oriented Architecture is considered as a best practice for the past three decades, [BAKE2006], its objective is to create: firstly, business agility and secondly, developmental economy in terms of reusability of IT components. However, this section describes the definition of the concepts of service and of Service Oriented Architecture (SOA). Figure 5 provides a sound working definition of what SOA is.

S	O	A
Service	**Oriented**	**Architecture**
Capabilities performed by one for another to achieve a desired outcome	Aligning architecture to enable a collection of services to be linked together to solve a business problem	The fundamental organization of a system by its capabilities, their interactions, and the enterprise environment

SOA is an architecture approach for organizing and using services to support interoperability between enterprise data assets and application

Figure 5: SOA Definition, [McLE2006]

According to Bloomberg and Schmelzer, [BLOO2006], SOA is an enterprise architecture[14]. SOA organizes the services that are involved in a business process aiming to respond quickly to consumers' demands. Therefore, the composition of any business process consists of individual business functions called services. Services may have (1) derived from

---

[14] See Appendix B

existing (legacy) systems such as Financial Management System or Order Management System or (2) developed in-house, or (3) "borrowed" from external service providers.

3.2.1 Service Concept

3.2.1.1 Overview of service concept

A service is a constituent part of a well-defined business process. Furthermore, service can be implemented either through the use of human effort only, IT effort only, or effort from both sides. In this sense, a business process involves several services and therefore more than service providers that are the owner of these services. The picture below expresses a conceptual view of relationships between services, activities and business processes as well as efforts involved in the implementation of services. The natural implementation involves both human efforts respectively IT efforts.

A service, due to its functional nature[15], operates independently of other services as well as independently of temporal aspects, i.e. state of any other services[16]. Services have a well-defined set of interfaces and operate through a pre-defined contract between the client of the service and the service itself, [IFEA2008].

Services can be either IT-based or human-based capabilities. In the first case, the procedure of user login (identity control) is a good example of a technical service. In the second case, the routine of checking customer credit is a good example of human driven capability. Both cases are based on interaction between consumers and service providers.

3.2.1.2 Specific properties of service concept

A service can be characterized by the following properties:

- A service may be a repeatable business task or reusable function.
- A service is always a self contained task (particular function that performs a well delineated business task). The only dependency is the environment which is given in terms of input and output relationships. The interesting of such task is not logic but the provided results as well as the effort used in that service. For instance, a service that provides information about the taxes of a person at a particular year, in this case, the service is given in terms of output.
- A service is always functional rather than multifunctional. This means that there is always one output from that particular service. However, the input may be more than one.
- A service is expected to be independent of context as well as state or location. This means that the same service can be used in several different environments. For instance, a service that calculate the salary of the

---

[15] Traditionally, a function can be expressed as a black box in order to indicate that our interest is not the internal logic of that function but the input and output dependencies either to other functions or to the environment. Furthermore, the value of functional architecture is derived from the interrelation between several functions that participate in the completion of the same tasks, that is, the provided value to consumers.

[16] Usually, the logic of function does not need to save intermediate states. In other word, the definition of function is time-independent. However, the logic of a process such as business one may be dependent of saving the value states involved in that process.

employees. However, several companies existing in different business or in different industries can use this kind of service.

- A service is expected to support the achievement of business agility.
- A service has a clear ownership. However, several different owners can provide the same service. For instance, Microsoft can provide service for calculating salary but the same service can be provided by IBM, or Oracle, etc.
- A service can be 'elementary[17]' or 'composite'. In the first case, for instance, a service providing personal information. In the second case, it can be a whole business transaction defined by so-called business process. In the last case, the owner of the service (a particular business task is not the same as the owner of the business process).
- A service can be implemented either of human effort, IT effort or a combination of both. However, in literature of SOA concerns only case of automation (a case where every service involved in transaction is implemented by the use of IT capability)
- A service can be informational, functional, decisional, transactional, communicational, etc. However, we must have a clear distinction between business services and infrastructural services. Infrastructural services are involved in the implementation of SOA whereas business services are only involved in the definition and delineation of SOA. This helps us to talk about the configuration, composition, customization, coordination, etc. at the understanding business level without consideration to how to implement such service.

3.2.1.3 A scenario of a composed service i.e. business process

The best demonstration of the service concept has been provided by Zimmermann Olaf et al., [ZIMM2004], in terms of a business scenario as given below:

- The work order is created when the customer calls to make an appointment.

- For each planned maintenance activity or operation, a separate work order item is created, containing details of the expected usage of parts, supplies, and labor.

- The inventory is checked to ensure that all necessary parts are in stock before the appointment is scheduled.

- A suitably-equipped service bay plus a suitably-qualified mechanic needs to be scheduled for each work order item.

- The estimated total cost is calculated, and the customer approves the appointment, or the scenario terminates and the work order is cancelled.

- Immediately before the appointment, the necessary parts, supplies, tools, and equipment are assembled in the selected bay.

---

[17] An elementary service is the result of the decomposition of a business process. It means that a service that cannot be decomposed more. However, functional decomposition that is implied by the service concept is always a relative concept.

- When the customer arrives, the planned activities are performed, plus any other activities that become apparent when the vehicle is inspected.

- Actual values for parts and supplies used and labor are recorded.

- On completion of all maintenance, the total cost is calculated.

- An invoice is created and presented to the customer.
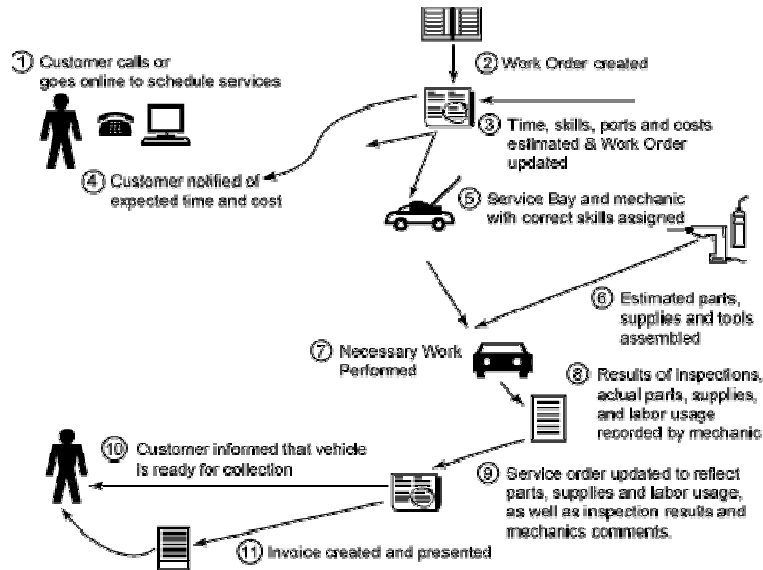
Figure 6: Macro flow example of Work Order, [ZIMM2004]

In the same sense, the services involved in this particular business transaction, i.e. business process, are given below:
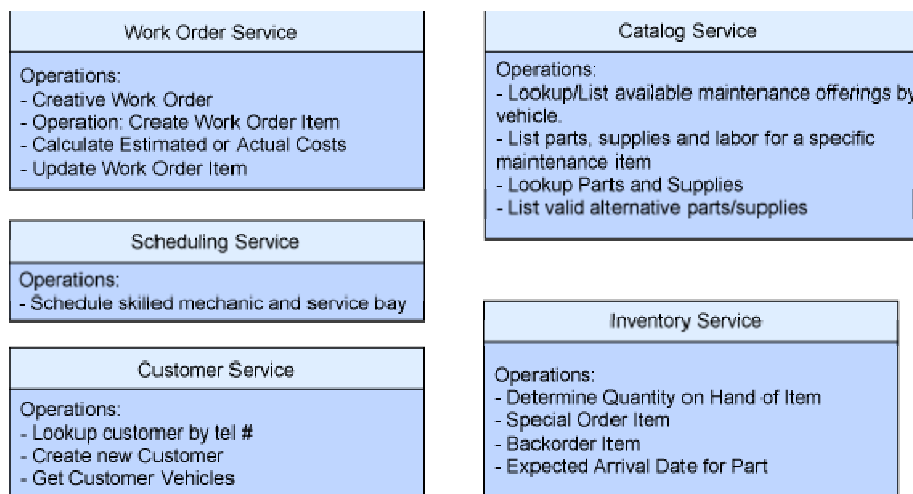
Figure 7: Services Model example of a Work Order, [ZIMM2004]

3.2.2 Service Oriented Architecture Concept

A dominated definition of Service Oriented Architecture states that SOA is a design principle for IT solutions based on standard, modular software design. It builds upon ideas of object-oriented analysis and design and client/server architecture. Instead of using fewer and larger pieces, SOA uses smaller pieces of software to communicate with one another. Accordingly, this definition equates the SERVICE concept with that of SOFTWARE.

Another dominated interpretation of SOA states that the SOA concept is derived from the process of breaking down an individual business functions into business services. This definition is better since business services can be understood, shared, and maintained etc. more easily, as well as implemented on different IT environments such as ERP, legacy systems, computing platforms, etc. in order to optimize their performance.

Thus there are thousand definitions of what SOA is. However, there are the above two interpretations that dominate the literature. In Figure 8, concept of Service Oriented Architecture is demonstrated.
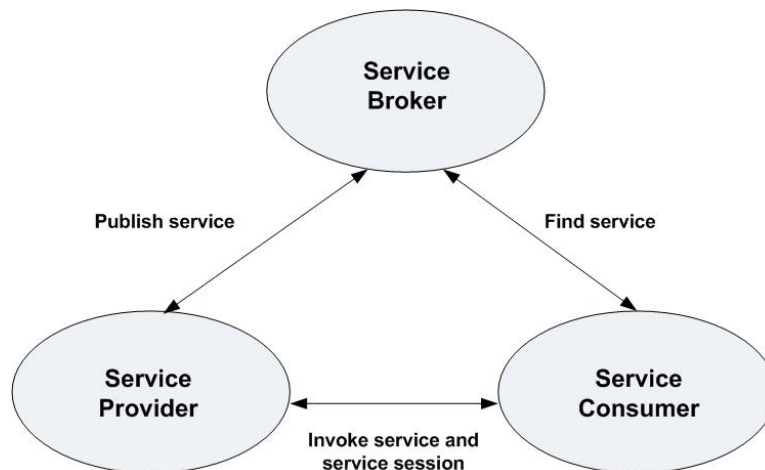
Figure 8: Concept of Service Oriented Architecture

Service Oriented Architecture defines a request-respond interaction between three parties; (1) Service provider who published a service description and provides the implementation for the service, (2) Service consumer who either uses the service or finds the service in a service registry and invokes the service, and (3) Service broker who provides and maintains the service registry, [ARSA2005]. Once service consumer requests for a particular service, the service broker will send the response back and affirm the available service providers. The transactions for the requested service are then made between service consumer and service provider. There are also transactions between service provider and service broker where new services are reported to service broker by service provider. Maintaining services are done between these two roles as well.

The characteristic of Service Oriented Architecture can be given in terms of the following requisites:

- The relationships as well as the interactions between services establish the so-called message based dependences i.e. loosely coupling between services
- The scope of SOA architecture is given in terms of a business process. Usually, such process consists of several more elementary services
- The services involved the composition of a particular business process may have different owners
- The coordination of services that belong to the same business process may be a case of orchestration[18] (a case of centralized coordination) or a case of choreography[19] (a case of decentralized coordination)
- In many cases, SOA architecture does not allow redundancy of services, this means that a new provider is introduced and take over the control of common services
- SOA is technology independent where it can be processed regardless operating systems and languages

However, SOA has been interpreted in many different definitions, for example:

- SOA is identical with enterprise architecture
- SOA is the alignment between business, needs, wants, and IT and human capabilities (this is considered as the best definition of SOA by far)
- SOA is the alignment between business process and IT capability (case of automation)
- SOA is the integration and coordination of required capability (activity or function) that belong to different collaborated owner (this is a very narrow definition because human capability, service broker and service provider are excluded)
- SOA is not a software but it is a service

The first step in building a SOA environment is to identify the various independent components that make up a business process. Each component is a logical grouping of the people, technology, and resources that deliver specific business value, with the potential to operate independently. Components have well-defined interfaces; each receives input, processes relevant tasks, and outputs the results to other components. These components are viewed as services which can be consumed by people or other IT components. Each service is connected with one another in order to communicate among services regardless of the platform and location.

In summary, it is very critical to understand the so-called business- or enterprise-driven SOA. Accordingly, the definition of SOA is derived totally from the requirements and

---

[18] Orchestration is a form of centralized coordination of action where a particular unit becomes responsible to coordinate a transaction with or without the assistance of IT system (think, the case of LIBRIS under the period from 1975 to 1995 where customers have no directly access to LIBRIS. In this case, a particular library becomes the coordinator of action).

[19] Choreography is a form or coordination out of equal effort perspective, several units that participate in a certain transaction must participate in the execution of transactions without some central authority.

expectations of business processes and their constituent parts i.e. services. In the same sense, the implementation of services is a distinct process. The same services can be implemented through reusable owned services or by borrowing services. The figure below demonstrates the independence of SOA from its implementation.
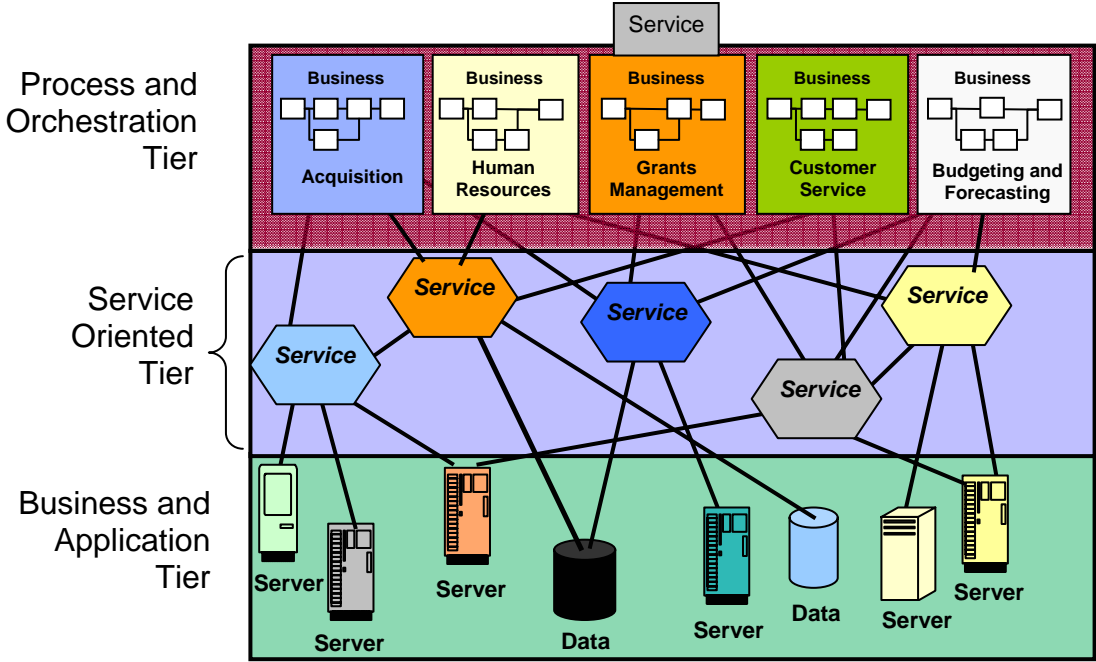
Figure 9: A clear distinction between Business Processes, involved services, and the technical implementation of services, [NICK2005]

As illustrated in Figure 9, services have two significant dimensions. The first dimension is definitional and concern the establishment of clear relationships between business processes and services. The second dimension deals with the implementation of services with available resources and capabilities. Some of these resources may be technical capabilities whereas other capabilities are just human ones.

3.2.3   *Critical characteristics of SOA*

Behind every architecture, there is always a core idea that dictates the choice of a particular pattern. Three such core ideas that dominate the discussion of SOA are identified; these are: alignment, agility and reusability. Thus, according to the current literatures, there are few core ideas or forces that are concerned as critical because the whole organization of the environment is directed by only these ideas. In any case, there is always possibility to define architecture through the consideration of more than one core ideas. In this sense, some literatures describe reusability as the critical core idea whether other literatures concern agility as the only and relevant core idea of SOA. However, there are also literatures that try to define architecture through the balance of reusability and agility. These cases are analyzed later on and we provide more explanation in the appendix.

In addition, IBM has explained the requisites of balance in very reasonable terms. In this sense, understanding SOA is understanding how the concepts of (1) business agility, (2)

alignment of business with technology through SOA, (3) modularity, (4) reusability, etc. hold together.

Aging systems impose serious roadblocks to business agility. If a business process is going to adapt rapidly as the environment changes, so too must the underlying systems that support it......What is needed is a prescriptive approach to modernizing this style of system engineering to one that fosters agility. The SOA approach has been shown to provide the value required by businesses today......SOA aligns IT function to business process function and significantly enhances agility by supporting modular systems. For the first time, businesses can easily construct and manage their business processes in real time without investing heavily in custom application development, [IMBb2006].

3.2.3.1 The issues of alignment

Alignment reflects the requisite of expectations and goals to be satisfied by IT assets, [IMBb2007]. Thus, consumers' every changing needs are expected to satisfied through the assets and capabilities of providers. However, this situation can result to participation of several providers in order to satisfy the consumers' requirements. It is a case of alignment supported by the concept of SOA. Accordingly, a Service Oriented Architecture organizes the services that can be owned by several providers, [OASI2006]. In any case, because of the nature of consumers' requirements, new providers can be included and existing providers can be excluded. The requisite of alignment can always be satisfied with different providers because of the heterogeneous and dynamic nature of consumers' expectations. One of the main issues of alignment is the problem of inter-operability between the involved owners of services, [OASI2006]. Inter-operability means the exchange of information between two or more business entities (organizations, departments, teams, etc.) through the use of their information systems. Therefore, the requisite of inter-operability presupposes some form of integration. In the case of SOA, this form of integration is given through loosely relationship between services but also between consumers and providers. Furthermore, according to OASIS, inter-operability indicates the needs of metadata in order to make the communication between entities possible.

October 27, 2005 - Capgemini has today announced the release of the first business-centred SOA Methodology and Notation available in the public domain; a part of Capgemini's complete Integrated Architecture Framework, this notation and methodology is aimed at helping organisations map out their business architectures and then use this to drive IT and business alignment, [BART2005].

However, it is not clear who is responsible to harmonize the communication between the involved entities. In this case, the question is whether it is a task of providers, broker, or some new entity. In summary, alignment is a necessary precondition for cooperation, collaboration, and inter-operation between consumers, providers, and brokers. Furthermore, alignment is a precondition for business agility which is explained next.

3.2.3.2 The issues of agility

Business agility may be seen as ability of business entity to respond on demand of consumers. However, business agility has many dimension and therefore many definitions. Firstly, business agility means ability to respond quickly to the changes of business environment. These changes can be (1) regulatory, (2) competition, (3) market oriented, (4)

consumers' expectations, (5) etc. In other word, there are changes dictated by business environment.

> *The infrastructure SOA encourages is also more agile and responsive than one built on an exponential number of pair-wise interfaces. Therefore, SOA can also provide a solid foundation for business agility and adaptability, [OASI2006].*

However, there are other definitions of business agility that demonstrate the ability of business entity to innovate either in terms of new product or in terms of a new business logic. In the same sense, business agility means the possibility of alternative ways to satisfy the environment driven requisite through the collaboration of several business providers. However, responsiveness and flexibility are just two views of business agility. Alberts, [ALBE2005], complete this view with resilience, robustness, adaptation.

However, these dominated forms of business agility represents only enterprise agility, there is another form of agility related to management in general and SOA governance in particular. As we have noted, the most usual form of enterprise involves several owners that together promise to satisfy consumers' requisites. However, according to Atkinson and Moffat, [ATKI2005], these kinds of organization (contractual arrangement, partnership arrangement, alliance arrangement, etc.) are based on network form rather than hierarchy form.

3.2.3.3 The issues of reusability

Many approaches indicate that the essence of SOA is reusability. It means that the services can be reused and recombined to support other business activities or other business processes.

> *SOA is a means of organizing solutions that promotes reuse, growth and interoperability. It is not itself a solution to domain problems but rather an organizing and delivery paradigm that enables one to get more value from use both of capabilities which are locally "owned" and those under the control of others, [OASI2006].*

However, reusability is a technology-driven goal aiming to quickly respond to the ever changing business requirements. Furthermore, reusability means cost saving. Lastly, reusability creates dependencies that in some cases can be undesirable because these dependencies are in conflict with business agility[20]. However, there are business cases that reusability is the only solution that quickly provide answer to a critical situation such as catastrophic event, etc.

3.2.3.4 The issues of information and communicative services

Most of literatures about SOA, concern the issues of collaborative processing and delivering the results of processing to the consumers. However, the issues of information and communication has not been treated equally, therefore in some cases we have found out that SOA must clarify the responsibility for both communication and information. The aspects below indicate the necessary and critical factors for inter-operability and information management.

---

[20] See Appendix E

- What kind of information should be exchanged?
- The information should be exchanged from where to where
- Who is responsible for exchanging the information?
- How often the information is exchanged?
- Does the information structure, semi-structure or unstructure?
- What particular way information exchange is established; direct response to receiver or receiver makes the service?
- What kind of information is exchanged; transactional, decisional or innovative?

However, the answers to the above critical factors are not so easy in the case where SOA covers several inter-organization relationships.

3.2.3.5 The issues of human capabilities

Most models we have studied, the human efforts are excluded from the concept of SOA or are isolated through some interfaces, for instance, banking via agents. This means that these models promote automation and communication, for instance, through internet or ATM.

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, [OASI2006].

However, these cases may be representative elementary services. In more composed services covered in the whole business process as that we demonstrated in the introduction, there are several services that involve human capabilities [21]

3.3 The foundation SOA Governance

SOA Governance is the structure of relationships between people involved in the decision processes that shape and manage SOA domains that together form a so-called SOA environment i.e. a service-based business environment. The main reason of governance existence is to coordinate all kind of capabilities in order to achieve the enterprise's goals by adding value while balancing risk and returns, [IBMC2006].

SOA Governance can be seen as the key for the alignment between Business Governance and IT Governance as illustrate in Figure 10.

SOA Governance is a key requirement for successful Business/IT alignment and SOA process success…SOA Governance focuses on the end-to-end SOA environment, [RAYD2007].

Accordingly, IT Governance concerning all issues of infrastructure involved in the realization/implementation of some parts of SOA architecture and characteristic such as visibility, accountability and making IT decisions. Business Governance, on the other hand, focuses on the choices of a particular architectural pattern that satisfy the requisite of comprehensibility, understandability, and attractiveness of the shaped environment.

---

[21] See Figure 6 and 7

Furthermore, in Business Governance determines even what capability should be realized by IT-based efforts respectively human-based effort.
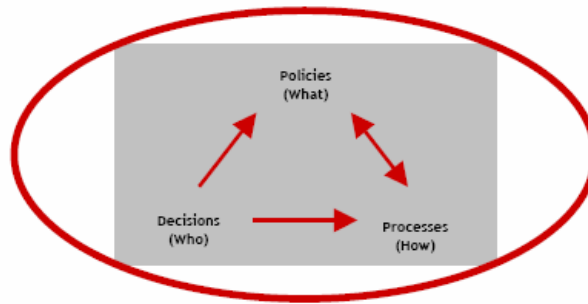
Figure 10: Essence of Governance of SOA, [AFSH2007]

Governance is assigning the rights to make decisions and deciding what measures to use and what policies to follow to make those decisions. The decision rights are assigned to roles in the organizations, not to individuals; so, an aspect of governance is determining organizational roles, [BROW2006].
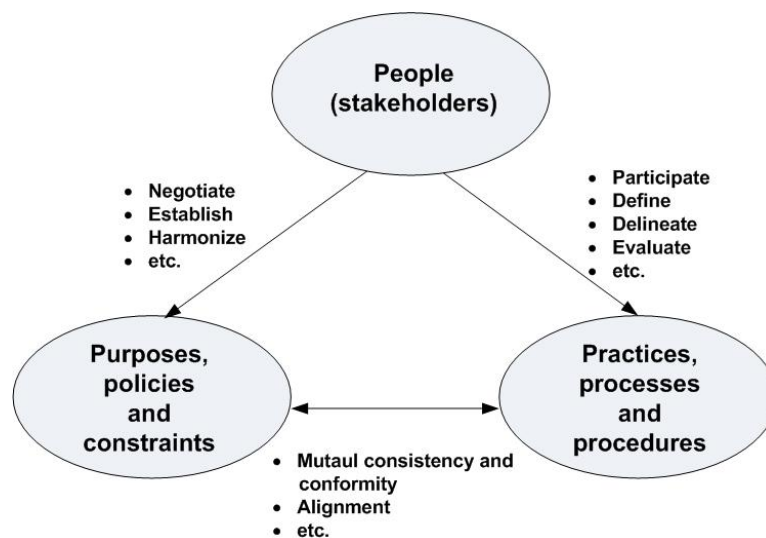
Figure 11: Concept of SOA Governance

In order to improve the successful of implementation of SOA, IBM has found that there is a requirement to develop SOA Governance early in the process. SOA requires improved governance to maintain the level of control needed to support the new Business/IT environment, [BROW2006] and [IBMb2007]. The values provided by SOA Governance are:

- Provides business process flexibility
- Allows improved time to market
- Assists in maintaining quality of services

- Ensures consistency of service
- Measurement of the right business and IT metrics
- Improve communication between business and IT

According to IBM, [IBMC2006], SOA Governance encompasses the set of services, policies, and best practices which enable IT organizations to gain visibility into their SOA, drive reuse of services, define and enforce policies, and manage the life cycle of services.

Thus, SOA Governance describes how people involved with the authority over aspect of the business will consider SOA in their supervision, monitoring, control, and direction of that business entity. How they apply SOA will have an impact on whether the company will be able to attain the vision, mission or strategic goals that the management of the company sets for it. SOA Governance specifies who has the rights to make decisions regarding SOA and IT, what decisions they can make, and an accountability framework that encourages the IT usage behavior corporate management seeks to exhibits.
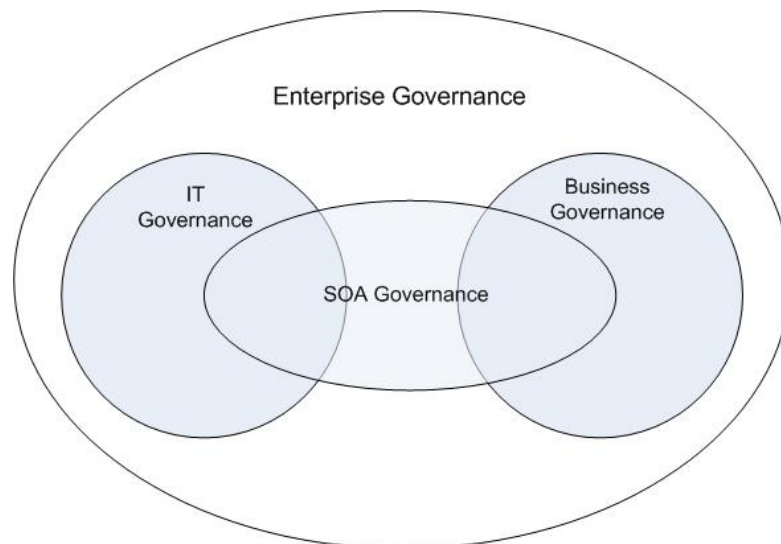
Figure 12: SOA Governance reflects the issues of alignment between Business and IT Governance

SOA Governance encompasses the set of services, policies, and best practices which enable IT organizations to gain visibility into their SOA, drive reuse of services, define and enforce policies, and manage the life cycle of services. In other word, SOA Governance is not equal to IT Governance. Usually, IT Governance concerns only IT issues, however, SOA Governance takes into account both IT issues, human capabilities, and any other kinds of assets required to satisfy stated and unstated business expectations [See Figure 12].

Effective SOA presupposes effective SOA governance. Accordingly, the enterprise of SOA Governance must address the following issues:

- Who has the authority to do it? (As defined by the roles and responsibilities of the service lifecycle processes)
- What has to be done? (The purpose of SOA Governance)

- How is it done? (The governance decision path based process), as well as, how is it measured? (The vitality and conformance checkpoints)

3.3.1 What has to be done? (The purpose of SOA Governance)

SOA Governance is about managing the quality, consistency, predictability, and interdependency of services. According to Schmelzer, [SCHM2007], the issues of loosely coupling is more critical than the issues of defining services.

The companies that are well on the path to SOA adoption know full well that the technical challenges of building and exposing Services are less significant than the hurdles of building loosely coupled, business-relevant Services leveraged across their continuously changing business processes. Indeed, the challenge of making loose coupling a reality is only surpassed by the even-greater challenges associated with organizational and cultural adoption of SOA, [ZAPT2008].

Thus, SOA Governance deals with issues such as; what happen when a service is changed? How can we be sure the service we are consuming is of high quality? How can we be sure a new service is compliant with IT, business and regulatory policies? How can we ensure predictable uptime of a service?

Several organizations apply SOA Governance in order to:
- Establish an architecture that support alignment of their business and IT
- Improve business agility and flexibility
- Optimize SOA benefits
- Enable management and control of services
- Provide traceable business goals within SOA
- Oversee and enforce policy (business design, technical design application security) that directs the organization
- Create policy that directs the organization
- Coordinate the people, policies, and process that provide the framework for management decision-making
- Take action to optimize outcomes related to an individual's responsibility
- Promote efficiency in the organization
- Determine the integrity of services

3.3.2 Who has the authority to do it? (As defined by the roles and responsibilities of the service lifecycle processes)

To describe the authorities, roles and responsibilities of SOA Governance, it can be made in terms of domains of knowledge. In this sense, the logical domain concerned by SOA Governance is described as follow:

- CEO: has the power over all functional domains of business– based on business function
- CTO: has the power over the technology-based domains – sometimes it is desirable to map services to technology-based domains (e.g. when functional service domains cover multiple platforms). Infrastructure services (e.g. error logging, event handling) are good candidates for technology-based domains

- CIO: has the authority of application-based domains – some services are associated with a particular application (e.g. financial accounting for SAP, addition of employees for PeopleSoft)

Another way to describe the authorities, roles and responsibilities of SOA Governance, it can be made in terms of metaphor below. For instance, Weill and Ross, [WEIL2004], as well as Davenport [DAVE1992] describe the forms of IT Governance, respectively the forms of Information Politics as follow:

- **Business Monarchy** – The definition of information categories and reporting structures by the firm's leaders, a group of business executives (CxO), who may or may not share the information willingly after collecting it.
- **IT Monarchy** – Similar to Business Monarchy except that the leaders are individuals or groups of IT executives.
- **Feudal** – The management of information by individual business units or functions, which define their own information needs and report only limited information to the overall corporation.
- **Federal** – An approach to information management based on consensus and negotiation on the organization's key information elements and reporting structures.
- **Anarchy** – The absence of any overall information management policy, leaving individuals to obtain and manage their own information.

However, in the case of a collaborative environment with many different parties like the case of LIBRIS, the issues of ownership of resources become obvious. Therefore it is crucial to see who the owner of resources is and how he or she influences the critical operational and strategic decisions. In any case, the issues of ownership have become an inseparable aspect of SOA definition, [OASI2006].

3.3.3 *How is it done? (The governance decision path based process)*

SOA Governance is about change management, to be more specific, lifecycle management.

According to Nadhan, [NADH2003], there are two approaches of SOA Governance; Central Governance and Distributed Governance.

- Central Governance
    - o Governing body has representative from each service domain and from subject matter experts who can speak to the key technological components of the solution. The central governing body reviews addition and deletion of services, as well as changes to existing services, before authorizing their implementation

- Distributed Governance
    - o Each business unit has control over how it provides the services within its own organization. Requires a functional service domain approach. A central committee can provide guideline and standards.

According to Brown et al., [BROW2006], SOA Governance lifecycle consisting of four phases: plan, define, enable, and measure. These actions are needed to establish, maintain, and enhance an affective SOA Governance framework.

Figure 13: SOA Governance lifecycle, [BROW2006]

Planning Phase
During this phase, the understanding of the overall scope of governance within the organization is focused; also identify areas of improved governance. This phase includes:

- Committing to a strategy for SOA within the overall IT strategy
- Explicitly determining the level of IT and SOA capabilities
- Refining the vision and strategy for SOA
- Reviewing current governance capabilities and arrangements
- Developing a governance plan

Define Phase
This phase focuses on defining and modifying the current governance arrangements and mechanisms. Some important decisions made during this phase include:

- Defining additional capabilities required
- Agreeing on policies for service reuse across lines of business
- Setting funding mechanisms
- Establishing mechanism to guarantee service levels

Enable Phase
After governance approach is designed, solutions are then put into action during this phase which includes:

- Deploying new and enhanced governance arrangement
  - Tracking the decision making process
  - Enabling the policy infrastructure
  - Providing the monitoring tools
- Deploying technology to discover and manage assets
- Communicating and educating business and IT decision-making people
- Enabling the policy infrastructure

Measure Phase

Governance arrangements and mechanisms that were deployed in previous phase are monitored in this phase. This gives the opportunity to evaluate the results and adjust these phases to enhance the effectiveness of SOA governance, if needed. This phase includes:

- Monitoring compliance with policies and governance arrangements
- Analyzing IT effectiveness metrics

In summary, the organizations can apply SOA Governance in order to ensure successful of SOA adoption and efforts by using the following steps, [SOAO2007]:

- Define enterprise goals, objectives, and values
- Form a cross-organization SOA Governance Council to oversee SOA implementation
- Define roles and responsibilities
- Decide funding model
- Perform analysis of existing/potential SOA artifacts and capabilities
- Define SOA principles, standards, and best practices to be followed
- Define SOA governance processes
- Define project management and development approaches
- Define SOA program and service level metrics
- Execute SOA Governance plan throughout lifecycle

3.3.4 *How is it measured? (The vitality and conformance checkpoints)*

SOA has become more common because of its flexible structure which offers an enormous benefit for organizations as to enable cost-efficiency, agility, adaptability, and control of legacy investment. There is a wide range of ways to incorporate system with the principle of SOA. One would be a collection of services and systems which running over a single, large infrastructure, existing unambiguously under a single authority, following business rules, and executing well-defined process[22]. Therefore, the boundary of ownership is clear and encompassing. Another set is a collection of heterogeneous service used by a set of diverse applications, facilitated by various infrastructure capabilities, and governed by business rules known within the applications. In such case, the ownership would be complex and highly ambiguous. Consequently, it is important to consider the relationship of services and applications, and boundaries of ownership, [SMIT2006].

3.4 Summary: Factors that inhibit SOA Governance

3.4.1 *A confusing world of SOA and SOA Governance*

Presently, there are the absence of comprehensible relationships between a service-based business environment, and the governance that creates and manages it. Several models and ideas about SOA governance exist, however, most of the ideas create confusion and misunderstanding rather than understanding. The resulted confusion is based on a plethora contradictory conceptualizations and fuzzy ideas.

---

[22] This situation is discussed later on. It is the case of proposal of new integrated libraries for the universities of Sweden that was totally rejected.

According to a preliminary interpretation of SOA and SOA Governance, there are several conceptions and misconceptions of SOA and SOA Governance.

In the first place, we must clarify the meaning of SOA.

- SOA deals not with Software Oriented Architecture but definitely with Service Oriented Architecture.
- SOA means neither Silos Oriented architecture nor Spaghetti Oriented Architecture but definitely with architectures that convert Silos, i.e. information islands with comprehensible and loosely coupled architectures rather than spaghetti ones.
- The A in SOA means Architecture. In this sense, an architecture is given in terms of principles aiming to support the organization of a particular environment, i.e. (1) identification of constituent parts and their roles, (2) identification of the relationships between these parts, (3) identification of the significant patterns of configuration, composition, collaboration, coordination, customization, etc. involved in such environment. (4) the evolution and development of such architected environment.

Furthermore, the most dominated interpretations of what SOA is can be given as follow:

- SOA is identical with enterprise architecture
- SOA is the organization of business process in terms of IT and human capabilities but omit issues related with broker and consumer
- SOA is the organization of business process in terms of solely IT capabilities
- SOA is an expression of alignment between business wants or requirements and IT and human efforts in environment that involves several providers

The first definition is good but does not concern the issues of alignment. The second one is good but very limited because it leaves outside both the issues of brokers and the issues related with consumers. The third definition deals with the limited cases of automation However, in many cases the human efforts are present and therefore cannon be taken away from the architected environment of SOA.

3.4.2 Contradictory core ideas underlying SOA

According to Mintzberg, [MINT1989], behind every meaningful forms, there is some core ideas or forces, such as, direction, proficiency, innovation, concentration, efficiency, etc. Thus, like the design of a formal organization presupposes one or more of the above forces. The organization of service oriented environment presupposes that SOA is derived from such core ideas. In the context of SOA, we have found that there are such ideas as business agility, reusability, efficiency, alignment, loosely coupling, etc. However, some of these ideas are contradictory; for instance, business agility presupposes relatively high independency of services whereas efficiency stated the requisite of high interdependency between services. Furthermore, when business agility solves the issue of interdependency through redundancy, the corresponding idea of reusability solve the problem through the freedom of redundancy. Alignment in general and strategic alignment between business goals, purposes, visions, missions, etc., and business capabilities and assets create the presupposition for agility. Thus, without alignment there is no chance for agility, flexibility or delivery on demand as IBM

proclaim. In summary, SOA expresses a good idea for balancing the contradictory ideas of reusability and agility. However, according to this study, we have not found explicitly some demonstrated cases where these two ideas exist in harmony. Later on, in the Discussion, we explain this issue with respect to LIBRIS.

3.4.3 *Fragmented and incomplete model of SOA and SOA Governance*

This study indicates the fact that SOA reference models and SOA governance reference models inhibits human-understanding due to their mutual isolation[23]. For instance, in the SOA reference models, we have a clear situation where several providers are involved. However, this situation indicates several issues. Firstly, the issue of decision rights with respect to centralization respectively decentralization of power, i.e. who shall decide and by what means? Secondly, in the case of collaboration between several service owners that belongs to different companies, then, who shall decide and by what mean? Thirdly, in the case of strategic development of consumers' wants that are defined in terms of several business processes then who should participate in the strategic decisions and who in the operational ones? According to this study, SOA reference models lack considerations to SOA Governance. In the same sense, SOA Governance reference models are too abstract because they lack reference to reality. In other word, the models define the stakeholders, what kind of decision should be taken by them as well as how the decision should be implemented and measured. However, what is the subject matter of these decisions? What part of reality should be changed though the implementation of these decisions. Lastly, we have nothing substantial to evaluate. In summary, until the critical issues of integration of models between SOA and SOA Governance, our understanding should remain fragmented and incomplete, and therefore, we don't believe that a meaningful service-based business environment can be reality.

Another reason of the misunderstanding is based on the absence of a common language that can be used to describe, evaluate, design, develop and manage communicate, etc. the issues of governance with respect to a service-based business environment, for instance, the case where a service should be provided by the collaborative efforts of several service providers. Each of the providers owns some part of the composite provided service. Which provider should coordinate the delivery of a particular service? How the responsibilities should be arranged? How the contribution of each provider should be managed? How the collaborative efforts should be improved?

Lastly, which of the stakeholders should be responsible to coordinate the further design development, deployment of such environment? How quickly the different decisions should be executed? With what goals, values, and purposes, visions, etc. in mind?

Thus, there is no such thing as a well organized and supportive service-based business environment without sound governance.

---

[23] According to our study, there is no any explicit established conceptual connection between these two kinds of model. By this way, any kind of relationship is the result of intuition rather than of reason.

4.   A New Model for the SOA Governance

This section concerns the ideas and construction of an integrated model for SOA Governance that follows the main purpose and expectations of this study. The resulted model is expected to promote the understanding of people i.e. stakeholders that are engaged in the establishment of a comprehensive, understandable, and meaningful architectural style[24] for a service based environment. The process of integration refers to two well communicated and defined models, namely, (1) the isolated and dominated models of SOA, and (2) the fragmented and dominated models of SOA Governance. Figure 14 illustrates the input and output of the process of integrations.

4.1   Understanding the elementary and composite forms of services

Services can be either 'elementary' or 'composite'[25]. In the first case, the example of log in or credit control indicates cases of elementary services. In the second case, a whole business process can be seen as composite services.
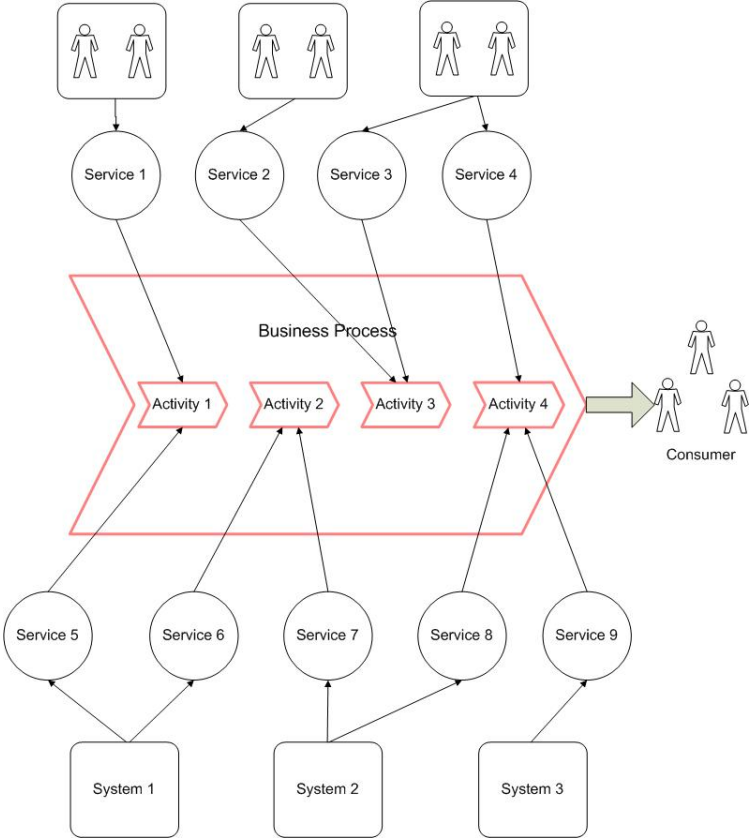
Figure 14: Conceptual view of relationships between services, activities and business process[26]

---

[24] Any style expresses similarities rather than differences. This means that two environments express the same architected pattern such as the SOA configuration with or without service broker [See Figure 18].

[25] The concept of elementary and composite services expresses the most difficult issues of design. In the first case, the question is where to stop the process of decomposition. In the second case, the question is where to stop the process of composition, [CHUR1971].

As mentioned above, a business process can involve one or several responsible organizations in order to provide the service requirements of a particular customer. Thus, Figure 14 illustrates a typical but incomplete view of a service-base business environment. However, a complete view of this environment must include even the domain of service broker.

4.2 Outlining qualities of service out of an architectural perspective

In this section, we provide several lists indicating the qualities of service. We do this with the assistant of FEM model. The model has been used in order to distill the literature and by this way provides comprehensive view of the quality of services.
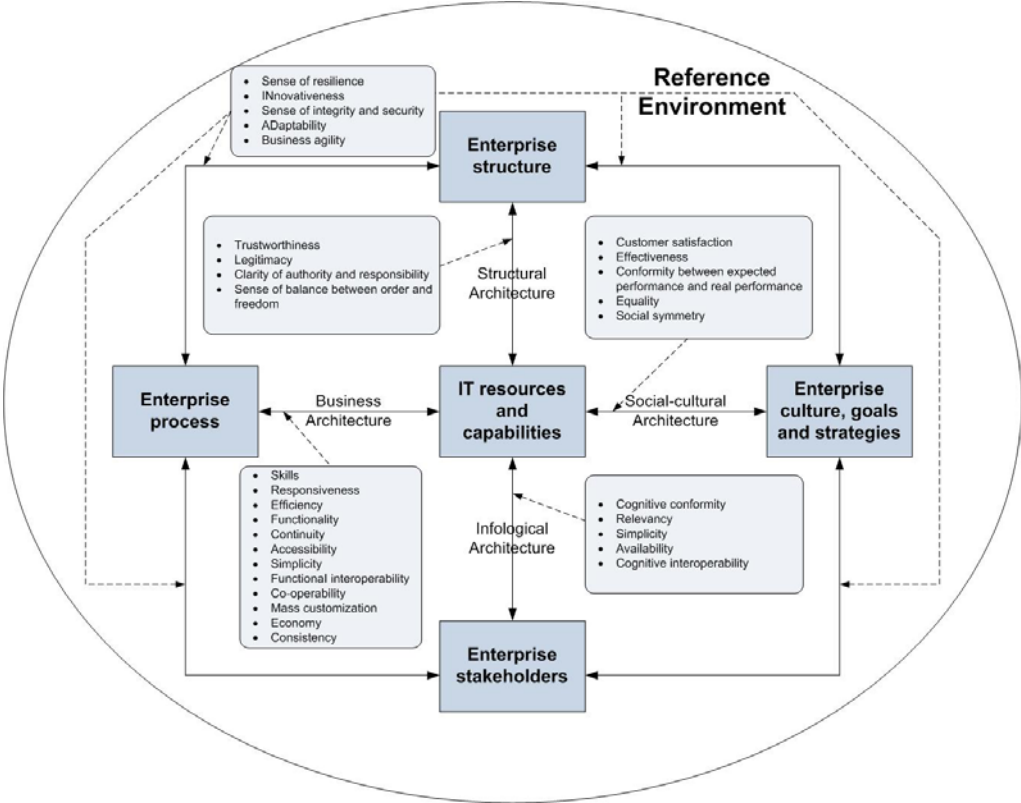
Figure 15: A multidimensional view of service qualities

The above figure indicates the different qualities of services. These qualities aim to characterize qualities of relationships as well as qualities of a whole. The last kind of qualities, we called contextual quality. The lists of qualities are neither complete nor well-understood with respect to their mutual independency. However, it is the most mentioned in literature.

4.3 Outlining an integrated framework for SOA Governance

Harold Koontz and Heinz Weihrich have defined Management as the process of designing and maintaining an environment, [KOON1988]. In the same sense, management

---

26 See Figure 9, A corresponding view of business-driven SOA

can be defined as the process of designing, evaluating, as well as changing a so-called SOA environment. The picture below provides the foundational framework for managing such environment.
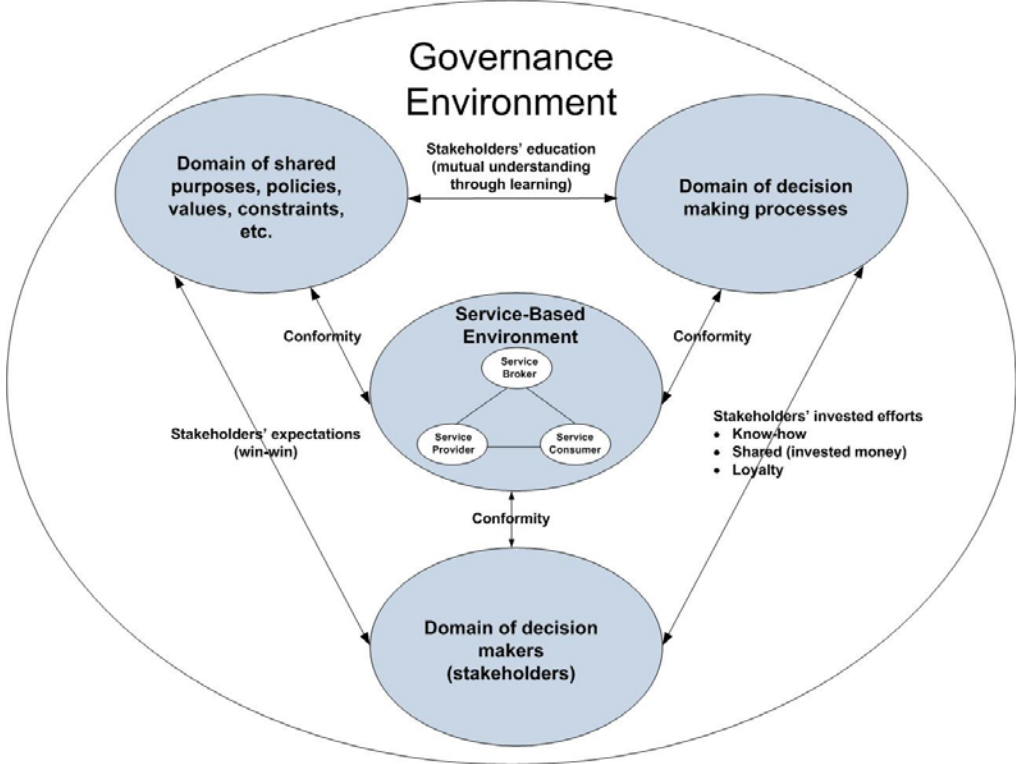
Figure 16: The integrated model of SOA Governance

However, there is a significant difference between the definition provided by Koontz and Weihrich, [KOON1988] and the definition stated in this work. Accordingly, in this study the aims that govern the activities of management deal with the requisite of attractiveness whereas in the work of Koontz and Weihrich the aims concern individuals that are working together in groups in order to accomplish efficiently selected aims. Thus, in our view, the concept of management refers to the process of creating (shaping, reshaping, evaluating, maintaining, etc.) a service-based business environment that holds together service providers, service consumers, and service brokers.

4.3.1 The basic building block of the framework

The integrated model of SOA Governance consists of four basic building blocks, namely:

(1) The architected service-based business environment. This can be public or business
(2) The domain of decision makers
(3) The domain of shared purposes, shared policies, shared values, shared world views, shared constraints, etc. and lastly,
(4) The domain of decision making processes

34

The proposed model is the result of integrating two foundational views of SOA paradigm, namely: (1) the dominated view that is provided by the models of an architected service-based business environment, and (2) the dominated views that are provided by the models of SOA Governance.

4.3.2 *Basic managerial relationship*

Thanos Magoulas and Kalevi Pessi have defined the activities and interest of management in the following terms (1) issues of architectural morphology[27], (2) issues of axiology[28], and (3) issues of praxeology[29], [MAGO1998]. These three concepts refer to the corresponding fundamental relationships that are established by three significant decisions:

- How people or organizations i.e. communities of people, shape their service-based business environments. This issue deals with the architectural configuration of a service-based business environment. Accordingly, issues of configuration belong to the domain of morphology, i.e. the theory of forms, shapes, arrangements, organizations, etc. of firms, institutions, and environments in general. Accordingly, every environment social or physical is the space upon which communication, inter-operation, collaboration, cooperation, and economic transaction take place. Such space can be given in terms of relationship between the domain of providers, the domain of consumers as well as the domain of broker.

.

- How people or organizations evaluate their service-based business environments. This issue concerns just the judgments of either the strategic or operational fit. The issues of fitness belong to the domain of axiology, i.e. the theory of economic, socio-cultural, personal, ethical, values[30]. For instance, in this work, we always refer to an attractive service-based business environment. However, what is the substance of this attractiveness and for whom? According to the discipline of social system design, attractiveness is something defined, negotiated and established by the acceptance of stakeholders. Thus, the real architect of service-based business environment is neither IBM nor Oracle nor any other consulting firms but the stakeholders[31].

- How people or organizations change their service-based business environment in order to improve their attractiveness. The issues related with the changes in human environment belong to the domain of praxeology,

---

[27] Theory of forms i.e. physical, social, functional, social-cultural, hierarchical, network informal, etc. It is explained in terms of relationship between two or more entities.

[28] Theory of values i.e. economic values, personal values, social values, cultural values, functional values, etc. It is stated that everything must be evaluated by the use of criteria; *systemic desirability* (rational view) and *cultural feasibility.* The activity is considered good or bad with respect to the degree that it satisfies these two criteria, [CHEC1985].

[29] Theory of action i.e. decision theory, innovation theory, production theory, etc.

[30] See Methodology of value-based management www.valuebasemanagement.net

[31] See Appendix where the consulting firm does not tolerate the stakeholders the business to dictate over them.

theory of human actions and is known nowadays as change management. However, praxeology refers both to the wisdom that is related with any kind of critical decision, as well as the efforts and struggling for excellence.

4.3.2.1 Architecting the Service-based Business Environment

The issue of architectural configuration of a so-called service-based business environment is not identical with the architecture of composition and coordination of a business process, i.e. a composed service that involves several service providers.

The service-based business environment that the process of configuration shapes covers the following domains; (1) the domain of service consumers, (2) The domain of service providers, and (3) The domain of brokers that are involved in the tasks of registry and/or repository.

In the same sense, an architecture of compound services such as a business process cover only the domain of providers that are involved in that process. In the case of several service providers, different coordination philosophies can be elaborated. However, the architecture of a business process is not the same with the architecture of an architected service-base business environment. In the first case, we concern just the organization of a particular domain whereas in the second case, we refer to the established relationships between at least three significant domains. This distinction is necessary in order to indicate that SOA architecture is identical with an enterprise architecture. However, an enterprise architecture is defined with respect to at least five sub-architecture, only one of these represents the business architecture.

Furthermore, the interactions involved in a composite service can be coordinated either through "orchestration" i.e. centralized coordination of efforts, respectively, "choreography" i.e. decentralized coordination of the involved efforts. However, these forms of coordination involve only the domain of service providers and not the other domains. Thus, in the case of JIT[32], the concept and efforts of coordination must cover even the domain of consumers and of broker.

4.3.2.2 Evaluating the Service-based Business Environment

The shape or form of a service-based business environment makes sense because behind such shape, there are always one or more core ideas that represent by the shape. For instance, a hierarchic pattern is a result of reason rather than of chance. The shape holds tightly or loosely SOA architecture. For instance, out of an information economic perspective, the most significant part of a SOA environment is the broker of services. Accordingly, a particular service consumer needs only two messages in order to become informed about the owners of the required services and thereby to coordinate consumer affairs (conduct the proper provider), [MALO1987] and [MALO1987b]. Furthermore, an architecture that dictates the elimination of redundant services decreases the costs of development and simultaneously minimizes the possible "pollution" and inconsistencies derived from similar services. Thus,

---

[32] Just In Time (JIT) means that the decision rights for supplying a certain product or service lies in the hands of providers. For instance, instead of calculation and plan about what kind of product should be purchased, in what quantity, when, where, at what price, etc. The firms outsource these responsibilities to the suppliers. By this way, the suppliers have the information about the daily consumption of consumers' supplies just in time the required quantities.

the operational fitness can be measured and evaluated in terms of the minimal number of messages required to inform a consumer. These measures may be given in terms of cost of services, quality of services, timeline of services, etc.

However, out of a socio-cultural perspective, this kind of operational fitness is not enough. The strategic fit represents just the responsiveness of such environment to the requisites and desirables of consumers. In this sense, the chosen architectural form makes sense in terms of business agility, business-driven reusability, business innovativeness, etc.

Furthermore, a Service Oriented Architecture, is best understood in collaborative environments such as Business-To-Business (B2B), Business-To-Consumers, (B2C), Supply Chain Networks (SCN), Customer Relationships Management (CRM), etc. In all these cases, the contribution of such architected environment is given in win-win terms. We discuss the issues of strategic fit later on.

In the early days of Informatics, Forrester, [FORR1961] had used the concept of attractiveness to denote something desirable, something that motivates the invested efforts of people, something that cover all interests, something that holds people together, etc. We use the same concept to indicate the desire for working together in order to improve the welfare of people in general and of stakeholders of collaborative organizations in particular with the support of technological efforts.

4.3.2.3 Changing the Service-based Business Environment

The concept of management has been defined by Churchman as the art and science of serving humanity, [CHUR1994]. The same concept has defined by Jay Forrester as the art of decision making. Decision is just the process that converts information and knowledge into action, [FORR1961]. Combining these two concepts of management we can say that management is the process that converts knowledge and information into attractive improvements in our social and business life.

However, Hedberg, [HEDB1980], has demonstrated that both knowledge and information are two critical necessity but not sufficient factors for securing attractive improvements. Wise decisions are participative as well as wise decisions are based on the collective knowledge and experiences of stakeholders. Thus, the establishment of an attractive business or public environment presupposes the motivation and acceptance of stakeholders under the condition of win-win expectations. This means that both the strength of motivation, acceptance and commitments are tentative. However, this is a tentative agreement rather than an forever-agreement. Thus, without strong commitment there is no chance for meaningful change. Without meaningful change there is no effect of meaningful improvements, i.e. improvements that satisfy the expectations of stakeholders.

4.4 Foundation of a new model for SOA Governance

The foundation of the purposed model consists of the distilled knowledge of the previous part and concern three aspects; morphological view, axiological view, and praxeological view.

4.4.1 The morphological view of a service-based business environment

- **Who is conceived as the consumer of services in the context of a service-based business environment?**
  The consumer of services can be either an end-consumer or a particular service provider. In the last case the provider is assigned as an intermediate role to support some transactions of the end- consumer.

- **Who is conceived as the provider of services in the context of a service-based business environment?**
  In most cases, the service providers are identical with the owners of services. However, there is a case where these two roles are occupied by different organizations. Such case, a service provider may "borrow" services from an external provider or may collaborate with other service providers. These situations are typical both in B2B contractual agreements and in the case of various forms of strategic alliances.

- **What kinds of services are provided to the consumers by the SOA environment?**
  In general, the available services within a so-called SOA environment belong to the following categories:
  o Informative
  o Transactional
  o Communicative
  o Decisional

  However, in literature there are other kinds of services that have been excluded from the present study. These services belong to the following categories:
  o Web services
  o Infrastructural services
  o Services dealing with the issues of security, identity, authorization, etc.

- **In what terms can we describe and understand the roles and responsibilities of service broker (LIBRIS)?**
  According to theory the roles and responsibilities of a "Pure" broker of services can be given in the following terms:
  o Common routine for the registration of services
  o Common conceptual scheme for the organization of a catalogue of services
  o Common catalogue with all registered services
  o Events dealing with the registration of services that can be provided by a particular service provider
  o Accessibility of the catalogue by the consumers of services

- **How are the various forms of business transactions contacted between service providers, service consumers and service broker?**
  The cooperation between service providers and service brokers may be seen as a typical phenomenon in a service-based business environment.

  Firstly, the users contact the service broker in order to find out which provider has available the desired services. Secondly, the service broker informs the

service consumer about the addresses of available providers as well as any useful information in order to interoperate with the providers. Lastly, the service consumer takes initiative and contact the chosen service provider. This contact begins and ends with the logic of a business transaction.

This logic can be formed either by the service provider alone or through the negotiations between the provider and consumer of services. However, in many cases the logic cover event the collaborative efforts of two or several service providers.

- **How is the transaction of acquisition and purchasing, for instance literatures, performed?**
  The concept of SOA refers to a collaborative service environment where every service provider is the owner of the provided services. However, the critical issue is the number of service providers that become involved in a business transaction.

  Usually the requirements and wants of a consumer are reflected in a business model. It is the business model that specifies what kind of services is required. However, the assignment of more than one provider does not belong to the business of consumers. The consumer contacts and negotiates his business with just one service provider.

  Lastly, in a service-based business environment, the consumers "borrow" rather than buy a particular service.

4.4.2 The axiological view of a service-based business environment

- **What are the significant events/arguments in the rise and development of a collaborative service-based business environment? (In other words, what is ultimate objective, vision or core idea, of collaboration between the academic libraries of Sweden?)**
  There is not just one core idea that explains the rise, as well as, the business value of SOA. Firstly, the existence of SOA can be given in terms of better alignment between business demands and integrated efforts such as Information Technological efforts and Human efforts. Secondly, the rise of SOA can be given in terms of cost saving through reusable services. Thirdly, the requisite for alignment presupposes the satisfaction of business agility in general and the responsiveness to ever changing customer's needs, wants, expectations, etc.

- **What are the most significant and critical factors for a successful service-based business environment with respect to the needs and expectations of direct stakeholders (consumers, actors, owners) and indirect stakeholders (local community and society in general)?**
  o **Service consumers**
    - Quality of service (value added service, explained in terms of relevancy, reliability, taste, etc.)
    - Minimal efforts
    - Responsiveness
    - Warm social environment

- o **Service providers**
  - Become efficient and effective (conformity between IT effort and required business service)
  - Having continuous contact with other providers
  - Harmonious collaboration (high inter-operability)
  - Freedom of action and independence

- **In what terms can we discuss and judge the expected, respectively real contribution of Service Broker (LIBRIS)?**
  - o The real value of broker can be given in terms of easy coordination of transactions

  - o Accordingly, the effort required to find a relevance service provider is minimal (see Theory of Transaction Cost Analysis, TCA)

  - o The value of service broker is very high if the provided information is up-to-date, comprehensive, and easy to understand

- **What are the most crucial constraints upon which the whole concept of collaboration is based?**
  - o **Ownership of services**
    A composite service such as a business process involves several owners. It follows that the success of such services depends on a group of different owners and their motivational and ethical investments

  - o **IT-based and Human- based efforts**
    Most of the SOA literature refers to IT-based efforts. However, there are cases that the literature refers even to Human-based efforts. Accordingly, a particular business process can indicate and use both kinds of efforts

  - o **Coordination philosophy**
    If the SOA environment has only one owner, then the issues of coordination is not a problem. Either the mode of "orchristration" or the corresponding "choreography" can easily arrange. However in a network that involves several owners the issues of coordination are not explicitly discussed

  - o **The language of Business**
    The SOA environment should be described in the language of business

- **What are the most crucial constraints upon which the whole concept of collaboration is based?**
  In a collaborative environment, the value of a provider can be given in terms of the participation in different service transactions. However, presupposes some form of reward is offered in order to maintain the motivation of the providers to continue the collaboration.

4.4.3 The praxeological view of a service-based business environment

- **Who are the real owners of the collaborative environment? (Who has the right to close down the collaborative service enterprise?)**
  Any form of collaboration between several service providers presupposes the establishment of some negotiated agreement. This agreement may be a contractual arrangement or some form of coalition, i.e. strategic alliance between two or several partners. In both cases the "owner" of a collaborative environment is regulated either by the contract or by the constitution of the concerned alliance.

- **How the rights of decisions have been distributed between the stakeholders of a collaborative environment?**
  In the case of contractual arrangements the decision rights may concern only the involvement of service providers in the various forms of business transactions. However, in the case of partnership and strategic alliances, the involvements of members cover both operational and strategic issues and therefore decision rights.

  The usual strategy or pattern for distributing the decision rights within an alliance is that of federation. Accordingly, strategic issues are treated globally by the leaders of coalition whereas any other operational decision belongs to the realm of a particular partner's enterprise. Lastly, federation means global thinking, local action.

- **What are the most critical strategic decisions in a collaborating environment?**
  A strategic alliance is a business arrangement in the form of a network of participating and cooperating organizations for their mutual benefit. The most crucial decisions in such alliance deal with:

  - The issues of independence

  - The issues of symmetric or proportional benefits

  - The evaluation over the assigned performance

  - The contribution of the member organizations to the strategic areas of the development of alliance, (innovation, competence, technology, etc)

  - The joint development of a sound architecture that uses Information Technology and makes the service environment attractive for all stakeholders environment.

  Members of the organizations may coordinate their efforts, resources, management philosophy, etc. for a variety of purposes.

- **Who is responsible to judge issues of compliance (harmony with established constraints, laws, routines etc. decided either by the community of the collaborative library environment or by the society in general)?**
  Any form of compliance concerns firstly the behaviour of the strategic alliance as a whole. In the same sense, compliance deals even with the behaviour of the participating members organizations.

  In both cases, it is expected that the constitutional agreement to make explicitly the obligations of the whole and its parts. The rules of compliance may concern the relationships between service providers and service customers.

- **Who define the policies to be followed (either through enforcement or through empowerment) by the members of a collaborative environment?**
  According to our theory and the support of many sources, an attractively architected environment presupposes effective *federated SOA governance*. Thus, the federated SOA governance has the rights to establish a sound framework to be followed by any interested party. The effectiveness of such governance is derived from its continuous efforts to hold the conformity between the architected environment and

  - The stated intentions, i.e. purposes, policies, principles, constraints, etc.

  - The processes, practices and procedures, etc. that successively implement the stated intentions

  - The wised decisions of stakeholders that convert the core ideas of SOA into reality

  The establishment of a so-called federated registry approach of SOA supports the coordination of any effort and allows maximum flexibility to the participated member organizations to take their local decisions.

- **Who define policies that deal with the expected quality of service as well as the accessibility and availability of services?**
  In a federated collaborative environment the authorities and responsibilities for the definition of the quality of service is in the hands of both the leading group of the alliance as a whole and the members of participating organizations.

  However, the concept of quality in general and quality of service in particular lack a uniform definition. In any case such quality can be understood in the following terms:

  - **Socio-cultural quality**
    Customer satisfaction = effectiveness = conformity between expected performance and real performance, equality = social symmetry, etc.

- o **Process related quality (service quality)**
  Responsiveness, efficiency, functionality, continuity, accessibility, simplicity, functional interoperability, co-operability, mass customization, economy, consistency, etc.

- o **Infological quality**
  Cognitive conformity, relevancy, simplicity, availability, cognitive interoperability

- o **Structural quality**
  Trustworthiness, legitimacy, sense of security, sense of balanced between order and freedom, etc.

- o **Contextual quality**
  Sense of resilience, innovativeness, adaptability, business agility, etc.

- **What kinds of decisions are global i.e. decisions that are related with the future development of collaboration?**
  As we have stated in (P03) the global critical decisions dealing with the future of collaboration focus and cover the following issues:

  - o The independence of participating organizations
  - o The symmetric or proportional benefits
  - o The evaluation over the assigned performance
  - o The contribution of the member organizations to the strategic areas of the development of alliance (innovation, competence, technology, etc.)
  - o The joint development of a sound responsive architecture
  - o Etc.

- **How dependent are the development of service providers to other providers?**
  Each service provider is an autonomous entity. This means that every decision concerning the development of a service provider is taken locally rather than collaboratively. However, in a federative environment, -an environment formed by contractual or relational arrangements of two or more organizations -, the development of a federated registry unit has significant consequences with all parties.

- **How the future development of service broker affects the development of service providers?**
  The federative registry unit, i.e. service broker, is the most significant part of the architected collaborative environment. However, behind every architecture lies a core idea that shape the structures, relationships and behaviour of processes, systems, people, etc.

  In the context of SOA, there are several such core ideas, namely; IT alignment with business, business agility, asset reusability, etc.

  IT alignment with business is a presupposition for any other core idea. This form of alignment expresses the dominance of business over technology.

Therefore, any attempt to put the enterprise into the "enterprise system" leads to misalignment.

However, Business – IT alignment does not take into account the costs of ad-hoc integration or the requisites of business agility. Therefore, the requisite of business agility provides the requisites of integration and interoperability though SOA pattern that establish loosely coupling integration, message based interoperability, and business based co-operability.

However, the satisfaction of requisites of reusability creates another pattern that increases the technological and developmental agility as well as the significant economies of IT.

Reusability means that required services may be available through (1) Legacy systems (2) External service providers such as IBM, Microsoft, Oracle, etc. (3) In-house development, etc. In any case this architectural form of SOA has been criticized.

- **How the performance of the collaborative environment can be measured?**
  The performance of a collaborative environment can be measured in several ways.

  Strategically, the value of the collaboration can be given in terms of alliance continuity, members' independence, symmetric/proportional[33] benefits, business and IT-alignment, etc.

  Operationally, the value of collaboration can be given in terms business agility, asset reusability, customers satisfaction, etc.

  In any case, the value of such environment depends on the efforts of federal registry[34] to be consistent, simple, accessible, available, etc. In the same sense, the value of a collaborative environment can be given in terms of empowerment rather than enforcement of the know-how competencies of people.

4.5 Summary

Scope: The value of SOA depends on the scope of the reference environment. The larger the environment is, the larger and meaningful the provided value of SOA. Reference environment can be:

1. One department consisting of many different business unit distributing in several geographical area

---

[33] Symmetric means equal benefit for everyone. Meanwhile proportional is a form of symmetric, however, this asymmetric can be explained by different amount of effort that has been provided.

[34] The concept of federation expresses the common interest of stakeholders to have the function of broker together. Federation represents the situation where several organizations either in form of alliance, contractual or partnership decides some form of coordination.

2. The whole organization i.e. purchase department, design department, inventory department, logistic department, sales department, top management, etc.
3. Relationships between organizations. This can be transactional or relational.

Human-centric view of governance: In the proposed model, the most significant characteristic of SOA governance is its human-centric view of architecture. This means changes in the values as well as significant changes in the logic of decisions must negotiate and accepted through a commitment between the stakeholders, otherwise, neither SOA nor any other architectures hold. The clearest case of human-centric view of service-based business environment is the case of consulting firms. In such case, the consumers require support, for instance organizational analysis, environmental analysis, architectural design, improvement of performance, etc. The providers are various kinds of consulting firms that through the use of documented existing experience and best practices, they build a collaborative network that has the characteristic of SOA. The success of such network depends on the empowerment and know-how of all parties, in other words, consulting agent and consumers representatives, [DAVE1997].

SOA Governance: As we have documented, SOA Governance is about sound and wise community of decision make (stakeholders). SOA Governance also is about service quality management as well as change management. However, what happen when SOA environment lack of an explicitly formal SOA Governance? Accordingly, the cost of an ungoverned SOA is (1) lack of reuse, (2) disruption and failure of business process, (3) escalation of support costs resulting from service outages, (4) security breaches, and (5) non-compliance with enterprise or government regulations.

Architectural Attractiveness: Accordingly, SOA Governance is a necessary condition rather than an option, for an attractive service-based business environment. Therefore, according to our interpretation of SOA and SOA Governance literatures, the primary objective or propose of SOA Governance is to maximize quality and trust within the service network that should make benefit and satisfaction to consumer. In the same way, SOA Governance must be aware and evaluate the conformity and consistency between (1) the architected service oriented environment, and the proposed policies and constraints, (2) the architected service oriented environment, and process routines and practices involved in managing changes in the architected environment, (3) conformity and consistency between the informal stakeholders' intention, and the architected environment.

Change Management and Compliance: The management of a dynamic heterogeneous and complex service-based business environment is complicated enterprise. We must understand that policies, goals, and constraints are always in question because their relevance and adequacy depend on the conformity and consistency with the architected environment. For instance, consider a situation where the regulation changes, how this change affects the policies and what kinds of changes are involved, how fast these changes should be implemented, etc.

Accordingly, a good service oriented architecture is an architecture that conforms the ever changing nature of expectations of direct and indirect stakeholders. These expectations are related to, firstly, the policies, proposes, principles, constraints, etc. that have been established by the stakeholders, and secondly, the activities that shape, evaluate, and change

the so-called service-based business environment. In summary, architecture is a well-organized and attractive reality aiming to support equally the expectations of the stakeholders. This attractiveness is understood and judged in terms of Strategic Alignment, Operational Alignment, Infological Alignment, and Motivational Alignment. In the same sense, the Strategic Governance is responsible for the Strategic and Motivational Alignment, whereas, the Operational Alignment has the responsibility for Operational and Infological (Cognitive) Alignment [See Figure 17].
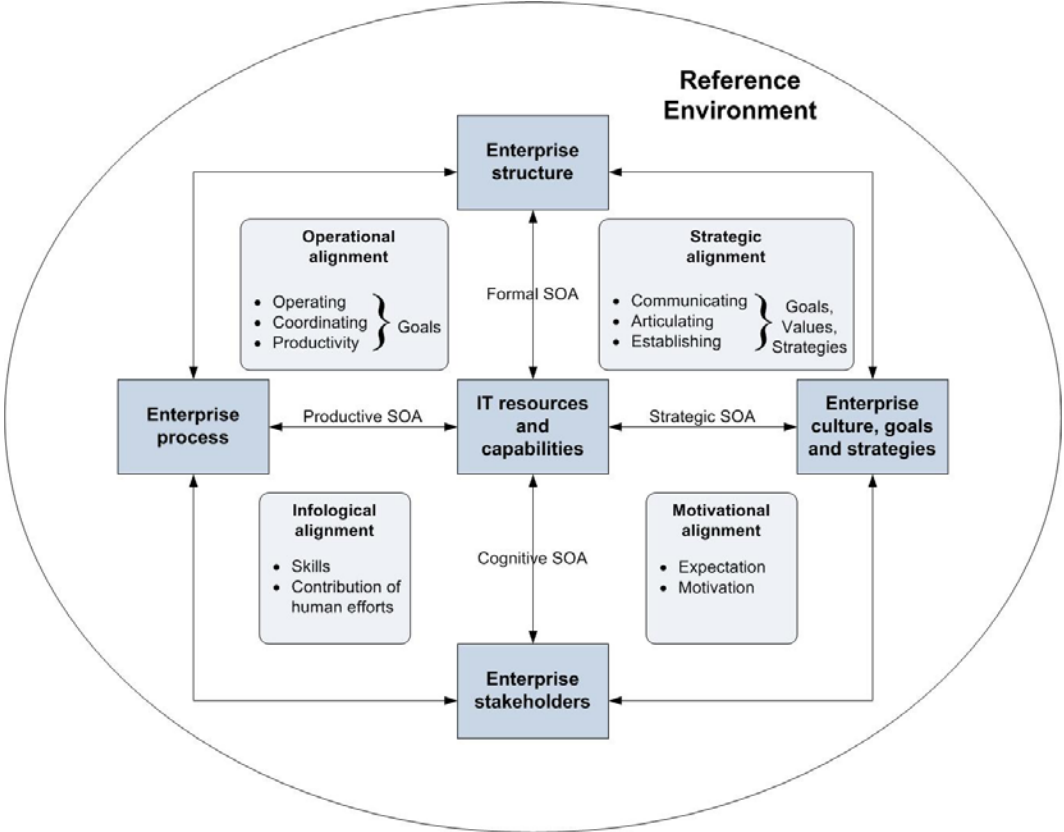
Figure 17: A validated view of SOA Governance in accordance to FEM Model

5. Empirical application of the model: LIBRIS Case Study

In this section we try to describe empirically of an interesting and attractive service environment that represents the collaborative effort of academic libraries in Sweden. In this context, LIBRIS is a so-called service provider that make the whole concept of SOA interesting not because the effect of information economics but because all interesting groups of stakeholders (consumers, employers, owners, etc.).

5.1 The LIBRIS environment

In order to demonstrate SOA concept for the better understanding, LIBRIS[35] is chosen as the example. LIBRIS is a national library system managed by Royal Library of Sweden in Stockholm, however, the owners of the system are all the collaborative parties. The main objective of LIBRIS is to support customers of any kinds such as students, teachers, researchers, industries, other libraries, etc. to locate a particular book or article because of the enormous variability and instability of customer requirement; this requisite is impossible to satisfy only by a particular local library. Therefore, the collaborative efforts of academic libraries in Sweden provide the best solution to the ever changing and variable customer needs. Accordingly, LIBRIS provides global references catalogue. By this way, any person or institutions or organizations can locate where a particular report is available and by this way, take the necessary action to loan a particular literatures.

In Figure 18, as we demonstrate out of information economic perspective, an environment without LIBRIS means 2*P messages (P = the number of provider in that environment). The same environment with LIBRIS, the effort of consumer should always equal to 2 messages (request message and respond message).

The academic environment before the rise of LIBRIS, in this case, the service environment operated without the support of service broker (Therefore, a coordination of a transaction require always 2 * the number of known provider)

The academic environment after the rise of LIBRIS, in this case, the service environment is supported by LIBRIS. (Therefore, there are always 2 messages to coordinate a transaction; request and response)

Figure 18: Two views of architectural patterns treated by information economics

---

[35] The National Catalogue of Swedish Library. In the beginning LIBRIS was manual functional through catalogue of literature such as author catalogue and subject catalogue. Later on, computerization took place and therefore both the efficiency and the accessibility of LIBRIS were improved radically.

More concrete, LIBRIS provides bibliographic services such as search facilities, cataloguing and support interlibrary lending. One of the main functions of LIBRIS is as a device assisting rationalization processes related to its library members and to profile, on a nationwide basis, the holdings of research libraries. LIBRIS offers the public free access to over 5 million titles held at Swedish libraries. There are at present approximately 300 libraries in Sweden who register their holdings in LIBRIS. The majorities of these are academic and research libraries. About 1,200 Swedish and 200 libraries mainly from the other Nordic countries use LIBRIS for inter-library loans, [LIBR2008]. In this case, LIBRIS represents global organization where local libraries act as individual organizations connecting to LIBRIS in order to process the particular library service.

To illustrate the acquisition and loan process, we use the following architecture pattern to demonstrate the real meaning of SOA. These are (1) the consumers, (2) the service providers, and lastly (3) the service broker.
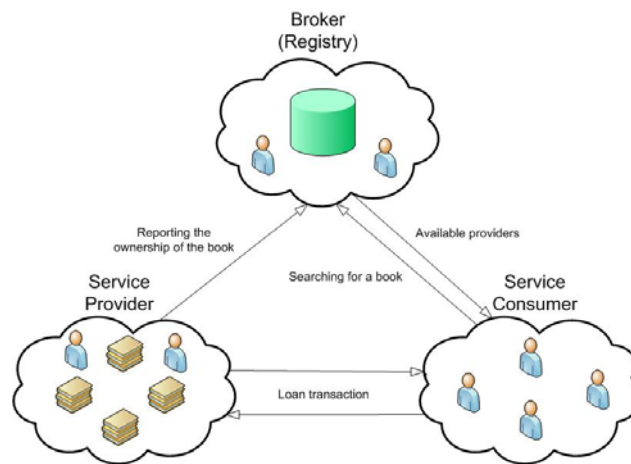
Figure 19: LIBRIS loan process

Figure 19 shows the relationships and connections between the following parties:

- The Broker i.e. LIBRIS System,
- The Service Providers, i.e. collaborating libraries in Sweden,
- The Service Customers, i.e. students, lecturers, researchers, etc.

Initially, the customer can conduct LIBRIS directly or indirectly through the support of service provider.

In the same way, when there is a donation or process of a new book to be registered in the library, the provider of this book will report the ownership to LIBRIS. Once the book reference is found in LIBRIS, the system will send acknowledgement to customer for the location of the book. Lastly, when the location of the book is identified, loan transaction is processed between the library and the customer.

In summary, the big idea with the LIBRIS environment can be given in terms of firstly customer satisfaction, secondly no extra effort for provider because any local transaction about donation and purchase is reported automatically to LIBRIS, and lastly, the information

economies resulted from this architecture (further explanation is explained below in relation to SOA governance).

5.2   The governance of LIBRIS

Now in this LIBRIS environment, we can identify the most crucial issues of SOA governance.

- Firstly, SOA governance concerns the kind of decision the collaborating parties usually take such as purchase of new literatures, cataloguing of new literatures, etc. (The issue of subject matters of SOA).

- Secondly, who participate and make these decisions. Accordingly, all libraries participate in the case of cataloguing decision concerning a few critical attributes such as ISBN, title, language, author, editor, edition, edition year, etc. whereas any purchase transaction that is relevant to the update of LIBRIS is performed locally and the same information is transmitted to LIBRIS. Furthermore, the collaboration concerns only the cataloguing of the literature and nothing more. In this way, the classification and the localization of the particular book is the local affair. In summary, LIBRIS does not constraint the autonomous character[36] of the local libraries and this is one of the main point of SOA governance and this must be seen as the main point of SOA governance because collaboration does not limit the authority and freedom of local libraries (The issue of participation and the amount of freedom).

- Lastly, SOA governance concerns even the issue of evaluation the effect of the above decision as well as their improvement in the case where the response to the requisite of stakeholders is not satisfied. Accordingly, we think that the effect is those related with information economies. This effect can be measure in terms of the effort that customers invest in time and money in order to find out where a particular book or report exists. Suppose that we have as much as 1,200 libraries and a customer tries to find out where a particular exist then it must ask 1,200 * 2 messages through the LIBRIS. The required messages to satisfy the customers are always two, [MALONE 1986]. Furthermore, information economies mean that the LIBRIS creates the virtual environment of global libraries where not two libraries need to own the same book. By this way, libraries can utilize the money to buy new literature rather than to buy the same literature. Lastly, the labor cost is zero because the local registration of any transaction is reported automatically to LIBRIS. Accordingly, prior to any purchase, libraries go and ask LIBRIS if particular book exists or not in order to invest their money to acquire new literature. Thus, all involved parties are very satisfied in such attractive solution. (The issue of measurement).

---

[36] LIBRIS collaboration does not mean that all decision should be taken by all parties. In our case, the character of these collaboration promotes the information economies without to restrict or limit the freedom of the parties to decide with respect to their local issues, for instance, in collaboration there is a promise to utilize the money in the process of new literature, however, if the local demand are too high, any libraries can decide to buy book that ready exist in the LIBRIS environment. Thus, no libraries can dictate over the other. No one library can have improvement in their services on the cost of the others. In this sense, mutual effort results in mutual benefit.

In summary, LIBRIS environment provides an illustrative case of both SOA and SOA governance. In the later case, such governance is given in the following term:

- The issue of subject matters of SOA i.e. what kinds of decision are taken by all parties and what kind of decision are locally.
- The issue of participation and the amount of freedom i.e. who participate in the decision
- The issue of measurement i.e. how we measure the effect of collaboration

5.3 A rejected proposal for the further development of LIBRIS

The study of Mats G. Lindquist [LIND2006] provides an interesting answer for the further development of LIBRIS. Accordingly, the issue of this study was to provide a clear view about the degree that libraries can reduce their cost through the definition of LIBRIS concept and the collaboration between LIBRIS and local library system. This reorganization means that local function transfers to LIBRIS by this way the functionality of LIBRIS is given not only in terms of a broker but answer in terms of loan routines that today are operated by the local system. Thus, the functionality of local system is coming to be decreased whereas at the same time the functionality of LIBRIS should be increased.

Accordingly, there are some argument for supporting this reorganization and some other arguments that are against the whole changes. In the case of positive argument, Linquist states that systems became older and therefore must be changed to other planed conditions. Furthermore, the cost for such a development shall be distributed to all participants rather than to one (in this case, the Kungliga Biblioteket). Lastly, we have became dependent of local solution such as (1) databases, (2) loan routine, (3) network, (4) links between network and local system, etc,

However, the participants do not like the proposed solution and they support the existing one saying that *'Varför byta model, vi har det bra'*. (Why shall we change our current model, we are satisfied of it).

According to Lindquist, the proposed solution means a migration from a pattern with central broker to a pattern where LIBRIS undertake function that today are locally. By this way, the role of LIBRIS as broker is changed and becomes provider.

In summary, the collaborating parties have decided that the proposed pattern of LIBRIS environment is not desired for the near future. Perhaps when other new solutions become attractive for LIBRIS environment, we can discuss the issue of reorganization. However, there is no more question today to change what is operating so good for all providers and all customers.

In any case, there are some points that are unclearly stated in Lindquist's report. These are the following:

- Firstly, the concept of system is meaningless because it is lack of context. In other word, what Lindquist means the system is a case of hardware rather than a case of enterprise operation. Any change of hardware in theory is

independent of both the local and global models of business services, [ANSI1975][37]

- Secondly, the routines and functions for loan service have nothing to do with the role of the broker but with the local libraries. Changing the current pattern of Planetary System, the sun takes even the role of a planet and this is according to our opinion, an inconsistent system because the two conflicting roles of the same unit (in this case LIBRIS).

- Thirdly, out of the management and governance perspectives, a solution can consume more resources if it promotes the comprehensibility, understandability, and attractiveness of the concerned environment. According to our opinion, the current LIBRIS environment satisfied the above three main principles of management and governance.

  o Firstly, the desirability of comprehensivablity, in other word is satisfied through the dependency between LIBRIS and local system are comprehensive (message based) and therefore manageable.
  o Secondly, everyone understands that in the environment, LIBRIS satisfies only the need of where a particular book or report is located. Furthermore, everyone understands that LIBRIS does not limit either local power or freedom. In this way, all people working in the different libraries form their routine and system in accordance to the local tradition, language, power, world views, etc.
  o Thirdly, the current pattern of LIBRIS is meaningful for all collaborating parties as well as for the employees and for the customers. Therefore, the current model is judged as very attractive. All stakeholders (owners, employees, and customers) are satisfied. With various arguments; (1) the owners, because they pay the same money for a global attractive service, (2) the customers, because they have maximum possible services, (3) the employees, because they does not invest some extra effort in the whole loan transaction. Furthermore, the culture, language, power, world views, etc. remain local. Lastly, the whole concept is attractive because they give the opportunity to the local stakeholders to save the local environment in accordance with the local know-how and local tradition.

5.4 A comparative view of the current and proposed pattern of SOA and SOA governance

According to our interpretation of LIBRIS environment, there are two main approaches in the line of today's governance. The first one is the current approach supporting the customer oriented solution where the role of LIBRIS is also a broker. In the next approach is that of documented by Lindquist dealing with the reorganization of LIBRIS environment because some routines and functions are centralized and transferred to LIBRIS.

---

[37] The ANSI-SPARC Architecture (American National Standards Institute, Standards Planning And Requirements Committee) is an abstract design standard for a Database Management System (DBMS), first proposed in 1975. Most modern commercial DBMS are based on this system. The ANSI-SPARC model never became a formal standard.

In summary, the first approach aims to serve the customer ever changing need through agility and responsiveness whereas the next approach aims to reduce the cost of developments and operations through reusability.

The most characteristic differences between the current and proposed pattern of governance are demonstrated below.

Primary Governance issue	Current pattern of SOA Governance	Proposed pattern of SOA Governance
What kind of decisions belongs to the collaboration of parties?	Most global decisions deal with the issues of Registry Broker and its functions. i.e. LIBRIS System. E.g. • Principles for coordinating the purchase of new literature • Principles for categorization of global information • Principles for global identification , e.g. ISBN • Principles for routines and rules dealing with the activities of registry, • Principles for message (event) driven communication	Global decisions deals with the issues of Broker and its functions. i.e. LIBRIS System. e.g. • Principles of purchasing new literature (+ eBooks) • Principles for categorization of global information • Principles for the classification of literature • Principles for global identification , e.g. ISBN • Principles dealing with the issues of global loan • Principles for global self-service • Principles for routines and rules dealing with the activities of registry, • Principles for repository driven communication • Principles for reusable systems design, systems development and systems maintenance

Primary Governance issue	Current pattern of SOA Governance	Proposed pattern of SOA Governance
What kinds of decisions are taken locally?	Local decisions deals with the development of local systems and their functions in accordance with local requirements • Principles for the classification of literature • Principles for the physical localization of literature • Principles dealing with the issues of local and global loan • Principles dealing with the issues of education of "users" • Principles for systems design, systems development and systems	Local decisions deals with the development of local systems and their functions in accordance with local requirements • Principles for the physical localization of literature • Principles dealing with the issues of local loan • Principles dealing with the issues of education of "users"

Primary Governance issue	Current pattern of SOA Governance	Proposed pattern of SOA Governance
	maintenance • Principles for desk service, i.e. personnel contact	

Primary Governance issue	Current pattern of SOA Governance	Proposed pattern of SOA Governance
Who participate and make these decisions?	Representatives from all collaborating libraries	CEO, CIO, CTO, and other IT experts
How do the parties evaluate the effects of these decisions i.e. in what terms do the parties evaluate the effects of collaboration?	• Global responsiveness • Agility • Relevancy • Local freedom for systems design, development • Locally driven operations	• Global cost effectiveness • Location effectiveness due to eLibrary • Redundancy-free effectiveness through the use of central repository • Effectiveness of Centrally driven systems design and development • Effectiveness of centrally driven operations

In summary, the current pattern of SOA Governance (distributed and autonomous organizational unit under the governance of representative from that unit) follows the classical principles of business driven enterprise and the schools of H. Simon [SIMO1969], M-Å Hugoson [HUGO1991], G. Galbraith [GALB1977] etc. However, the proposed pattern of SOA where everything is located in one single place and everything is under the control of one single authority was rejected. The proposed pattern of SOA Governance follows the technological interpretation of a so-called IT-monarchy that we have mentioned earlier.

Both patterns (decentralize respectively centralize) are associated with advantages and disadvantage. However, despite the rejection of the proposed new pattern in LIBRIS case, this study does not follow such hard philosophy. There are always success cases where the balance between agility and reusability as well as centralization and decentralization has been the case. Therefore, such situation where the architectural design should based on the balance between agility and reusability, then some kind of analysis must always provide and follow by clear argument, why reusability should be balanced with the core idea of agility? Only in the case where the analysis provides significant benefit and the absence of any risks, the balance should be accepted.

6. Analytical views of SOA Governance

This section covers the three most significant parts of SOA Governance, namely, how people shape their environment, how people judge the quality of value of the environment, how people change their environment. Accordingly, we analyze firstly, the similarities and differences of theoretical and empirical views of SOA morphology, i.e. architectural configuration of service environment. Secondly, we analyze the similarities and differences of theoretical and empirical views of SOA axiology, i.e. architectural and social values of a service environment. Lastly, we analyze the similarities and differences of theoretical and empirical views of SOA praxeology, i.e. wise management of changes affecting the quality of a SOA environment.

6.1 A morphological view of service-based business environment

SOA Governance may be viewed as wise architecture aiming to shape an attractive architecture that is able to satisfy the expectation of the stakeholders. This architecture is given in terms of architectural morphology. Accordingly, the morphological view of an enterprise environment is given in terms of an enterprise architecture covers the following themes:

- The consumer of services
- The providers of services
- The broker of services
- The service categories
- The logic of business transactions
- The acquisition and purchase of assets

The following tables provide fruitful information between the theoretical and empirical views of the above themes as well as their similarities and differences.

Theme: The Consumer of Services		
Issue (M01): Who is conceived as the consumer of services in the context of a service-based business environment?		
Theoretical views	**Empirical views**	**Similarities and differences**
The consumer of services can be either an end-consumer or a particular service provider. In the last case, the provider is assigned as an intermediate role[38] to support some transactions of the end-consumer.	The end-consumers of library services can belong to different groups such as: - Students - Researchers - Teachers - Local communities In the case of knowledge acquisition as well as in the case of inter-library transactions, all these people can be informed by LIBRIS.	As we can see, there is a strong agreement with respect to the relationships between the broker and the end-consumers. However, local service providers may have the responsibility to coordinate a transaction and therefore it can play an intermediate role. In some literature even service providers are conceived as consumers of services.

---

[38] Intermediate role can be conceived as an agent or middle entity involved in a transaction between broker and end-consumers in order to support the communication between these two entities.

	However, transaction interactions are treated through the collaboration of local providers.	

Theme: The Providers of Services

Issue (M02): Who is conceived as the provider of services in the context of a service-based business environment?

Theoretical views	Empirical views	Similarities and differences
In most cases, the service providers are identical with the owners of services. However, there is a case where these two roles are occupied by different organizations. Such case, a service provider may "borrow" services from an external provider or may collaborate with other service providers. These situations are typical both in B2B contractual agreements and in the case of various forms of strategic alliances.	In an academic collaborative environment, we can identify all local libraries as the primary service providers. However, in this case, the service providers are identical with the owners of services. Note that the domain of providers covers all the academic libraries of Sweden. However, every library may consist of several physically distant operating units that can "play" the role of service provider.	As we can read from the theoretical and empirical views, there is a strong agreement with respect to who is conceived as a service provider. However, the local service provider may have the responsibility to collaborate with other service providers and by this way to satisfy the needs and wants of consumers. In this case, the service provider become responsible to coordinate just a transaction but has nothing to do with the ownership of involved services.

Theme: The Service Categories

Issue (M03): What kinds of services are provided to the consumers by the SOA environment?

Theoretical views	Empirical views	Similarities and differences
In general, the available services within a so-called SOA environment belong to the following categories: • Informative • Transactional • Communicative • Decisional However, in literature there are other kinds of services that have been excluded from the present study. These services belong to the following categories: • Web services • Infrastructural services • Services dealing with the issues of security, identity authorization, etc.	• Global and local informative services, e.g. where can I find a particular book? Furthermore, the service providers inform new students on the regulation and services, teach them how to search in library catalogue and other databases • Transactional services such as : (1) Inter-library loan on books (2) Local-library loan on books • Communicative services with respect to the registry issues as well as the issues related with the contractual arrangement between service providers • Decisional services, i.e. services related with the purchase of new literature that are primary dictated by the	As we can observe, there is a strong agreement between the theoretical and empirical views of SOA environment. However the theoretical view contains even services that belong to the infrastructure of the service environment rather than its architecture. In the same sense, we can note that there are other services that are related to human efforts or are just self-services.

	wants of needs of service consumers	
	• Other kinds of services	
	• Courses dealing with the introduction to LIBRIS, 2 hours for new students and 4 hours for master students (how to search in details and how to write reference)	
	• Expert support services, i.e. booking librarians and expert support	
	• Copying machine and scanner services	
	• Services dealing with accessibility of search catalogue on LIBRIS	
	• Services dealing with the use of computer work stations	

Theme: The Broker of Services		
Issues (M04): In what terms can we describe and understand the roles and responsibilities of service broker (LIBRIS)?		
Theoretical views	**Empirical views**	**Similarities and differences**
According to theory, the roles and responsibilities of a "Pure" broker of services can be given in the following terms: • Common routine for the registration of services • Common conceptual scheme for the organization of a catalogue of services • Common catalogue with all registered services • Events dealing with the registration of services that can be provided by a particular service provider • Accessibility of the catalogue by the	The roles and responsibilities of LIBRIS are limited only to the systematization and availability of informational services. These can be given in the following terms: • LIBRIS provides answers with respect to 'if' the book is available and 'where' • Common unified routine, MARC21[39], for the registration of references of the academic literature owned by a particular library. Common conceptual scheme for the organization of the literature catalogue. It consists of a few attributes that are shared by all libraries, i.e. authors, title,	As we can see, there is a good agreement between the theoretical and empirical views with respect to the roles and responsibilities of a broker of services. The roles of a Pure Broker are related with the tasks of registry and the tasks of repository. In the first case the role of a broker is associated with continuous management of a reference catalog of services. As well as the accessibility of the catalogue. In the second case, i.e. the tasks associated with the repository, the role of the broker is to maintain and distribute a common unified scheme for managing the semantic issues that are related with the registration as well as the

---

[39] The MARC21 formats are widely used standards for the representation and exchange of authority, bibliographic, classification, community information, and holdings data in machine-readable form [MARC2006]

	edition, publisher, ISBN, etc.	communication between service consumer and service provider.
consumers of services	• Common catalogue with all registered literature • Events dealing with the registration of references by the efforts of service providers • Accessibility of the catalogue (any time, any place) by the consumers of informational services • Courses are provided for new cataloguers	However, a large volume of literature deals with technical consideration rather than business ones. These have been excluded from this study.

Theme: The logic of business transactions (such as Inter-library Loan)

Issue (M05): How are the various forms of business transactions contacted between service providers, service consumers and service broker?

How is the process of inter-library loan conducted between service providers (local libraries) and service broker (LIBRIS)?

Theoretical views	Empirical views	Similarities and differences
The cooperation between service providers and service brokers may be seen as a typical phenomenon in a service-based business environment. Firstly, the users contact the service broker in order to find out which provider has available the desired services. Secondly, the service broker informs the service consumer about the addresses of available providers as well as any useful information in order to interoperate with the providers. Lastly, the service consumer takes initiative and contact the chosen service provider. This contact begins and ends with the logic of a business transaction. This logic can be formed either by the service provider alone or through the negotiations between the provider and consumer of services. However, in many cases the logic cover even the	Firstly, the users send the request of the book to their own library. The users then fill their details into CHANS or directly to LIBRIS which the selected library will get these details on inter-library loan module. The librarian will receive this information immediately and check with the selected library whether the book is on loan. The user will be informed within a day. If the book is on loan, the user can borrow it after a few weeks, otherwise the book will be delivered the day after or within a week. The period of the loan depends on the owner of the book, maybe two weeks, three months or six months. Usually if there is another reservation queue, it will be two-week-loan. Sometimes the users can loan the book as long as they want until the owner needs it back. The book can also be renewed. Every library has its own rules, that	As we can see there is a strong agreement between the theoretical and the empirical views of transactional logic. The agreement is demonstrated to the fact that the LIBRIS acts as a service broker. The current state of LIBRIS implementation mediates in the bookings and distribution of Consumer Orders. However, these functions do not change the assigned role of service broker.

| collaborative efforts of two or several service providers. | is, local regulations. For example, if the user borrows the book from Stockholm, it will take a few days to deliver.

However, several times during the day, a librarian conduct the LIBRIS to see if there are some consumers' order to be managed and she/he take the initiative to do that (this is the form of choreography because the absence of central coordinating unit). | |

Theme: The Acquisition and Purchase of Assets		
Issue (M06): How is the transaction of acquisition and purchasing, of assets i.e. literatures, performed?		
Theoretical views	**Empirical views**	**Similarities and differences**
The concept of SOA refers to a collaborative service environment where every service provider is the owner of the provided services. However, the critical issue is the number of service providers that become involved in a business transaction. Usually the requirements and wants of a consumer are reflected in a business model. It is the business model that specifies what kind of services is required. However, the assignment of more than one provider does not belong to the business of consumers. The consumer contacts and negotiates his business with just one service provider. Lastly, in a service based environment the consumers "borrow" rather than buy a particular service.	• There are three ways of purchasing literatures as follow: o At least one copy of course book o Look from list of new literatures and see if needed o Recommend from other people • The acquisition librarian is responsible for acquisition and purchase literatures • Currently, the acquisition and purchase of the literatures follow more of the traditional process rather than the obligation • To purchase e-periodical through BIBSAM[40], local libraries join together as an association in order to have more power than the local libraries buy it themselves • Libraries must have at least one course book locally. However, it is possible to purchase many copies as the cost might be cheaper than conduct inter-library loan • When the book is delivered to	In this case, the theoretical and the empirical views are not in agreement. One of the main characteristic of a service based environment is the borrow-like transactions rather than purchasing ones. The whole concept refers to the relationships between providers and consumers of services rather than the business–to-business relationships between intermediates providers. Lastly, in a service based environment the providers are the owners of the borrowed services.

---

[40] The organization for academic libraries under the Royal Library.

	the local libraries, the cataloguer catalogs the book on LIBRIS i.e. title, author, description of the book, ISBN, classification, number of pages, publisher, etc. which are library standard. The day after, the cataloguer imports the book details from LIBRIS to local database and put item barcode in order to link between bibliography and the physical book	

6.2   A axiological view of service-based business environment

The following themes have been defined as representative views of how the stakeholders of a service architected environment, concern the issues of values and the crucial constraints that have an impact on that environment.

- The rise of service oriented environment (Service-based business environment)
- The Critical Success Factors (CSF) for excellent performance
- The expected and real value of Service Providers
- The expected and real value of Service Broker
- The crucial constraints of collaboration

Theme: The rise of Service-oriented environment

Issues (A01): What are the significant events/arguments in the rise and development of a collaborative service-based business environment? (In other words, what is ultimate objective, vision or core idea of collaboration between the academic libraries of Sweden)

Theoretical views	Empirical views	Similarities and differences
There is not just one core idea that explains the rise, as well as, the business value of SOA. Firstly, the existence of SOA can be given in terms of better alignment between business demands and integrated efforts such as information technological efforts and human efforts. Secondly, the rise of SOA can be given in terms of cost saving through reusable services. Thirdly, the requisite for alignment presupposes the satisfaction of business agility in general and the responsiveness to ever changing customer's needs, wants, expectations, etc.	The rise of the collaborative environment of academic libraries in Sweden began on 1975. The core idea behind the establishment of the so-called Service Architecture was the creation of a virtual library in order to response to the ever changing needs, requirements and wants of students, researchers, teachers, and society in general. However, another argument was the cost saving goal through "reusable" literature and the use of these savings in the purchase of new literature.	There is a significant similarity between the theoretical and empirical views of SOA. This can be given in terms of business or organizational agility with respect to responsiveness of customer focused services. However, the second goal dealing with cost saving, refers to different sources of arguments. (1) The case of saving through reusable services is a technology-driven argument. (2) The case of saving through reusable literature is an enterprise-driven argument. A significant difference is that the scope of services in the domain of empirical study is included in the scope of services in the domain of theoretical view. This was expected because the scope of the service

		categories in the domain of theoretical view must be the broadest and richer one.

Theme: The CSF[41] for excellent performance

Issues (A02): What are the most significant and critical factors for a successful service-based business environment with respect to the needs and expectations of direct stakeholders (consumers, actors, owners) and indirect stakeholders (local community and society in general)?

Theoretical views	Empirical views	Similarities and differences

• **Service consumers**
  o Quality of service (value added service, explained in terms of relevancy, reliability, taste, etc.)
  o Minimal efforts
  o Responsiveness
  o Warm social environment
 	
• **Service providers**
  o Become efficient and effective (conformity between IT effort and required business service)
  o Having continuous contact with other providers
  o Harmonious collaboration (High inter-operability)
  o Freedom of action and independence
	• **Service consumers**
	  o Good service
	  o Quick response for researchers
	  o Warm environment for student especially when writing master thesis
	
	• **Service providers**
	  o Easily contact with other libraries that are outside the collision of Swedish academic libraries
	  o Communication with users with respect to satisfy both technical and operative needs, for example, to improve search keywords
	  o Global access to every big libraries in the world through the use of search engine
	  o High degree of independent between local and global tasks, for instance, cataloguing maybe seen as a global task that aims to have registry up-to-date. However, classification is a local task in some libraries, this means that some libraries have their own classification scheme, i.e. SAB, UDC, Dewey, LC, etc.)
	
	• **Owners (The states of Sweden)**
	  o One library for each university
	  o Inter-library loan
	  o Search database
		• **Service consumers**
		In general, there is a good agreement between the theoretical and empirical views of the architected service-based business environment and by this way, we believe that the consumer of the library must be satisfied.
		
		However, the best way to be sure about this is to undertake a special survey. If this is done, it must be focus on just the quality of service. Accordingly, if something can be improved, it is just the different views of the service quality.
		
		• **Service providers**
		There is a relatively high agreement between the theoretical and empirical views of service-based business environment because the architecture satisfies the expectation of service providers in all involved dimension. As we have demonstrated, the core concept of the empirical environment is business agility i.e. high responsiveness.
		
		• **Owners**
		According to our interpretation, the owner must be very satisfied of the real affect of the collaborative environment of libraries.
		
		Accordingly, owner invested

41 Critical Success Factor

	Investment, according to the state, the number of librarian should be decreased because the money is now spent on electronic resources and journals which are expensive. The state thinks that it is better for librarians to educate more to the users	resources in the enterprise of academic libraries results to the huge effect. This effect can be measured in terms of number of local and global loans. Another factor that can be judged can be given in terms of the contribution of the libraries to the knowledge development of Swedish society in general.

Theme: The expected and real value of Service Broker

Issues (A03): In what terms can we discuss and judge the expected, respectively real contribution of Service Broker (LIBRIS)?

Theoretical views	Empirical views	Similarities and differences
• The real value of broker can be given in terms of easy coordination of transactions • Accordingly, the effort required to find a relevance service provider is minimal (see Theory of Transaction Cost Analysis, TCA) • The value of service broker is very high if the provided information is up-to-date, comprehensive, and easy to understand.	• The primary value of LIBRIS is given in terms of information that is up-to-day, comprehensive and easy to understand • LIBRIS assists the consumers of services to access other libraries • LIBRIS is conceived and evaluated as a very good catalogue for researchers and other consumers • LIBRIS means no extra work for personnel of local libraries (registration on LIBRIS is 1 time through local) • LIBRIS balance solution supports both for global interest and local interest	According to our opinion, LIBRIS is a successful service broker because all stakeholders seem to be satisfied. Thus, the value of LIBRIS is derived from the mutual satisfaction of all stakeholders rather than the typical value of information economics, i.e. reduced co-ordination costs.

Theme: The Crucial constraints of Collaboration

Issues (A04): What are the most crucial constraints upon which the whole concept of collaboration is based?

Theoretical views	Empirical views	Similarities and differences
• **Ownership of services** A composite service such as a business process involves several owners. It follows that the success of such services depends on a group of different owners and their motivational and ethical investments. • **IT-based and human- based efforts**	• **Ownership of services** The collaborative environment of academic libraries consists of many owners of "products" (owned literature) that become available for the consumers of the involved service providers. However, the motivational and ethical investment of partners may be seen as the ground for the collaboration.	There is a strong agreement between the theoretical views and empirical views with respect to the issues of (1) ownership, (2) the kind of efforts, (3) the mode of coordination and (4) the language of business. The environment of the academic libraries must promote their collaboration.

Most of the SOA literature refers to IT-based efforts. However, there are cases that the literature refers even to human-based efforts. Accordingly, a particular business process can indicate and use both kinds of efforts. • **Coordination philosophy** If the SOA environment has only one owner, then the issues of coordination is not a problem. Either the mode of "orchestration" or the corresponding "choreography" can easily be arranged. However in a network that involves several owners the issues of coordination are not explicitly discussed. • **The language of business** The SOA environment should be described in the language of business.	• **IT-based and human- based efforts** The collaborative environment of academic libraries presupposes the employment of both IT- respective human efforts. However, the labor costs for the collaboration self is trivial because all the work is done locally. Thus, the collaboration does not result in extra work load. • **Coordination philosophy** Any case of inter-library loan is coordinated by the library that becomes responsible to manage a particular consumer order. In this sense we have a case of "orchestration" • **The language of business** This environment has always described in the language of business.	However, according to the existing policies, any action that limit of the local freedom of a library should be avoided.

Theme: The expected and real value of Service Providers		
Issues (A05): What is the perceived and real value of collaborated providers?		
Theoretical views	**Empirical views**	**Similarities and differences**
In a collaborative environment, the value of a provider can be given in terms of the participation in different service transactions. However, any form of participation presupposes some form of reward in order to maintain the motivation of the providers to continue the collaboration.	The Local libraries can provide more services through collaboration. For instance, Linholmen library exports literatures more than import, and therefore, they get money from the Royal library. However, the primary source of motivation is given in terms of higher local performance. These performances are impossible to obtain without collaborative efforts of the other providers.	There is a strong agreement between the theoretical and empirical views of the value of providers. In any case, seldom the expected and the real value of providers can be in harmony or balance. The more providers are involved in the satisfaction of consumers wants under a particular period of time, the more is their real value and therefore their rewards.

6.3   A praxeological[42] view of service-based business environment

The preconditions for understanding the role of governance can be provided by a few themes grouped into four areas of interests, namely: (1) Who decides, (2) What is the subject matter of these decisions i.e. purposes, principles, policies, etc. that clarify how to do the right things, i.e. providing wise direction for development and success, (3) How these decisions are understood and executed. The subject of these decisions is given in terms of developmental activities as well as best practices, procedures and the like. The aims of these decisions clarify how to do things in the right way, and lastly, (4) evaluating the results of development in terms of performance. Thus, the themes of praxeology, i.e. themes dealing with the practical rather than the rational, wisdom of governance are the following.

- Stakeholders and their involvement in decisions
    - The owners of the collaborative service environment
    - The distribution of decision rights
    - Critical strategic decisions
    - Managing the issues of compliance
- The establishment of policies
    - Policies dealing with the kind of provided service
    - Policies dealing with the expected quality of service
- Decisions dealing with the issues of development
    - Decisions dealing with the definition and coordination of the global development
    - Local inter-dependences between Service providers
    - The development of Service Broker
- Evaluation of performance
    - Evaluating the performance of service providers

6.3.1.1 Stakeholders and their involvement in decisions

Theme: The owners of the collaborative service environment		
Issues (P01): Who are the real owners of the collaborative environment? (Who has the right to close down the collaborative service enterprise?)		
Theoretical views	**Empirical views**	**Similarities and differences**
Any form of collaboration between several service providers presupposes the establishment of some negotiated agreement. This agreement may be a contractual arrangement or some form of coalition, i.e. strategic alliance between two or several partners. In both cases the "owner" of a collaborative environment is	The owner of the collaborative environment in this case is the Royal Library under the government where the Royal Library has the right in most issues. The Royal Library gets money support from the government. However, the members of the alliance have the right to end their collaboration and withdraw their efforts.	There is a relatively good agreement between the theoretical and the empirical views of "ownership". In our case, the Royal Library is treated as the owner because is the crucial financial link between the real owner, i.e. the state and the academic libraries that belong to the strategic alliance. However, there is a lack of official

---

[42] Praxeology deals with the theories of decision and action. In this sense, any kind of decision taken by the SOA governance that shapes, evaluates or changes the business or social environment belongs to domain of praxeology.

regulated either by the contract or by the constitution of the concerned alliance.	Official information about the conditions and arrangements upon which the concerned alliance is established is missing. .	information about the conditions upon which the concerned alliance was established. What we know is that the alliance exists and that the members of the alliance have always the right to end the collaboration.

Theme: Distribution of decision rights		
Issues (P02): How the rights of decisions have been distributed between the stakeholders of a collaborative environment?		
Theoretical views	**Empirical views**	**Similarities and differences**
In the case of contractual arrangements the decision rights may concern only the involvement of service providers in the various forms of business transactions. However, in the case of partnership and strategic alliances the involvements of members cover both operational and strategic issues and therefore decision rights. The usual strategy or pattern for distributing the decision rights within an alliance is that of federation. Accordingly, strategic issues are treated globally by the leaders of coalition whereas any other operational decision belongs to the realm of a particular partner's enterprise. Lastly, federation means global thinking, local action.	• **Strategic decisions** In the case of critical issues such as the integrity of collaboration, decision rights have all leaders of the participating libraries. • **Structural decisions** The decision rights of the academic libraries are not affected by their transactional relationships between the academic libraries. Thus, the autonomy of library-members remains the crucial constraint for the collaboration. • **Operational decisions** These are pure local decisions. However, issues that have to do with the co-operability and interoperability of libraries have been arranged through negotiated routines. • **Technical decision**s Most of the technical issues that deal with the interconnectivity and interoperability of local information systems with LIBRIS have been delegated to the IT department of LIBRIS.	There is relatively a good agreement between the theoretical and the empirical views with respect to the distribution of decision rights. Thus, the distribution of decision rights in this environment is based neither on the principles of "business monarchy" nor the principles of 'IT-department monarchy' Accordingly, like any other forms of alliances the dominated model is that of federalism where strategic issues are treated by the participating members of coalition. In the same sense, other operational issues are treated locally.

Theme: Critical strategic decisions		
Issues (P03): What are the most critical strategic decisions in a collaborating environment?		
Theoretical views	**Empirical views**	**Similarities and differences**
A strategic alliance is a business arrangement in the form of a network of participating and cooperating organizations for their mutual benefit. The most crucial decisions in such alliance deal with: • The issues of independence • The issues of symmetric or proportional benefits • The evaluation over the assigned performance • The contribution of the member organizations to the strategic areas of the development of alliance, (innovation, competence, technology, etc) • The joint development of a sound architecture that uses Information technology and makes the service environment attractive for all stakeholders environment. Members of the organizations may coordinate their efforts, resources, management philosophy, etc. for a variety of purposes.	The first critical decision of stakeholders was to establish the constitution of a strategic alliance. Another critical decision was to change from card system to LIBRIS online in 1975. However, at that time this was applied only to academic libraries but not public libraries. Lastly, a crucial strategic decision was to change catalogue standard from Swedish MARC to MARC21 in 2000 which was decided by LIBRIS. (MARC21 is a library system for describing, cataloging and classifying library materials in accordance with standard rules, record-formats, and terminology)	There is a strong agreement between the theoretical and empirical views dealing with the critical decisions of a coalition. The first third theoretical aspects are constitutional because they establish the frame or platform upon which the goodness of collaboration should be judged. The forth aspect in our case, deals with the development of an IT enabled collaborative environment. The last aspect concerns the development of the collaborative environment with or without the use of information technology. It is true that the current form of environment is much more attractive. However, the architectural concerns dealing with the core underlying idea of collaboration of independent parties is the same as it was with the rise of the alliance.

Theme: Managing the issues of compliance		
Issues (P04): Who is responsible to judge issues of compliance (harmony with established constraints, laws, routines etc. decided either by the community of the collaborative library environment or by the society in general)?		
Theoretical views	**Empirical views**	**Similarities and differences**
Any form of compliance concerns firstly the behaviour of the strategic alliance as a whole. In the same sense, compliance deals even with the behaviour of the participating members organizations. In both cases is expected that the constitutional agreement to make explicitly the obligations of the whole and its parts. The rules of compliance may concern the relationships between service providers and service customers.	The responsibility to follow Government decision on big (critical) issues is a strategic affair that is managed by the strategic group of the alliance. The responsibility to follow the constitutional aspects of the alliance and its development belongs to the leaders. The same holds with the issues based on the coordinated development of collaboration, and the required competences. However, the leadership can delegate some issues of compliance for deeper discussion to different groups. The responsibility for pure technical detailed issues is delegated to either LIBRIS department or local libraries on detail. They have to follow rules, not in details but within framework.	There is a relatively good agreement between the theoretical and empirical views of compliance. For some strategic issues the responsible authority is the leaders of alliance. For other issues the responsibilities to follow the "rules" of the expected behavior is assigned to participating libraries and librarians. However, the rules dealing with the behavior of consumers are local rather than global, (does not regulated by the constitutional frame of the alliance.

6.3.1.2 The establishment of policies

Theme: Policies dealing with the kind of provided service		
Issues (P05): Who define the policies to be followed (either through enforcement or through empowerment) by the members of a collaborative environment?		
Theoretical views	**Empirical views**	**Similarities and differences**
According to our theory and the support of many sources, an attractively architected environment presupposes effective *federated SOA governance.* Thus, the federated SOA governance has the rights to establish a sound framework to be followed by any interesting party. The effectiveness of such governance is derived from its continuous efforts to hold the conformity between the architected environment and (1) The stated intentions, i.e. purposes, policies, principles, constraints, etc. (2) The processes, practices and procedures, etc. that successively implement the stated intentions. (3) The wised decisions of stakeholders that convert the core ideas of SOA into reality. The establishment of a so-called federated registry approach of SOA supports the coordination of any effort and allows maximum flexibility to the participated member organizations to take their local decisions.	Representatives from local libraries have the right to define and redefine the policies to be followed by any developmental activity of LIBRIS. However, such decisions affect only the relationships between the LIBRIS and the member – libraries Mostly, the department of LIBRIS follows the policies stated by the leaders of the alliance. However some purely technical issues have delegated to the LIBRIS department together with	With respect to the issues of policy establishment, there is a strong agreement between the theoretical and the empirical views. However the attractiveness of such architected environment is based on the flexibility provided by LIBRIS. In this sense, LIBRIS provides ***unity in plurality*** because firstly, the architecture establishes the grounds for both global effectiveness and global efficiencies, secondly, the architecture provides maximum flexibility to any effort aiming to develop the local enterprise in accordance with their local traditions, expectations, and other needs. Thus, the architected environment is characterized by a uniform collaboration between heterogeneous participating enterprises.

Theme: Policies dealing with the expected quality of service		
Issues (P06): Who define policies that deal with the expected quality of service as well as the accessibility and availability of services?		
Theoretical views	**Empirical views**	**Similarities and differences**
In a federated collaborative environment the authorities and responsibilities for the definition of the quality of service is in the hands of both the leading group of the alliance as a whole and the members of participating organizations.	The policies that deal with the accessibility and availability of services in a collaborative environment are defined by group of people from LIBRIS while political decisions are made by the Royal Library. However, responsible for the	In principle there is a good agreement between the theoretical and empirical views of service quality. Quality is given as a sense of excellence and is expressed in terms of fitness between the foundation upon which the

However, the concept of quality in general and quality of service in particular lack a uniform definition.

In any case such quality can by understood in the following terms:

- **Socio-cultural quality**
  Customer satisfaction = effectiveness = conformity between expected performance and real performance, equality = social symmetry, etc.

- **Process related quality (service quality)**
  Responsiveness, efficiency, functionality, continuity, accessibility, simplicity, functional interoperability, co-operability, mass customization, economy, consistency, etc.

- **Infological quality** Cognitive conformity, relevancy, simplicity, availability, cognitive interoperability

- **Structural quality**
  Trustworthiness, legitimacy, clarity of authority and responsibility, sense of balance between order and freedom, etc.

- **Contextual quality**
  Sense of resilience, innovativeness, sense of integrity and security, adaptability, business agility, etc.

quality definition of services are mostly the manager of local libraries. These qualities can be judged in the following terms:

- **Socio-cultural quality**
  Customer satisfaction, quickness of service, quality of the loan items, social symmetry, privacy, etc.

- **Process related quality (service quality)**
  Responsiveness, efficiency, functionality, continuity, simplicity, economy, etc.

- **Infological quality**
  Cognitive conformity, relevancy, simplicity, customization, etc.

- **Structural quality**
  Legitimacy, security, authorization, privacy, manageability, etc.

collaboration is based and the experienced socio-cultural, process-related, infological, structural, contextual etc. aspects of quality.

Thus, the leaders of the alliance as well as the leaders of the participating libraries do not only define the degree of service quality but they are responsible for the assurance of such quality.

There is a significant difference between SOA and the collaborative environment of libraries, namely: the ultimate owner of services is the state rather than a network of private interests.

6.3.1.3 Decisions dealing with the issues of development

Theme: Decisions dealing with the definition and coordination of the global development		
Issues (P07): What kinds of decisions are global i.e. decisions that are related with the future development of collaboration?		
Theoretical views	**Empirical views**	**Similarities and differences**
As we have stated in (P03) the global critical decisions dealing with the future of collaboration focus and cover the following issues: • The independence of participating organizations • The symmetric or proportional benefits • The evaluation over the assigned performance • The contribution of the member organizations to the strategic areas of the development of alliance (innovation, competence, technology, etc.) • The joint development of a sound responsive architecture • Etc.	• Decisions that have to do with the use of internet as well as improving and increasing the accessibility of LIBRIS catalogue • Decisions that have to do with the better organization of information in the LIBRIS catalogue • Decisions that constraint to any libraries to change the conceptual scheme that govern the organization of catalogue (MARC21) • Decisions that concern the training and education of librarians in general. More specifically, the people that are responsible for the update of LIBRIS catalogue • Decisions dealing with infrastructural issues of a global nature	There is a good agreement between the theoretical and empirical views with respect to the global decisions. Here the agreement deals just with the contribution of the member organizations to the informational and technological developments of the common federated registry, i.e. LIBRIS.

Theme: Local inter-dependences between Service providers		
Issues (P08): How dependent are the development of service providers to other providers?		
Theoretical views	**Empirical views**	**Similarities and differences**
Each service provider is an autonomous entity. This means that every decision concerning the development of a service provider is taken locally rather than collaboratively. However, in a federative environment, - an environment formed by contractual or relational arrangements of two or more organizations -, the development of a federated registry unit has significant consequences with all parties.	In the context of collaboration of academic libraries, every library has the right to further develop its enterprise without any consideration of the other libraries. Thus, changes in the local enterprise as well as its systems remain a local affair. The only dependences that exist are those of LIBRIS. However, these dependencies do not affect the decisions that have to do with the local development of libraries.	There is a strong agreement between the theoretical and the empirical views of the above issue. The collaborative environment of academic libraries may be seen as the best expression of SOA intentions with respect to the core idea of agility, economy customer satisfaction etc.

Theme: The development of Service Broker		
Issues (P09): How the future development of service broker affects the development of service providers?		
Theoretical views	**Empirical views**	**Similarities and differences**
The federative registry unit, i.e. service broker, is the most significant part of the architected collaborative environment. However, behind every architecture lies a core idea that shape the structures, relationships and behaviour of processes, systems, people, etc. In the context of SOA, there are several core ideas, namely; IT alignment with business, business agility, asset reusability, etc. IT alignment with business is a presupposition for any other core idea. This form of alignment expresses the dominance of business over technology. Therefore, any attempt to put the enterprise into the "enterprise system" leads to misalignment. However, Business – IT alignment does not take into account the costs of ad-hoc integration or the requisites of business agility. Therefore, the requisite of business agility provides the requisites of integration and interoperability though SOA pattern that establish loosely coupling integration, message based interoperability, and business based co-operability. However, the satisfaction of requisites of reusability creates another pattern that increases the technological and developmental agility as well as the significant economies of IT. Reusability means that required services may be available through (1) Legacy systems (2) External service providers i.e. IBM, Microsoft, Oracle, etc. (3) In-house development. In any case this architectural form of SOA has been criticized.	LIBRIS should not limit the freedom of local libraries. Any proposal for more centralistic solutions usually is rejected. For instance, the proposal to change the role of LIBRIS as service broker to the role of service provider was rejected. The core idea behind the architecture of the collaborative environment is based on enterprise-wide agility. This decision was taken much before any discussion of an IT-based federated registry. However, any attempt to decrease the independence of participating organizations through the further development of collaboration through reusability was rejected.	There is a strong agreement between theory and reality. However, the debate between reusability and agility remains crucial decision issue. In this work the participating organizations a clear preference about to the pattern of SOA that promote business agility rather than the pattern of SOA aiming to promote the ideas of reusability. There are always those organizations that prefer a SOA pattern that promote a sound balance between business agility and assets reusability. Lastly, the possibility of SOA paradox must not be forgotten. This means that the requisite of services from business cannot be satisfied by the available assets and capabilities. In the same sense, the available assets and capabilities consumes only resources without providing some meaningful benefit. The reason is derived by analogy from the corresponding information paradox.

6.3.1.4 Evaluation of performance

Theme: Evaluating the performance of service providers		
Issues (P10): How the performance of the collaborative environment can be measured?		
Theoretical views	**Empirical views**	**Similarities and differences**
The performance of a collaborative environment can be measured in several ways. Strategically, the value of the collaboration can be given in terms of alliance continuity, members' independence, symmetric/proportional[43] benefits, business and IT-alignment, etc. Operationally, the value of collaboration can be given in terms business agility, asset reusability, customers satisfaction, etc. In any case, the value of such environment depends on the efforts of federal registry[44] to be consistent, simple, accessible, available, etc. In the same sense, the vale of a collaborative environment can be given in terms of empowerment rather than enforcement of the know-how competencies of people.	There are several ways to evaluate the performance of local libraries, for instance, • Statistical • Directly ask the users about the services • Electronic feedback form • Request box • Email address for feedback and complaint However, the current function on the website, "Ask the Librarian", will be terminated since several questions are not asked in the right objectives. The evaluation of LIBRIS out of a consumer perspective is given in terms of amount of daily accesses.	There is a good agreement between theory and reality with respect to the operational view of performance. Today there is a lack of a systematic evaluation of the collaborative environment as a whole. We know that this kind of information should be available in the near future because as we have stated before, the collaborative environment of academic libraries of Sweden is the best empirical and understandable view of SOA and SOA Governance.

6.4 A summary of similarities and differences of SOA and SOA Governance

The three dimensional of SOA Governance defines the crucial roles of governance in general, [MAGO1998], and of SOA Governance in particular. In summary, the roles of governance may be given in the following terms:

- How governance configure or shape their business environment, i.e. issue of morphology
- How governance evaluate their business environment, i.e. issue of axiology

---

[43] Symmetric means equal benefit for everyone while proportional is a form of symmetric. However, this asymmetric, proportional, can be explained by different amount of effort that has been provided.

[44] The concept of federation expresses the common interest of stakeholders to have the function of broker together. Federation represents the situation where several organizations either in form of alliance, contractual or partnership decide some form of coordination.

- How governance changes their business environment i.e. issue of praxeology

However, the wisdom[45] of governance is given in terms of a sound architecture that aligns human capabilities and IT assets with the ever changing expectations of stakeholders.

---

[45] The wisdom of governance is expressed in terms of golden mean rather than opportunistic or pessimistic extremes.

7. Discussion

This section deals with the strategic and operational roles of SOA Governance. Furthermore, we discuss other experience of this inquiry dealing with factors that inhibit the understanding of SOA as well as an architectural design for balancing the requisite of efficiency and business agility. Lastly, we make a meaningful proposal for future research,

7.1   The strategic and operational roles of SOA Governance

The process of this inquiry has focused to elucidate the following issues: *How SOA Governance promotes the attractiveness of a service-based business environment as well as what factors promote or inhibit the role of SOA governance?* In order to improve our understanding we have decompose the issue into two constituent parts, namely, (1) the efforts of governance to configure, i.e. shape, an attractive model of SOA, aiming to cover the interest of all involved stakeholders, and (2) the coordinating efforts of service providers aiming to satisfy the ever changing demands of consumers. In this context the attractiveness of such environment is promoted by the established and management of a service broker. For instance, the rise and development of LIBRIS system within the collaborative environment of academic libraries is a good example of an attractive environment.

In any case, without the wise efforts of a SOA Governance it is impossible to have an attractive service environment. The model below indicates in very simple terms why SOA Governance matter[46].
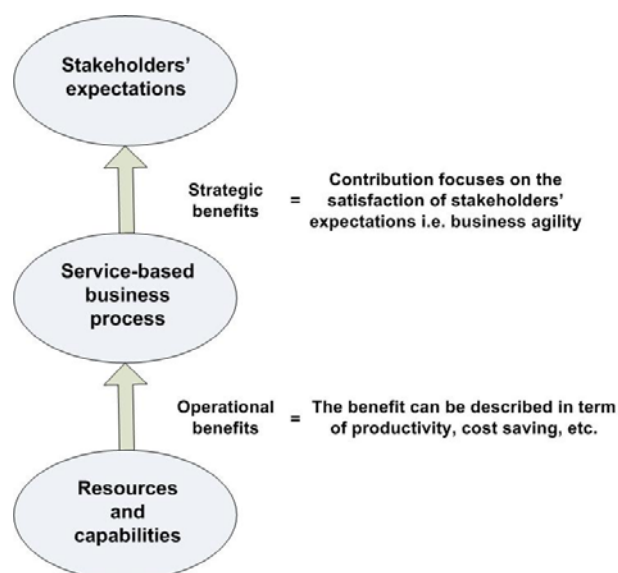
Figure 20: A simple model for understanding the scope and nature of governance

---

[46] The construction of this model is based on the ideas that have been presented in [FEA 2007].

The strategic role of SOA Governance is given of an enterprise architecture that cover the relationships between the expectations of the stakeholders and the service-based business processes aiming to satisfy- in high degree- both stated and unstated expectations.

In the same sense, the operational role of SOA Governance concerns an end-to-end business architecture that defines both the involvement of human and IT capabilities and assets as well as the coordination philosophy applied in that environment.

According to this thesis, the concept of governance concerns both a strategic role and unoperational role.

The strategic role concerns all the decisions that together define wise framework guiding the operational governance. Accordingly, this strategic framework consists of purposes, policies, principles, constraints, etc. The primary aim of this strategic framework is to define a long term orientation for any developmental activities. The impact of this activity may be defined (1) in terms of better training and education of employees or (2) in terms of better information system and ICT support or (3) effective and attractive business processes or (4) any other aspect that make the architected environment attractive. Thus, the ultimate concern of this strategic framework is given in terms of attractiveness aiming to satisfy consumers' expectation, employees, providers, etc.

However, the establishment of such strategic framework is the result of negotiation between parties having different interests as well as the result of empowerment rather than enforcement. Therefore, in this study, we indicate how critical is to establish and maintain both strategic and motivational alignment [See Figure 25].

In the same sense, the operational part of governance consists of processes, practices, procedures, standards, and everything that can make the developmental activities comprehensive, mutually understand, and meaningful. However, this part of operational decision can delegate to the local units of business or make some arrangement with external providers.

In any case, both parts are harmonized through the continuous evaluation of the effect of development. By this way, the stakeholders learn either to adjust the strategic framework or to adjust the logic, capabilities, assets, etc. of development activities. The main result of this adjustments can be expressed in terms of operational and infological alignments [See Figure 25].

In summary, the two sides of governance are given in terms of doing the right thing and doing this in the right way. In the first case, we believe the wisdom of stakeholders whereas in the second case, we believe in the excellent know-how of the stakeholders.

A last word, in the confusing world of SOA and SOA Governance, we have tried to provide representative view of SOA concept and its governance rather than a complete one. We have decided to follow the discipline stating that there is no difference between Enterprise Architecture and SOA. The existing differences are a result of

misunderstanding rather than a substantial aspect. Therefore, many people in this domain predict such differences should be disappeared within 3-5 years period. We have chosen the case of LIBRIS and the academic collaborative environment of Swedish libraries to demonstrate that SOA has been existed much before it becomes the subject of marketing. Accordingly, this environment demonstrates the most attractive application of ICT in order to improve the interoperability and co-operability of academic libraries, by this way, satisfied all the involved stakeholders.

7.2   Other experiences from the work of this inquiry

There are at least three factors that inhibit the crucial role of SOA Governance. Accordingly, each one of these factors needs its specific treatment. Together these "strategies" should lead to a better understanding of how to support the crucial role of SOA Governance. The factors that inhibit the governance can be stated in the following terms.

Firstly, the terminology used in the domain of SOA is confused, inconsistent, vague, fuzzy, incomplete, weak, etc. All these attributes indicate one and the same thing, namely, uncertainty and equivocation in communication.

Secondly, another factor that prevents the role of SOA governance can be explained in terms of the core ideas underlying the architecture. Accordingly, there is not just one core idea underlying the concept of SOA but many, (alignment, business agility, reusability, efficiency). The existence of these different contradictory forces makes the role of governance very difficult.

Lastly, the third factor that inhibits the role of governance can be derived from the fact that the models of SOA are totally isolated from the model of SOA Governance. Therefore, the expected support from the various frameworks leads to misunderstandings and equivocation rather than understanding and clarity.

7.2.1 Managing the language of SOA and SOA Governance through the use of FEM model

Governance means decision and decision always presupposes communication between all people that can affect or being affected by these decisions, i.e. the stakeholders However, effective communication, in order to be effective must be concentrated to a particular domain of interest such as the domain of business. By this way, the language of SOA can be proved by separating the domain of business from the domain of technology. This does not mean that the information technology takes a secondary role in the development of SOA. Instead, this means that the architected service environment firstly shall be described in common and understandable language. Therefore, we must avoid language that allows the communication between experts and inhibit the real stakeholders to participate actively in the decisions that aim to establish the SOA architecture.

The requisite to define or describe the SOA concepts in business terms is a critical factor for improving the comprehensibility, mutual understanding, and possibility to success.

In this work we have applied the FEM model, (Framework for Enterprise Morphology) in order to satisfy the following analytical tasks:

- The use of FEM model in the analytical work of architecture follows the new trend of other researchers that equate the Service Oriented Architecture (SOA) with the Enterprise Architecture (EA), [BLOO2006][47]. If this trend holds then SOA must be described, defined, evaluated, and managed as any other style of EA. The contribution of the FEM model can be given in the treatment of the following issues:
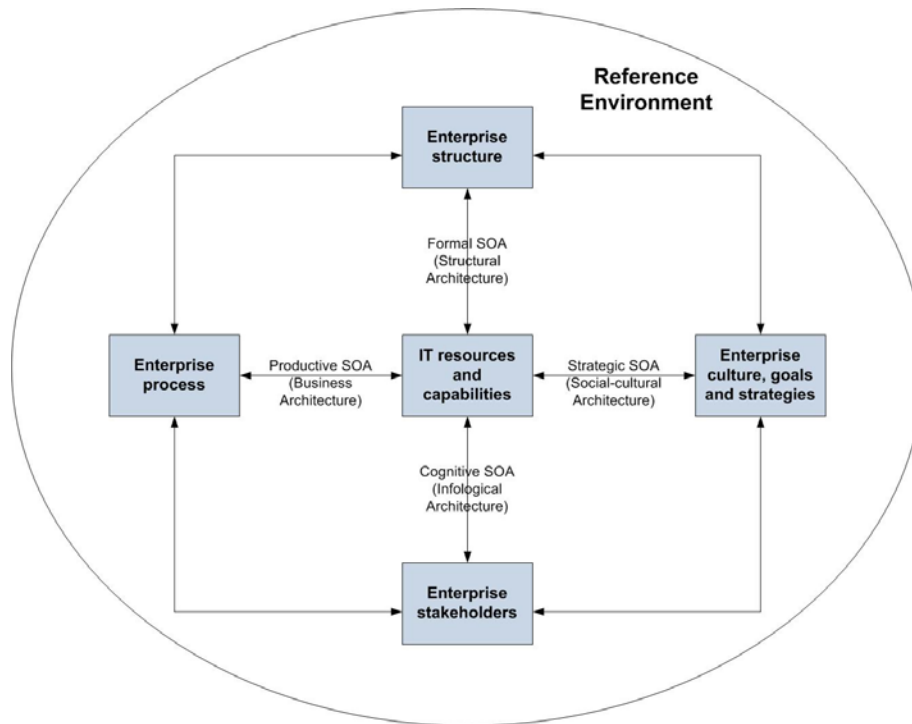
Figure 21: The application of the Framework for Enterprise Morphology (FEM) Model in the analysis of SOA

  o Firstly, we have used the model in order to isolate the business models from the influences of languages from the technical and infrastructural domains. By this way we isolate what the consumer wants from what the providers supply in order to satisfy the stated or even unstated wants. Furthermore, in all design methodologies the issues of architectural design have always treated independently from the issues of the implementation in general and the issues of technical implementation in particular.

  o Secondly, we have used the model in order to distil and describe a list of relevant expected business service qualities that are directly associated with the defined sub-architectures of the FEM Model.

---

[47] See Appendix B where several consultants say that SOA = EA

o   Thirdly, we have used the model in order to make visible the strategic and operational parts of SOA governance. The first part dealing with establishing goals, purposes, policies, principles, constraints, etc. This is the part that direct and coordinate every effort towards doing the right thing. The other part dealing with the issues of efficiency, cost effectiveness, timelines, etc. has to do with the doing things in the right way. Together these two parts define the substantial matter of governance where is expected to be treated by the stakeholders or their (representatives) of business.

o   Fourthly, the model has been used in order to clarify systematically, the similarities and differences between dominated architectural styles. We will demonstrate this issue later on.

7.2.2   *Managing the SOA paradox*

A paradox reflects a situation of self-contradiction or a situation where apparently several true views together lead to contradiction. The most famous paradox is so-called management paradox, [THOM1967]. The paradox deals with the apparent independence of organizations from the local authorities. Accordingly, the larger the organization, the more dependent becomes the local authority by the success or failures of these concerned organizations.

Another paradox, - called the information paradox - states that the information required by the users can seldom be provided by the information systems and the information maintained by the information systems nobody wants, [LANG1984], [ACKO1967] and [HEWI1986]

Even the so-called information processing paradigm may be treated as a paradox. According to Galbraith, [GALB1977] uncertainty is a state where the information processing capacity of the organization is larger than the corresponding available information processing capacity. Therefore, the more information processing capacity is provided, the less the perceived uncertainty. The same holds for the relationship between required and available information. However, according to Langefors the more information becomes available, the more uncertain becomes the decision makers, [LANG1984].

In the same sense, we can study what we called service paradox. Accordingly, the required services that a particular business process defines, are expected to provided by the available reusable services. These services can be owned either by the organization self or can be borrowed from other organizations. However, the situation of the requisite balance between required and available services is not so easy. We shall demonstrate these difficulties through a very relevant model, [DAFT1978].
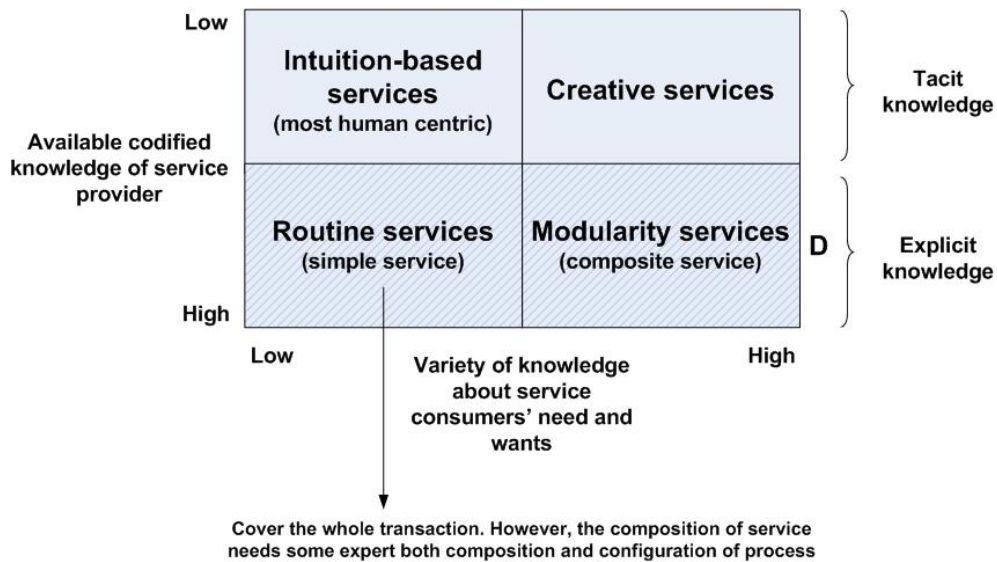
Figure 22: Four states of balance between required and available services

As the above model demonstrates there are four particular situations where the issues of requisite balance must be solved.

The first situation, characterized by known requirements with low variety where can be satisfied by the available owned means of the organization. It is just a repeating situation addressed usually by standard operating procedures.

Secondly, when the variety of requirements is high then some form of modularization is expected, as well as, the implementation of modular requisites by the available knowledge, routines, procedures, etc. available.

Lastly, the other two situations, lack explicit codified knowledge and therefore states the requisites of satisfying requirements though the employment of human efforts and capabilities.

Now, the repeating realm of many formalized services is expected to be easily implemented with reusable services. However, the situation is not the same in the case we have requirements of high variety. Here is the case of reorganizing rather than implementing. According to the law of requisite variety, [ASHB1956], *only variety can absorb variety*. Thus, Simon, [SIMO1969], presupposes a nearly decomposable organizational form rather than reusable services. The same reasoning has been followed by the theory of Galbraith, [GALB1977].

However, given that we have a complete architecture of a particular business process that is defined in terms of required services then the issues of the requisite of balance can be understood better through the following three conditions.

Condition 1   The Wanted capabilities (W) equal the Available capabilities.
Accordingly,   **W=A**. This is an utopian situation where the requisite services can be satisfied by formalized and repeating reusable services.

Condition 2   The Wanted capabilities are greater than the Available capabilities. Accordingly, **W> A**. This situation expects that the requisite balance can be obtained through one of the following "strategies":

1. Outsourcing
2. Alliances
3. Human-Centric Efforts i.e. education, expert support
4. Reorganization , more autonomy, and decoupling
5. Customization of required services to be equal with the available services

If the strategies of outsourcing, alliance, and human efforts are not preferred, then the balance can be obtained by strategies of reorganization or the strategy of customization.

Condition 3   The Wanted capabilities are less than the Available capabilities, Accordingly, **W < A** holds. This situation demonstrates that no one wants our available services. However, the available services can be very useful in other situations that are not repeating in the same sense, that the concept is used in the current literature. For instance, the situations of emergency and large catastrophes may be seen as such case where the condition can be explained in economic terms.

In summary, we can state the following considerations:

- SOA can create balance not because the redundancy but because the possibility of outsourcing (borrow IT-service based from external provider that own the service)
- SOA can create balance through the establishment of an alliance such as library alliance in Sweden
- SOA can create balance, through education, training, development and use of human capabilities

Furthermore, in order to explain the customization strategy, it is necessary to provide some understanding of how different concepts are used within the context of SOA and SOA Governance.

The foundation (or paradigm) of any architecture may be defined as a framework consisting of the following parts (1) the core ideas, (2) the basic building blocks (3) the corresponding architectural styles that hold together the blocks, (4) the formal purposes i.e. the goals of architects, (5) the principles and rules to be followed, (6) some logic of know how architectural design, (7) etc.

Accordingly, three terms of architectural design are necessary to be understood. These are, the concepts of (1) configuration (2) composition and (3) customization.

Configuration
Conceptual View

Broker

Provider — Consumer

Enterprise Architecture = SOA

Composition
Operational View

Broker

Provider
A1 A2 A3 — Consumer

Composition of BP

The composition of BP through
business architecture ≠ SOA

Customization
Adaptive View

Broker

Provider — Consumer

Alignment can be obtained to two
strategies;
(a.) Providers assets and capabilities
customized too much consumers
expectation
(b.) Consumers expectation and wants are
adapted to the available assets and
capabilities of providers

Figure 23: Understanding the concept of configuration, composition and customization

Firstly, the configuration of a SOA architected environment is just a conceptual view of architecture covering both the scope of business as well as the scope of responsibilities. Therefore a sound definition of a SOA must cover (1) the domain of Service Consumers, (2) the domain of Service Providers and (3) the domain of Service Broker. Lastly, a sound and complete definition of SOA must follow the definition of enterprise architecture.

Secondly, the concept of composition refers to a particular business process. The process can be defined through the use of business architecture. However, such architecture is a meaningful part of enterprise architecture. Other views that are parts of the same enterprise architecture are (1) the socio-cultural, i.e. the business goals, values, etc. to be satisfied by the business efforts, (2) the structural view, i.e. the rights of decision over any business effort, (3) the infological view, i.e. agreement of interpretations (4) contextual view, the mutual impact of sub-architectures as well as the issues multi-lingual, environmental constraints, etc.
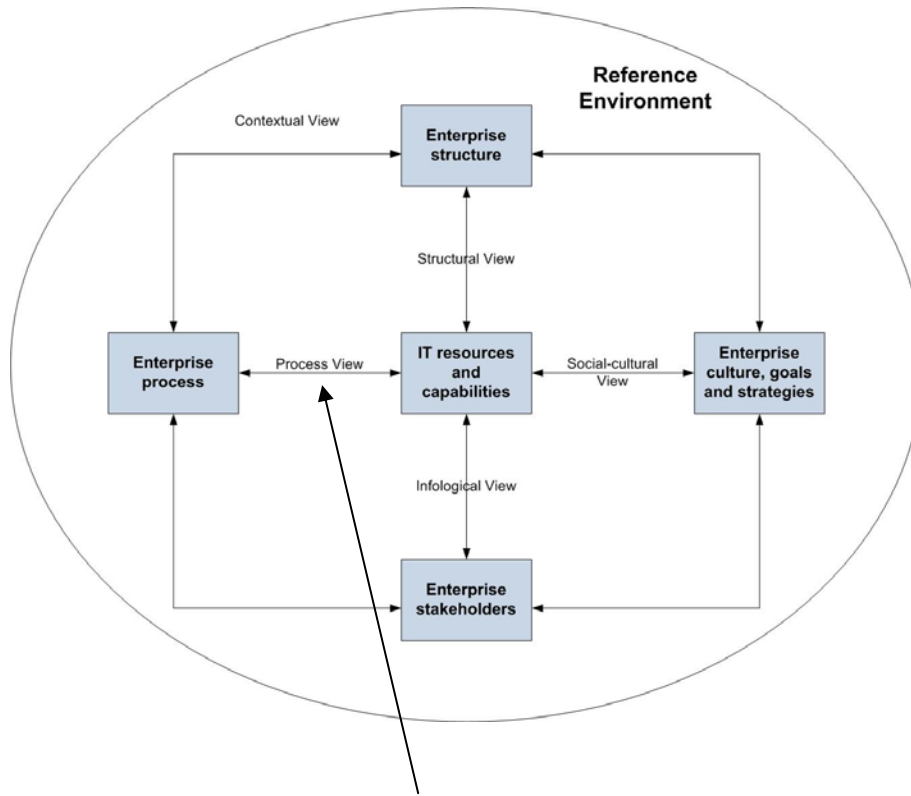
Figure 24: Scope of composition

Lastly, the concept of customization refers to two distinct kinds of changes. In the first case the defined business process is completely satisfied by the available services. However, this definition of the process can be seen as a workable view. New events may require more services and other events may point out the establishment of new relationships between services, etc. In this sense, the business process is customized by the unstable nature of the business environment. In the second case, the requirements defined by a particular business process must be redefined and adapted just to the available services. This form of customization follows the logic of 'putting the enterprise into the enterprise system, [DAVE1998].

7.2.3 *Managing the balance between core architectural ideas*

Underlying every architecture there is always a core idea or force[48] that dictates the composition and/or configuration of enterprises in general and of service oriented enterprises in particular. Within the context of SOA and SOA Governance, there are several such core architectural ideas.

Core idea underlying architecture (Desired or enforced behavior)	Relevant architectural style (Delivered behavior)	Primary value of service oriented architecture
Business efficiency	Architecture that excludes/ inhibits redundant services. Thereby it creates some form of inter-dependence	Ability to manage reusability

---

[48] See Minztberg, [MINT1989]

Business responsiveness	Architecture that promote the redundancy. Thereby satisfies the requisite of independence	Ability to satisfy quickly consumers' wants through the employment of simple of composite services
Business innovation	Architecture that promotes mutual understanding and thereby satisfies the requisites for communication, learning, etc.	Creativity, ability to create new products, or ability to create new logic of producing the existing products
Business adaptation	Architecture that quickly satisfies the requisites of an ever changing business environmental with respect to compliance (rules that enforce changes)	Ability to adapt quickly to a lawful state of affairs

Table 4: Relationship between core ideas architectural styles and values

The above four situations expresses desired behavioral characteristics that are expected to be satisfied by the chosen architectural style of SOA. The first core idea or force refers to the requisite of business efficiency. However, the three other forces are expressions of business agility.

In any case, there is a common, presupposition, namely, the requisite of architectural balance between business wants and expectations, and the available assets and capabilities. This kind of balance concerns the strategic or operational alignment (harmony, compatibility, fitness, matching, etc.) between business ends and business efforts. This kind of alignment can be satisfied through the proper balance between the above core ideas. Such balance called by Mintzberg "configuration", that is just a balance architectural form, [MINT1989]. However, the last aspect can be conceived in terms of investments in human capabilities and IT assets. Accordingly, this is a long term aspect of enterprise development that cannot satisfy immediate consumers' need or expectation.

A last word, without alignment, there is no place either for the requisite of efficiency or for the requisite of agility. Therefore, enterprise architecture in general and service oriented architecture in particular, expresses a state of alignment. That is architecture because without alignment there is not architecture.

In summary, any kind of enterprise architecture or SOA expresses the critical balance or alignment between the expectations and wants from the consumer side, and human capabilities and IT asset from the provider side. Not any architecture can provide this kind of balance, harmony, fitness, matching, etc. Thus, the architecture used in the composition of a process is sub-architecture if it lacks reference to the consumer expectation.

.
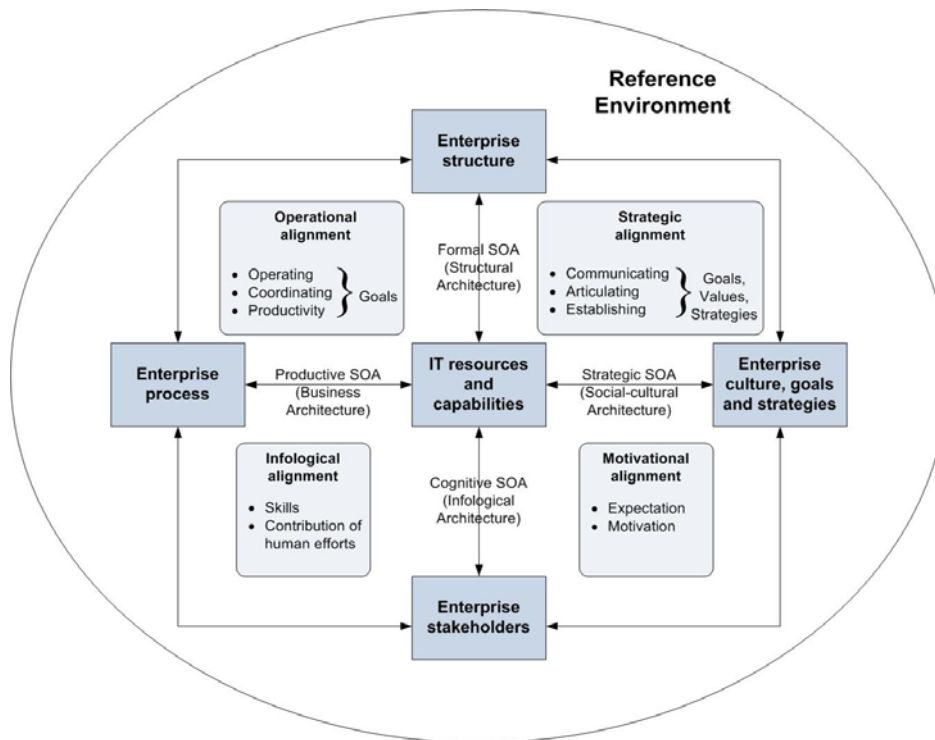
Figure 25: The difference view of alignment between business expectations and business efforts[49]

A comparison between the architectural designs that satisfy the core idea of business efficiency, respectively business agility is given below.

Main architectural characteristics	Business Efficiency	Business Agility
Enterprise architecture of a service based environment	The core idea of such architectural style is based on efficiencies via reusable services.	The core idea of such architectural style is based on business agility via self-contained and loosely related services.
The overall configuration of a service shaped business environment that can be further analyzed in terms of (1) business process architecture, (2) socio-cultural architecture, (3) infological architecture, (4) governance architecture, and (5) contextual architecture.	The architected environment covers the following constituent parts (1) Service consumers, (2) Service providers, (3) Service broker , and (4) different kinds of services	The architected environment covers the following constituent parts (1) Service consumers, (2) Service providers, (3) Service broker , and (4) different kinds of services

---

49
    Usually, alignment is defined in terms of shared goal both by business and by IT department. This kind of alignment is given in financial or economical terms such as ROI (Return of Investment). However, this form of alignment is difficult to measure and it is not a matter of management. The matter of management can be given in terms of Strategic Alignment, Operational Alignment, Infological Alignment, and Motivational Alignment. Thus, for this study, alignment is a matter of attractiveness of architectural design rather than economical effects.

| Socio-cultural architecture

Relationships between mission, vision, goals, objective and the like. the planned or in place, informational, transactional, communicative, etc. services | Overall efficiency and productivity

Meaning ability to reuse already existing (own or outsourced) services in order to (1) eliminate developmental time (2) save costs, (3) secure quality of services, (4) etc. | Business agility

Meaning, ability to respond and satisfy quickly the wants and expectations of consumers |
| --- | --- | --- |
| Infological architecture

Relationships between Stakeholders and the planned or in place informational, transactional, communicative, etc. services | The architecture covers the requisites of the main stakeholder, the IT Department.

However, the crucial issues of inter-operability, provision of rich services and interpretations are excluded, | The architecture covers the requisites of Business Departments

However, the crucial issues of inter-operability , the provision of rich services and the issues of interpretation are excluded |
| Business process architecture

Relationships between the overall structure of a business process and the planned or in place informational, transactional, communicative, etc. services | The architecture covers the requisites of a complete business process (end-to-end) that is defined in terms of reusable services | The architecture covers the requisites of a complete business process (end-to-end) that is agile. |
| Governance architecture

Relationships between the overall structure of authority and responsibilities and the planned or in place informational, transactional, communicative, etc. services | The requisite pattern of governance is a centralized structure of authorities and responsibilities

Accordingly the metaphor of IT monarchy is very representative pattern. | The requisite pattern of governance is a decentralized (or autonomous) pattern of authorities and responsibilities.

Accordingly the metaphors of (1) business monarchy, (2) federation, and (3) duopoly, are representative patterns of governance. |
| Contextual Architecture | Unknown treatment.
The issues of single lingua or multi-lingua,
Ownership of services
Rewards
Environmental Constraints
Motivational strategy, i.e. enforcement or empowerment | Unknown treatment
The issues of single lingua or multi-lingual,
Ownership of services
Rewards
Environmental Constraints
Motivational strategy, i.e. enforcement or empowerment |

Table 5: A comparative view of two dominated architectural designs

However, the architectural reality is created by unintegrative logic that try to establish the balance between agility and efficiency rather than analytical logic that holds the concerned objective isolate of each other. Therefore, according to the law of golden mean, there are always situation that can be stated through balanced view of efficiency and agility.

7.3 Architectural style for balancing business efficiency and business agility (Answer to the issue of conflicting ideas)

In any case, the primary value underlying SOA is given in terms of business agility. However, there can be situation where the issue of efficiency and economy are significant. In such case, we shall find an architectural style that balances the requisite of business agility and business efficiency. However, any such architectural style must demonstrate that it does not followed by significant risk and does not affect the requisite of agility.

The last letter of SOA is "architecture" as such it must provide a preferred architectural style that both promote or protect. However, SOA cannot always do it because the contradictory requisites that must be satisfied by such architecture. Thus, the contingent nature of these characteristics can result the following three options:

1. Dominance of developmental economics satisfied by the architectural characteristic of reusability
2. Dominance of business agility satisfied by the architectural characteristic of modularity and loosely coupling of services
3. Satisfying both objectives by balancing the requisites of reusability and loosely coupling

Therefore, the soundness of the architectural style is based both in the understanding of consequences as well as the requisite for attractive SOA Governance.

In our case such architectural pattern should be based on the message exchange between participating providers, consumers and broker.

For instance, in the case of collaboration between the academic libraries through LIBRIS, the proposal for more integrated solutions where the share ability of services could be accessed through a global, single, and common repository of services stated by the requisite of reusability created undesirable inter-dependability and therefore rejected. The rejection of such architectural was based in the undesirable, interdependency and riskful enterprise, and as well as very bureaucratic enterprise behavior.
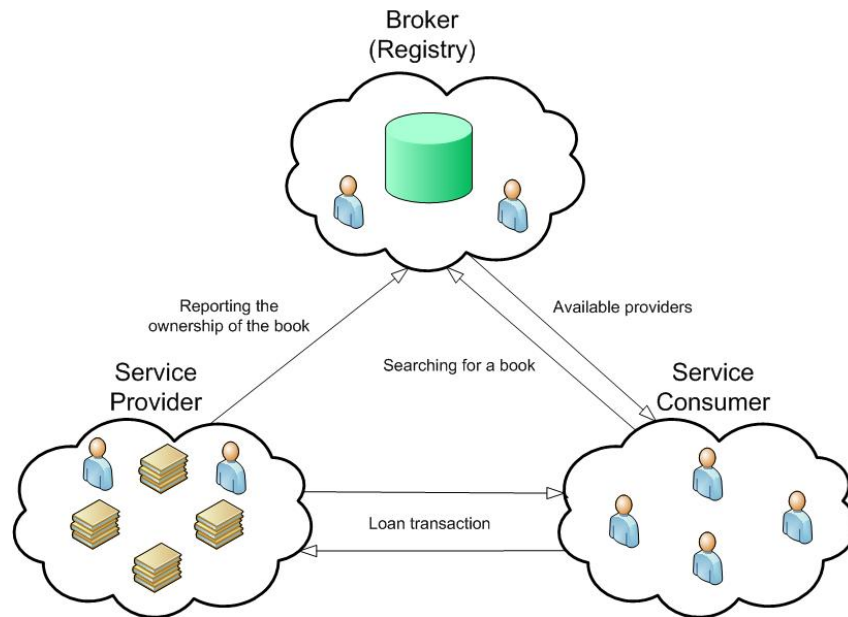
Figure 26: Common SOA configuration

As we can observe in the above most familiar visualization of SOA [See Figure 26], there is no place for a shared global and single repository of services. However, both the service providers and the service consumers should use a common service broker because the constituent parts of the above architectural pattern – providers, consumers, and brokers promotes firstly, the economics of information and secondly, it does not conflict the requisite of business agility. However, the soundness of the above pattern ceases to exist if the broker becomes even a repository unit.

The attractiveness of SOA governance should be understood in terms of partners autonomy: Firstly, the attractive of SOA governance should be given in terms of partners relatively high degree of autonomy. Accordingly, the satisfaction of the requisite of reusability and the associated objective of developmental economy, may limit the degree of partners autonomy and thereby the desirability of business agility cannot be provided. Thus collaboration does not mean necessarily centralization.

The primary purpose of SOA should be the agility of the business enterprise: Service Oriented Architecture (SOA) should organize and shape a collaborative global environment in such a way that primarily promotes the agility of the business enterprise.

Furthermore, SOA holds together the services that satisfy a particular business process. Accordingly, these services usually belong to different owners, i.e. providers. This fact may be seen as another argument for the avoidance of patterns that promote share ability and centralization.

7.4   Proposals for future research

7.4.1   Clarifying the content and configuration of a SOA environment

Our study of SOA has indicated that the basic building blocks i.e. the involved domains of an architected environment is not three but more. As we illustrate in figure below, there is a domain dealing with the issues of distribution as well as a domain dealing with the issues of membership of such environment.
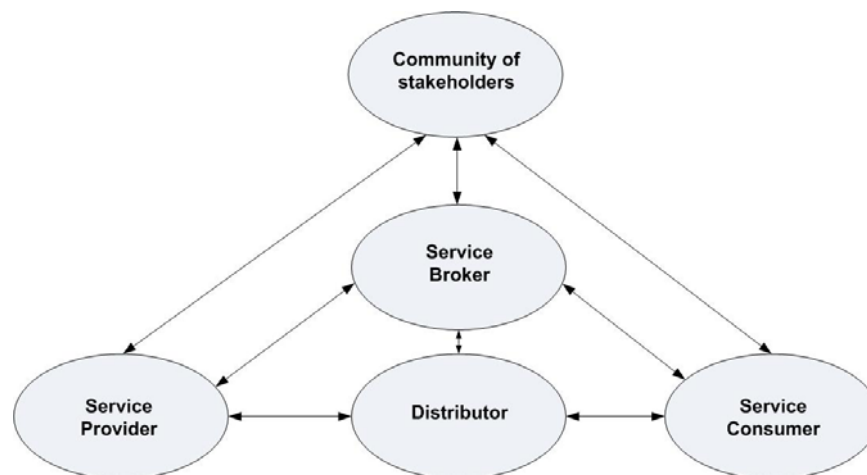
Figure 27: Hidden view of service-based business environment

Therefore it is interesting in the future to improve the existing knowledge that deals with the configuration of a SOA based environment with respect to the belonging domains and the assigned roles of the actors.

Another aspect that remains unclear refers to the role of a service broker. Tendencies to change and redefine the role of broker as a service broker into a service provider, like the case of LIBRIS perhaps are not unique. Therefore, the roles of actors within a service-based business environment must be better clarified.

7.4.2   Clarifying the forms of interoperability in the context of SOA

P. Evans, [EVAN1997], has proposed a model for understanding the issues of interoperability and co-operability ´between organizations and people through the use of information systems. Accordingly, Evans states that information technology improves radically the exchange of information and therefore the interoperability between organizations, people and systems. The model of Evans is very simple. It consists of two dimensions. The first dimension cover the number of receivers, or consumers, or providers, or any entity involved in the concerned co-operations and interoperations. The other dimension refers to the richness of information or richness of services to be provided. Accordingly different media upon which the issues and requirements of interoperability are implemented have different impacts on communication [See Figure 28].
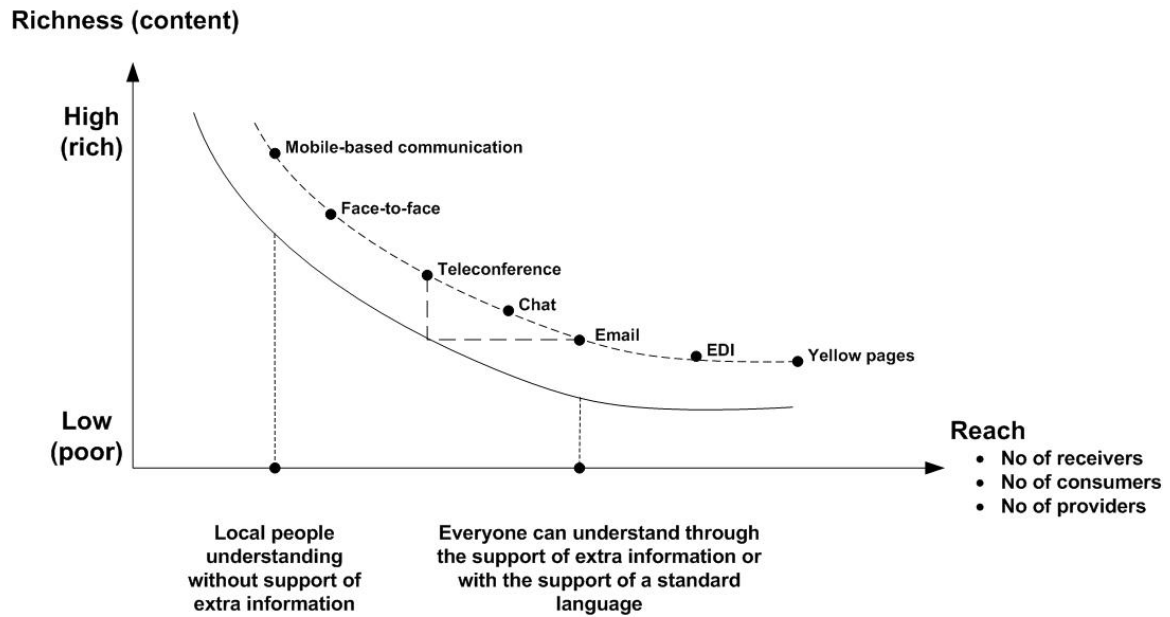
Figure 28: The relation between richness and reach as well as the right media for interoperability

However, the positive view of Evans is not really aligned with the theory of Richard Daft [DAFT1978] and [DAFT1986]. According to the last mentioned theory, and with respect to the nature of interoperability a sound SOA implementation must satisfy the following requirements:

1. Immediate feedback (chat rather than email or EDI[50]-based mail)
2. Communicative cues i.e. something unwritten or unspoken that can alert the awareness of interoperated people
3. Shared language (mostly natural)
4. Personalized (customized) messages

Thus, an attractive SOA environment is implemented by media devices or infrastructures that can satisfy the above four criteria. However, the issue of communicative services has not been addressed in any of the current models, frameworks, methodologies and the like. Therefore, we propose a study that investigates and clarifies the various forms of interoperability within the context of SOA.

7.4.3 *Improving the quality and attractiveness of services and interfaces*

The same SOA architecture can provide different forms of services. For instance, a bank transaction can be managed by three different interfaces. Firstly, the transaction is treated as a self-service like ATM-services. Secondly, the case where the transaction is supported by human agent service. Lastly, the case that the same transaction becomes an internet-based service, etc. However, in all three cases, the architecture remains the same, because the logic of the transaction remains just the same. It is like Tango dance with different parties. The rhythm and the steps of the dance are just independently of who is

---

[50] Electronic Data Interchange

dancing with whom. However, this kind of implementation is convenient for the consumers of services.

The issue of providing relevant and meaningful information of services is an architectural issue. Today most service providers supply unsatisfactory services because the information we want is either not available or irrelevant and the information provided is just the information we do not want. For instance, if the consumer specifies to Google to find information about "SOA Governance", the response is 445,000, however, if we change to "SOA Governance issue", the response changes to 360,000. The modification of "significant issue" is 15,200 results. Lastly, "critical significant issue" is 58,000. The conclusion of this experiment is who shall support or educate the consumer about how to specify a particular service request.

The finding of a service becomes adventure because no one can judge the relevant of responding service. However, the combination of both IT and human capability perhaps provide a more significant response to consumer.

A strategy for improving the quality of both provided services and interfaces can be based on Langefors infological equation, [LANG1984]. Accordingly, information is knowledge exchanged through the use of some familiar language. Furthermore, information can be seen as the result of interpretation and the contribution of such interpretation to the existing knowledge base of receiver.

Thus information is just the result of a cognitive process and can have a significant impact like a change in the existing knowledge base of people (1). In the same sense we can define the impact either as the difference between two cognitive states, the state prior the receiving of the message and the state after the interpretation and understanding of the message. In other words our current state of knowledge is equal with our previous state of knowledge plus the new contribution of information (2).

However, the above two conditions represent the situation where communication is based on people who knows each other and shares the same language. But what happens if we have communication or cooperation between people who neither knows each other nor share the same language? According to the theory, the situation can be treated by extra "data" and by extra time for the same interpretational process (3).

(1)	$I = i (D, S, t)$ OR $S' = i(D, S, t)$
(2)	$I = S' - S$, OR $S' = S + I$
(3)	$I = i (D + D_E; S; t + t_E)$

$I =$ The result of interpretation of D
D and D_E Meaningful symbols that belong to some language
$i =$ The process of interpretation
$S =$ The state of human knowledge base at a particular moment of time
$t =$ Time required in the interpretation of D
$D_E =$ Extra data
$t_E =$ Extra time to be consumed in the interpretation of the message communicated through the D

Table 6: Langefors' infological equation, [LANG1984] and [LANG1986]

Very few designed interfaces follow the axioms of infology. One such case is the Amazon. The attractiveness of this service provider can be evaluated in the terms of infology because the "stranger" consumer looking to buy some book can find extra information dealing with the quality of a particular book.

Therefore it is desirable to investigate the possibilities of infological design and infological criteria addressing the crucial issues of service quality. If we cannot provide a service environment that follows the above infological ideas, can we explain the reasons for our inabilities? Can we understand the undesired consequences of our ignorance?

8.   Conclusions

This last section deals with the subject matter of the thesis, namely, the creation of integrated theory of SOA Governance. Furthermore, the thesis indicates the most critical issues of SOA Governance. Lastly, we make a fruitful comment to Software Engineers with respect to the result of this thesis.

8.1   Towards a sound theory of SOA Governance

The primary objective underlying the study is to improve the understanding of people with respect to how to shape, evaluate and change service-based business environment. Accordingly, the study intends to answer the following issues:

What are the critical roles of SOA Governance that promotes the attractiveness of a service-based business environment as well as what factors can inhibit the role of SOA Governance?

The answers (and therefore the conclusion of this study) to the above issues can be stated as follow:

Firstly, with respect to the first issue dealing with the roles of SOA Governance, namely, *What are the critical roles of SOA Governance that promotes the attractiveness of a service-based business environment?*

Accordingly, the primary and most significant role of SOA Governance is the establishment and management of a negotiated and accepted SOA.

The crucial second role of SOA governance is to use the established architecture and coordinate every related effort that promotes the attractiveness of an architected reality. Therefore, the service oriented architecture follows the wisdom of SOA Governance. With other words, the role of SOA Governance is to manage any effort that changes the service-based business environment in accordance with the patterns, principles, and intentions of established service oriented architecture.

In summary, we can conclude that the wisdom of SOA Governance promotes the attractiveness of a service-based architected environment.
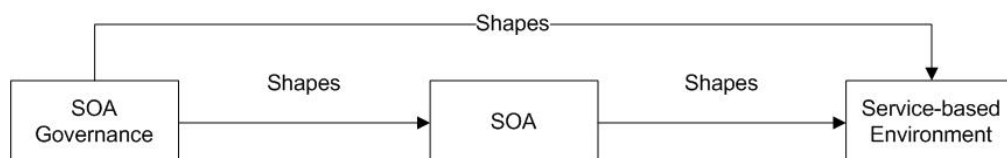
Figure 29: Towards a sound theory of SOA Governance

Our empirical results verify the above relationships that are expected to hold between the SOA Governance and the SOA it uses to coordinate both the operations and development of the service environment. In summary the strategic role of SOA governance is just to establish a service oriented architecture. In the same sense the operational role of SOA Governance concerns the management of any kind of local development or operation in accordance with the patterns and principles of the established architecture. Thus, the attractive architecture that shapes the collaborative environment of the academic libraries of Sweden reflects the harmony between (1) the idea of global thinking of a wise strategic governance, and (2) the idea of local understandable and meaningful actions of the operative governance.

Secondly, with respect to the second issue, namely, *what factors can inhibit the role of SOA Governance?* According to our study, there are three main factors that inhibit the roles of wise governance. Firstly, a fuzzy, inconsistent, incomplete, ambiguous terminology upon which the concept of SOA and SOA Governance are described, designed, evaluated, etc. Secondly, conflicts of interest and contradictory core ideas (alignment, agility, reusability, efficiency, etc.) provided by different disciplines, i.e. Software Engineering, IT Management, Enterprise Architecture, etc., that inhibit the choice of a comprehensive architectural pattern for a service-based business environment. Lastly, the plethora of interesting models of both SOA and SOA Governance are fragmented and therefore incomplete or .unclear and fuzzy. In the face of these critical issues that inhibit the role of wise governance, our study has developed a theory of SOA Governance aiming to clarify the relationship between governance, architecture, and service-based business environment.

In summary, the provided answers that together form the content of this thesis have been derived from an integrated model for SOA Governance. The model has created in order to organize both theoretical and empirical knowledge of SOA Governance.

According to our knowledge, the proposed model satisfies the requisite of validity, i.e. relevance with the most dominated and accepted theoretical views. In the same sense, the model satisfies the requisite of reliability, i.e. harmony between the model and the mental views derived from the interviews. These two criteria together with the treatment of the whole inquiry satisfy the requisites of both rigor (with the principle of science) and relevance (with the nature of humanity) scientific procedures.

8.2   Three critical issues of SOA Governance

***The language of SOA and SOA Governance*:** Governance means decision and decision always presuppose communication between all people that can affect or being affected by these decisions. However, communication, in order to be effective must be concentrated to a particular domain of interest such as the domain of business. By this way, the language of SOA can be proved by separating the domain of business from the domain of the technology. This does not mean that the information technology places a secondary role in the development of SOA. This means that the architected service environment firstly shall be described in common and understandable language, not the language that limits the communication only to the experts.

***The balance of contradictory and conflicting idea*s:** Behind every architectural concept, lies particular core ideas. This principle is applied to the concept of SOA. However,

there are situation where we have several core ideas underlying the concept. For instance, alignment, business agility, reusability, efficiency, etc. are typical core ideas of SOA. However, the existence of these different contradictory forces makes the role of governance very difficult. Therefore, in order to satisfy the requisite of business wants and expectations, the balance of these core ideas should be taken into consideration.

An integrated framework for SOA Governance: In this study, two dominated models are integrated in order to form a new model of governance. These significant models are (1) the model of service-based business environment, and (2) the model of SOA Governance. According to a fuzzy and incomplete view of SOA and SOA Governance, this purposed model aims to give a better understanding and attractiveness view, as well as the relationship between these two aspects, [See Appendix]**.**

The efforts and focus of this work has been defined, delineated and managed with respect to the requisite of collaborative environment representing different practices, purposes, interests, etc. as well as a domain of providers where everyone owns some of the assets to be used by the whole environment. Therefore,

- The attractiveness of SOA governance should be understood in terms of partners autonomy

- It follows that, the primary purpose of SOA should be the agility of the business enterprise

- It follows even that, the soundness of architectural pattern should be based on message exchange

8.3   A last word

A last word for software engineers and software architects. In the formative age of any concept, it is acceptable some degree of equivocation and ambiguity. However, this contradictory nature of SOA must promote the requisite for comprehensibility, communication, shared understanding, and manageability, otherwise, SOA fails. In any case, without comprehensibility, there is neither chance for communication and mutual understanding nor manageability. This is a case of architectural design in general and SOA architecting in particular. It is just the organization of reality in such a way that promotes both communication and understanding. In summary, this means that the success of any developmental project is totally based on our understanding of SOA. Because any action converts our mutual understanding to desirable and controllable affect. Intuition, it is not enough to manage critical and very complex issues of social reality.

Appendix A: Presentation of FEM (Framework for Enterprise Morphology)

Our reality consists of formed objects, entities, environments, processes, etc. These may be planned or in place. In the same sense, the "space" or environment that there objects occupy is given in terms of their relationships. Thus, the enterprise is just an organized space and this organization, the form taken by the enterprise, is called enterprise architecture. Morphology deals with the theories of forms in general and the enterprise architectures in particular.

Accordingly, an enterprise architecture consists of (1) building blocks, and (2) relationships between the building blocks. However, these relationships together create some distinct patterns of organization such as hierarchical, network, contractual, etc. In the same sense, with respect to processes we can distinguish between sequential, overlapping, parallel, etc. patterns of organization.

The building blocks of the model

The framework consists of five building blocks or domains of interest. These are:

- The domain of value-added respectively supportive activities and processes of the enterprise
- The domain covered by formal structure of the enterprise, i.e. the hierarchical, network, contractual, etc arrangement of decision rights and responsibilities
- The domain of goals, objectives, values, mission, visions, world views and the like of the enterprise
- The domain of stakeholders, in a very narrow view, the participants in the activities and decisions of the enterprise. In a broad view, any person that can influence and being influenced by the enterprise
- The domain of any kind of structured or unstructured information systems
- The boundary that separates the above mentioned domains from the external world

All mentioned domains are composite objects. For instance, a value-added process consists of a sequence of meaningful activities that together satisfies a consumer's stated or unstated expectations. The same holds for the other domains.

The relationships among the building blocks

The relationships between domains are the relationships that belong to the enterprise architecture. What is distinct here is that these relationships are "relationships of relationships because they concern the relationships between patterned domains. In any case the concerned relationships are given below:

- Functional Architecture , i.e. the relationships between business processes and information systems
- Structural Architecture, i.e. the relationships between formal responsibilities and decision rights and information systems
- Infological Architecture, i.e. the relationships between human's cognitive capabilities and information systems

- Socio-cultural Architecture, i.e. the relationships between enterprise's goals, objectives, values,, etc. and the information systems
- Contextual Architecture, i.e. all indirect relationships that can affect and being affected by the directly established architectures

The scope covered by the framework

The scope covered by the FEM model is the enterprise. We use the term enterprise as a general term. In this sense the model covers the functional, structural, infological, socio-cultural, and contextual architectures of the enterprise of (1) one organization, or (2) two or more organizations, as well as (3) the transactional relationships between organizations. The figure below represents the concepts and ideas underlying the FEM model.
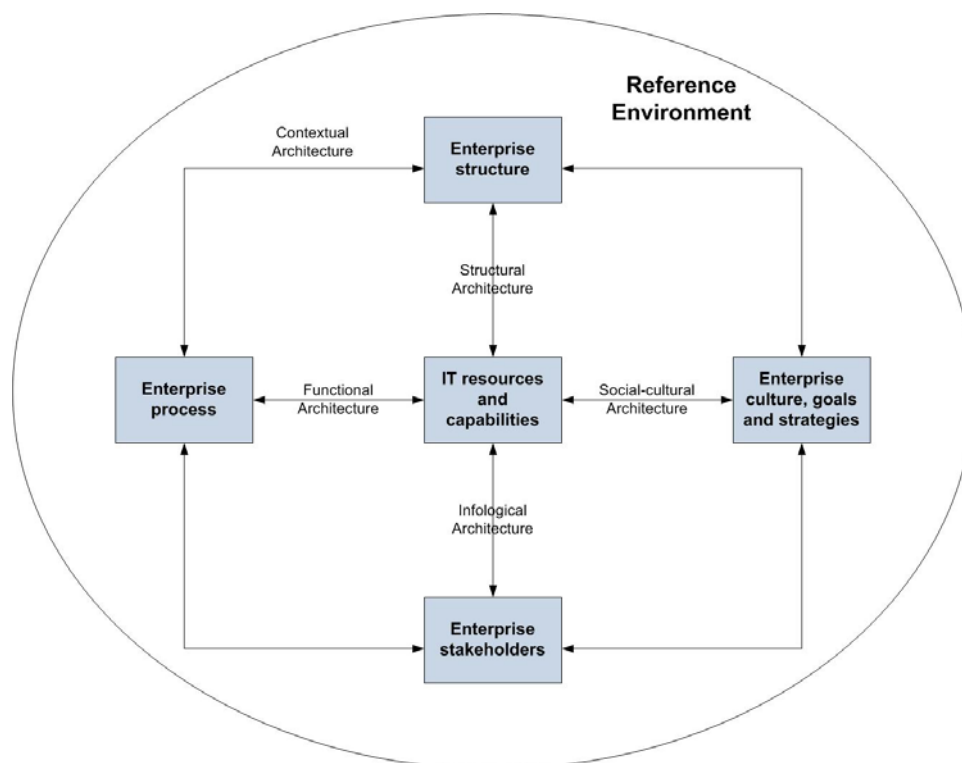
Figure 30: FEM Model

Questions with Respect to SOA-Sub architectures

Functional Architecture

How the content and form of the functional architecture promote or hamper the agility of the business environment?

- *To what extend various kind of business services affect the agility of the business environment*

- *To what extend various kind of communicative services promote respectively hamper the agility of the business environment*

- *To what extend various forms of cooperation are depended on the agility of the business environment*

To what extend various kind of business services affect the agility of the business environment

- Information service
- Decision service
- Transactional service
- Developmental service
- Administrative service
- Operational service
- Relational service
- Communicative service
- Political service, i.e. involving negotiation
- Others

To what extend various kind of communicative services promote respectively hamper the agility of the business environment

- Mobile service
- Telephone service
- Teleconference service
- Yellow pages
- Internet-based front-end services
- Email
- Face-to-face
- Others

Communication is very essential for the business, devices used to connect stakeholders are needed.

To what extend various forms of cooperation are depended on the agility of the business environment

- Informal
- Contractual
- Cooptative
- Coalition (Alliance)
- Others

There are several types of collaboration between firms, therefore it is necessary to know what form dominates or preferred.

References: Galbraith, J. Thompson, T. Magoulas, K. Possi

Structural Architecture

How the content and form of the structural architecture promote or hamper the agility of the business environment?

- *To what extend the following kinds of ownership promote respectively hamper the agility of the business environment*

- *To what extend the following kinds of responsibilities are relevant and fruitful with the issues of agility of the business environment*

- *To what extend the following stakeholders are involved in the decisions about the policies, patterns, processes, etc. of SOA*

To what extend the following kinds of ownership promote respectively hamper the agility of the business environment

- All providers
- One provider
- Few providers
- A third parties
- All the stakeholders
- Others

According to OASIS, SOA concerns a collaborative business environment that consists usually of several owners. Accordingly the more owners the more difficult become to decide quickly. However a powerless owner can decide quickly in just operational issues. Lastly, strategic issues presuppose the participation of all owners.

In a collaborative business environment, it is expected that every participating owner should secure his own autonomy (Learning from LIBRIS).

To what extend the following kinds of responsibilities are relevant and fruitful with the issues of agility of the business environment

- Developmental, i.e. covering the issues of continues development of SOA
- Operational, i.e. covering the issues of SOA within one operational unit (A case of Orchestration)
- Organizational, i.e. covering the issues of SOA between operational units that belong to different departments (A case of choreography)
- Inter-organizational, i.e. covering the issues of SOA between different organizations
- Strategic, i.e. covering the issues of SOA's long-terms fulfillment of stakeholders expectations
- Others

Some approaches treat the issues of responsibility as developmental, other as operational only. Still other approaches treat the same issues as organizational or strategic.

To what extend the following stakeholders are involved in the decisions about the policies, patterns, processes, etc. of SOA

- Business CEO
- Business unit's chefs
- CFO
- CIO
- Partners
- Others

According to literature, SOA affects the behavior of the business environment, i.e. its agility. Accordingly, who is the architect that shapes the behavior of a business environment?

References: T. Davenport, Weiss

Infological Architecture

How the content and form of the infological architecture promote or hamper the agility of the business environment?

- *To what extend the existence and treatment of the following cognitive issues presupposes an agile business environment*

- *To what extend SOA aiming to promote an agile business environment is depended of the following cognitive characteristics*

- *To what extend the following characteristics of SOA can affect negatively the agility of the business environment*

To what extend the existence and treatment of the following cognitive issues presupposes an agile business environment

- Inconsistencies
- Asynchrony
- Contradictions
- Perceived uncertainty
- Equivocation
- Terminological connectivity
- Causal connectivity
- Temporal connectivity
- Communicative connectivity
- Evidential connectivity
- Others

In turbulent business environment, cognitive issues cannot be solved by global models of by Standard Operating Procedures (SOP). Only effective communication can absorb some amount of the above issues.

References: C. Hewitt. R. Daft, N. McIntosh

To what extend SOA aiming to promote an agile business environment is depended of the following cognitive characteristics

- Bounded rationality
- Mutual understanding
- Continuous learning
- Creativity
- Others

The quality of the SOA pattern can be affected by the way of gaining knowledge.

References: H. Simon, B. Langefors, C. Argyris, R. Boland

To what extend the following characteristics of SOA can affect negatively the agility of the business environment

- Formalization
- Centralization
- Globalization
- Synchronization
- Standardization (Unification)
- Differentiation
- Integration
- Others

According to literature, theses SOA-dimensions have a direct affect on human cognitive world.

References: C. Hewitt, B. Langefors, R. Daft, Galbraith, C. Argyris

Socio-cultural Architecture

How the content and form of the socio-cultural architecture promote or hamper the agility of the business environment?

- *To what extend the following objectives compete with the primary requisite of business agility*

- *To what extend the following categories of stakeholders are the beneficial of an agile business environment*

- *To what extend a few basic attributes can be used to judge or measure the value of an agile business environment*

To what extend the following objectives compete with the primary requisite of business agility

- Business process optimization

- Reusability
- Business agility
- Operational economy
- Outsourcing
- Others

These issues address the motivation of applying SOA to the organizations.

To what extend the following categories of stakeholders are the beneficial of an agile business environment

- The owners of SOA
- The actors (employees of service provider)
- The clients
- Others (Community, Authorities, Competitors, etc.)

The concept of SOA must cover both its primary and the secondary stakeholders, i.e. those covered by "Others".

References: P.Checkland, R. Ackoff, W. Churchman, B. Hedberg. B. Langefors

To what extend a few basic attributes can be used to judge or measure the value of an agile business environment

- Efficiency (Less cost and less time)
- Effectiveness (Higher quality)
- Responsiveness (Timelines and "smoothness")
- Partial attractiveness (Customers satisfaction)
- Total attractiveness (Stakeholders satisfaction)
- Security
- Degree of innovativeness
- Cultural feasibility
- Social feasibility
- Privacy (Integrity)
- Openness/Closeness
- Others

In order to have the best SOA environment, these terms should be measure, depending on the goal of the organization.

Contextual Architecture

How the content and form of the contextual architecture clarify the character of the agility of the business environment?

- *To what extend the agile business environment is characterized by various kinds of languages*

- *To what extend the agile business environment is characterized by various categories of value*

- *To what extend the agile business environment is characterized by various categories of legislations and laws*

To what extend the agile business environment is characterized by various kinds of languages

- Only the national language (Homogeneous environment)
- National and English languages (Partially homogeneous)
- Several languages (Heterogeneous)
- Expert languages
- Others

The higher heterogeneity, the more attractive the business environment is.

References: EU's Cultural integrity, Amazon.com

To what extend the agile business environment is characterized by various categories of value

- Economic values (J. Thompson)
- Human values (Abraham Maslow)
- Cultural values (P. Checkland, W. Churchman)
- Social values (Keen, B. Hedberg, B. Langefors)
- Managerial values (Comprehensibility, shared understanding, meaningfulness = win-win)

The more heterogeneity, the more difficult become to find unity in diversity.

To what extend the agile business environment is characterized by various categories of legislations and laws

- Local legislations and local laws
- Regional legislations and laws
- National legislations and National laws
- Global legislations and global laws
- Others

The more legal authorities, the more complexity in making decision.

Appendix B: What is SOA?

Yet another attempt to define services!

SOA means different to different people as it can be applied in various degrees. This leads to various definitions or understandings among different communities like some listed by searchWebservice, webopedia. Various terminologies like service, service orientation, reusability of services, processes, registry, repository, service data objects, enterprise service bus, enterprise information integration, composite applications etc have made their way into SOA discussions.

However, least is bothered about the degree of understanding when such terminologies are used. Standard bodies like OASIS are working on the creation of SOA blueprints one of the outcomes which is to arrive at an agreement on such terminologies.

Among all the terminologies the most important and very basic unit of SOA which is service, the definition of which is still unclear or not agreed upon. Apart from the standard bodies, there have been attempts to define what a service is in various groups and communities like SOA community, SOA yahoo groups, Wiki, blogs and glossaries (IBM, BEA).

Since SOA is bringing both business and technology community together and traditionally both communities have had an orthogonal way of thinking about the supporting of business by IT systems, having two definitions for services (as applicable for each community) rather than a common definition, would be an interesting attempt.

So here I give another shot at defining what a service is!

Definition for a Business Analyst

A service is a common activity which when applied under various contexts over different business entities results in a business circumstance that is uniquely distinguishable.

Definition for a Technologist

A service is a self contained, replaceable and reusable module that exhibits high cohesion of functional/semantic relatedness of activities and loosely coupled through multiple standard interfaces and bindings.

How do you define a Service?
Steve Jones
March 01, 2005

I've been using various tools from different vendors over the past 12 months, all claim that they are for "Service Oriented Architecture" and yet none have an effective way to define what a service is.

Some, like BEA's Weblogic Platform or the unbelivably large install of WBI-SF 2GB for godsake, provide a software container (BeeHive for BEA and WS-IF for IBM) but these are really developer tools rather than architecture tools to define a service.

The objective of a *service definition should be to define a discreet boundary within a system,* this should then be de-composable into further services which can be co-ordinated together.

The key is that *service comes before process.* Most of these tools (especially ones like WBI-Modeller) stress a "Visual COBOL" approach to systems design where *the process is more important than the service....*

So when vendors say they do SOA, right now every one I've seen means "We do *Process Oriented Architecture",* it was wrong with COBOL, and its still wrong now.

What is SOA, really?
Uncle Bob
April 11, 2007

The good news is, you probably already know. The bad news is, you probably know too much. This article describes Service Oriented Architecture in a simple and easy to understand way that is devoid of buzzwords and vendor spin. It's the introduction to SOA that you haven't been able to find anywhere else.

There are things in a business that don't change very often. Gas stations in the U.S., for example, still sell gasoline by the gallon. Restaurants still sell meals from a menu. Dentists still sell cleanings every 6 months. Every business has these aspects that don't change very frequently. They often represent a huge part of the business. We'll call these things the _core business functions.

There are other things in a business that change very frequently. Prices, tax rates, catalogs, new products, new marketing campaigns, advertising, new business areas, new customer areas, etc. Indeed, businesses must be able to change, and change quickly, in order to survive. And yet, it is vital that those changes do not adversely affect the core business functions.

Software developers have known for years that software that changes frequently should be decoupled from software that changes infrequently. When applied to individual programs and systems this principle is sometimes called *The Common Closure Principle*. When it is applied to the information management of an enterprise, it is called SOA.

SOA is the practice of sequestering the *core business functions* into independent services that don't change frequently. These services are glorified functions that are called by one or more presentation programs. The presentation programs are volatile bits of software that present data to, and accept data from, various users.

To make this clear, imagine an internet store-front. Customers use a browser to talk to the presentation software that displays the store's website. The presentation software interprets the gestures of the customer and invokes services that do things like acquiring the data for the current catalog, or registering the customer's order. Note that the services have no idea they are talking to a website. They could just as well be talking to a thick client, or a 3270 green screen. They simply accept and return data in a standard format that the web system happens to be able to use.

That's really all there is to it. The rest of SOA is just a matter of details. At the highest level, SOA is nothing more (and nothing less) than separating changeable elements from unchangeable elements. But why is this important?

Consider that internet store-front again. It presents the user with a catalog, allows the user to move items into, and out of a shopping cart, and accepts the eventual order.

The presentation of these concepts is very volatile. Marketing people are likely to want to change it frequently. For example, they might want to change from a shopping cart metaphor to scrollable receipt on the sidebar. They may wish to present more or less descriptive data in the product list. They may want to experiment with different colors, font-faces, and layouts. Indeed, it's feasible that they'll want to try applets, JStart clients, Ajax, and a myriad of other presentation options. But none of this has anything to do with the core business functions encapsulated by the services. Those services that acquire catalogs and register orders remain unchanged despite all the presentation thrashing. That's why the separation is important. It protects the information processing assets of the business from the constant jitter and spin of the presentation.

But presentation is not the only thing that jitters and spins. So do the *business processes*. Again, consider our store-front. Perhaps our business has decided to offer fine wines as one of the products it sells. Selling alcohol requires that the age of the customer be verified. Let us say that we have a service that provides this verification. This service must be called for any order that contains alcohol products. The decision to call this service is neither a presentation decision, nor a service decision. Rather it is part of the business process for a particular kind of order. Business processes are volatile and they breed like rabbits. As businesses evolve they add more and more steps and forks to their business processes. The services being used by those processes don't change much; but the pathways through the processes do. Therefore we want to separate the business process from the services and from the presentation. Smalltalkers had a name for this separation when it appeared in a single program. They called it Model-View-Controller.

Notice that we have yet to mention even one of the plethora of technologies that are so commonly associated with SOA. That's because SOA is not about any particular technology. Rather it is a design philosophy that decouples well heeled business functions from volatile processes and presentations. It is the MVC of enterprise software.

In my next blog on this topic, we'll look at the next level of detail in an attempt to understand HOW services can be constructed, and how the decoupling of presentation, process, and functions can be achieved.

Service Oriented Ambiguity
Fowler M.
http://www.martinfowler.com/bliki/ServiceOrientedAmbiguity.html
1 July 2005

(Updated just to add a link to this wonderful moral support from David Ing - whatever you do please don't stop blogging!

Whenever ThoughtWorks rashly lets me out in front of a client, one question I'm bound to be asked is "what do you think of SOA (Service Oriented Architecture)?" It's a question that's pretty much impossible to answer because SOA means so many different things to different people.

- For some SOA is about exposing software through web services. This crowd further sub-divides into those that expect the various WS-* standards and those that will accept any form of XML over http (and maybe not even XML).

- For some SOA implies an architecture where applications disappear. Instead you have core services that supply business functionality and data separated by UI aggregators that apply presentations that aggregate together the stuff that core services provide.

- For some SOA is about allowing systems to communicate over some form of standard structure (usually XML based) with other applications. In it's worse form this is "CORBA with angle brackets". In more sophisticated forms this involves coming up with some form of standard backbone for an organization and getting applications to work with this. This backbone may or may not involve http.

- For some SOA is all about using (mostly) asynchronous messaging to transfer documents between different systems. Essentially this is EAI without all the expensive EAI vendors locking you in.

I've heard people say the nice thing about SOA is that it separates data from process, that it combines data and process, that it uses web standards, that it's independent of web standards, that it's asynchronous, that it's synchronous, that the synchronicity doesn't matter....

I was at Microsoft PDC a couple of years ago. I sat through a day's worth of presentations on SOA - at the end I was on the SOA panel. I played it for laughs by asking if anyone else understood what on earth SOA was. Afterwards someone made the comment that this ambiguity was also something that happened with Object Orientation. There's some truth in that, there were (and are) some divergent views on what OO means. But there's far less Object Ambiguity than the there is Service Oriented Ambiguity.

So what do we do? For a start we have to remember all the time about how many different (and mostly incompatible) ideas fall under the SOA camp. These do need to be properly described (and named) independently of SOA. I think SOA has turned into a semantics-free concept that can join 'components' and 'architecture'. It's beyond saving - so the concrete ideas that do have some substance need to get an independent life.

What Is Service-Oriented Architecture
Hao He
September 30, 2003

"Things should be made as simple as possible, but no simpler." -- Albert Einstein

Introduction
Einstein made that famous statement many decades ago, and it's still relevant today for building superior software systems. Unfortunately, as anyone who has been in the IT industry for long can point out, far too many software systems have failed Einstein's test. Some are made too simple to carry out the duties they are supposed to perform. Others are made too complex, and the costs of building and maintaining them have rocketed, not to mention the nearly impossible tasks of integrating different systems together. It seems that reaching the right level of simplicity is more like a dream than reality. Where have we gone wrong?

Loose Coupling
We don't have to look far to find the problems. As we build more and more software systems, we see similar situations and patterns appearing. Naturally, we want to reuse the functionality of existing systems rather than building them from scratch. A real dependency is a state of affairs in which one system depends on the functionality provided by another. If the world

only contained real dependencies, Einstein's test would have been satisfied long time ago. The problem is that we also create artificial dependencies along with real dependencies.

If you travel overseas on business, you know that you must bring power adapters along with you or your life will be miserable. The real dependency is that you need power; the artificial dependency is that your plug must fit into the local outlet. Looking at all the varying sizes and shapes of those plugs from different countries, you would notice that some of them are small and compact while many others are big and bulky.

The lesson here is that we cannot remove artificial dependencies, but we can reduce them. If the artificial dependencies among systems have been reduced, ideally, to their minimum, we have achieved loose coupling. In that sense, Einstein was just talking about was loose coupling. We might rework his famous principle thus: "Artificial dependencies should be reduced to the minimum but real dependencies should not be altered."

SOA Defined and Explained

Now we are able to define a Service Oriented Architecture (SOA). SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners.

This sounds a bit too abstract, but SOA is actually everywhere. Let's look at an example of SOA which is likely to be found in your living room. Take a CD for instance. If you want to play it, you put your CD into a CD player and the player plays it for you. The CD player offers a CD playing service. Which is nice because you can replace one CD player with another. You can play the same CD on a portable player or on your expensive stereo. They both offer the same CD playing service, but the quality of service is different.

The idea of SOA departs significantly from that of object oriented programming, which strongly suggests that you should bind data and its processing together. So, in object oriented programming style, every CD would come with its own player and they are not supposed to be separated. This sounds odd, but it's the way we have built many software systems.

The results of a service are usually the change of state for the consumer but can also be a change of state for the provider or for both. After listening to the music played by your CD player, your mood has changed, say, from "depressed" to "happy". If you want an example that involves the change of states for both, dining out in a restaurant is a good one.

The reason that we want someone else to do the work for us is that they are experts. Consuming a service is usually cheaper and more effective than doing the work ourselves. Most of us are smart enough to realize that we are not smart enough to be expert in everything. The same rule applies to building software systems. We call it "separation of concerns", and it is regarded as a principle of software engineering.

How does SOA achieve loose coupling among interacting software agents? It does so by employing two architectural constraints:

1. A small set of simple and ubiquitous interfaces to all participating software agents. Only generic semantics are encoded at the interfaces. The interfaces should be universally available for all providers and consumers.

2. Descriptive messages constrained by an extensible scheme delivered through the interfaces. No, or only minimal, system behavior is prescribed by messages. A scheme limits the vocabulary and structure of messages. An extensible scheme allows new versions of services to be introduced without breaking existing services.

As illustrated in the power adapter example, interfacing is fundamentally important. If interfaces do not work, systems do not work. Interfacing is also expensive and error-prone for distributed applications. An interface needs to prescribe system behavior, and this is very difficult to implement correctly across different platforms and languages. Remote interfaces are also the slowest part of most distributed applications. Instead of building new interfaces for each application, it makes sense to reuse a few generic ones for all applications.

Since we have only a few generic interfaces available, we must express application-specific semantics in messages. We can send any kind of message over our interfaces, but there are a few rules to follow before we can say that an architecture is service oriented.

First, the messages must be descriptive, rather than instructive, because the service provider is responsible for solving the problem. This is like going to a restaurant: you tell your waiter what you would like to order and your preferences but you don't tell their cook how to cook your dish step by step.

Second, service providers will be unable to understand your request if your messages are not written in a format, structure, and vocabulary that is understood by all parties. Limiting the vocabulary and structure of messages is a necessity for any efficient communication. The more restricted a message is, the easier it is to understand the message, although it comes at the expense of reduced extensibility.

Third, extensibility is vitally important. It is not difficult to understand why. The world is an ever-changing place and so is any environment in which a software system lives. Those changes demand corresponding changes in the software system, service consumers, providers, and the messages they exchange. If messages are not extensible, consumers and providers will be locked into one particular version of a service. Despite the importance of extensibility, it has been traditionally overlooked.

At best, it was regarded simply as a good practice rather than something fundamental. Restriction and extensibility are deeply entwined. You need both, and increasing one comes at the expense of reducing the other. The trick is to have a right balance.

Fourth, an SOA must have a mechanism that enables a consumer to discover a service provider under the context of a service sought by the consumer. The mechanism can be really flexible, and it does not have to be a centralized registry.

Additional Constraints

There are a number of additional constraints one can apply on SOA in order to improve its scalability, performance and, reliability.

Stateless Service

Each message that a consumer sends to a provider must contain all necessary information for the provider to process it. This constraint makes a service provider more scalable because the provider does not have to store state information between requests. This is effectively "service in mass production" since each request can be treated as generic. It is also claimed that this constraint improves visibility because any monitoring software can inspect one

single request and figure out its intention. There are no intermediate states to worry about, so recovery from partial failure is also relatively easy. This makes a service more reliable.

Stateful Service

Stateful service is difficult to avoid in a number of situations. One situation is to establish a session between a consumer and a provider. A session is typically established for efficiency reasons. For example, sending a security certificate with each request is a serious burden for both any consumer and provider. It is much quicker to replace the certificate with a token shared just between the consumer and provider. Another situation is to provide customized service.

Stateful services require both the consumer and the provider to share the same consumer-specific context, which is either included in or referenced by messages exchanged between the provider and the consumer. The drawback of this constraint is that it may reduce the overall scalability of the service provider because it may need to remember the shared context for each consumer. It also increases the coupling between a service provider and a consumer and makes switching service providers more difficult.

Idempotent Request

Duplicate requests received by a software agent have the same effects as a unique request. This constraint allows providers and consumers to improve the overall service reliability by simply repeating the request if faults are encountered.

Deriving Web Services from SOA

Everyone knows roughly what a "web service" is, but there is no universally accepted definition. The definition of web service has always been under hot debate within the W3C Web Services Architecture Working Group. Despite the difficulty of defining web services, it is generally accepted that a web service is a SOA with at least the following additional constraints:

1. Interfaces must be based on Internet protocols such as HTTP, FTP, and SMTP.

2. Except for binary data attachment, messages must be in XML.

There are two main styles of Web services: SOAP web services and REST web services.

SOAP Web services

A SOAP web service introduces the following constraints:

1. Except for binary data attachment, messages must be carried by SOAP.

2. The description of a service must be in WSDL.

A SOAP web service is the most common and marketed form of web service in the industry. Some people simply collapse "web service" into SOAP and WSDL services. SOAP provides "a message construct that can be exchanged over a variety of underlying protocols" according to the SOAP 1.2 Primer. In other words, SOAP acts like an envelope that carries its contents. One advantage of SOAP is that it allows rich message exchange patterns ranging from traditional request-and-response to broadcasting and sophisticated message correlations. There are two flavors of SOAP web services, SOAP RPC and document-centric SOAP web

service. SOAP RPC web services are not SOA; document-centric SOAP web services are SOA.

SOAP RPC Web Services

A SOAP RPC web service breaks the second constraint required by an SOA. A SOAP RPC Web service encodes RPC (remote procedure calls) in SOAP messages. In other words, SOAP RPC "tunnels" new application-specific RPC interfaces though an underlying generic interface. Effectively, it prescribes both system behaviors and application semantics. Because system behaviors are very difficult to prescribe in a distributed environment, applications created with SOAP RPC are not interoperable by nature. Many real life implementations have confirmed this.

Faced with this difficulty, both WS-I basic profile and SOAP 1.2 have made the support of RPC optional. RPC also tends to be instructive rather than descriptive, which is against the spirit of SOA. Ironically, SOAP was originally designed just for RPC. It won't be long before someone claims that "SOAP" actually stands for "SOA Protocol".

REST Web Services

The term REST was first introduced by Roy Fielding to describe the web architecture. A REST web service is an SOA based on the concept of "resource". A resource is anything that has a URI. A resource may have zero or more representations. Usually, people say that a resource does not exist if no representation is available for that resource. A REST web service requires the following additional constraints:

1. Interfaces are limited to HTTP. The following semantics are defined:

   o HTTP GET is used for obtaining a representation of a resource. A consumer uses it to retrieve a representation from a URI. Services provided through this interface must not incur any obligation from consumers.

   o HTTP DELETE is used for removing representations of a resource.

   o HTTP POST is used for updating or creating the representations of a resource.

   o HTTP PUT is used for creating representations of a resource.

2. Most messages are in XML, confined by a scheme written in a scheme language such as XML Scheme from W3C or RELAX NG.

3. Simple messages can be encoded with URL encoding.

4. Service and service providers must be resources while a consumer can be a resource.

REST web services require little infrastructure support apart from standard HTTP and XML processing technologies, which are now well supported by most programming languages and platforms. REST web services are simple and effective because HTTP is the most widely available interface, and it is good enough for most applications. In many cases, the simplicity of HTTP simply outweighs the complexity of introducing an additional transport layer.

SOA: Separating Myth from Reality

Using the facts to lead the SOA journey

[Mark Potts](#)

Jan. 28, 2006

There is still enough hype around service-oriented architecture (SOA) that it's difficult to get a handle on the realities of implementation. While most IT managers now understand what an SOA can do for IT, many still need help figuring out how to get started. New questions about management, reuse, and legacy applications are also emerging as more and more companies think about how to adopt SOA solutions.

The myths and mixed messages surrounding SOA hinder companies from starting on their path to SOA adoption, and in many cases actually prevent companies from realizing that some SOA principles may already be in place. Here are some of the questions that we have been addressing with our customers to help them move from thinking about SOAs to actually creating a plan.

What do I need to get started - services, software, hardware?

SOA is not a "rip and replace" strategy; rather, it is a way for enterprises to leverage existing investments, manage them better, and become more flexible and aligned with the real needs of the business. This journey is not facilitated simply by buying a new set of products or adopting new technologies. Realizing the value of SOA requires expertise and governance from the formal design process through implementation, provisioning, operation, and change - i.e., the entire life cycle. The vital starting point, therefore, is not technology, but rather identifying the business-specific services in an organization.

Due to the different focus - business-driven tops-down rather than technical bottoms-up - many customers require SOA expertise in the beginning of the process in order to map out a strategy that makes sense for the particular company and its goals. This is why so many large vendors like HP are offering services specific to SOA as a part of their offerings. The appropriate management tools that link the development life cycle to the operational life cycle become essential as an organization embarks on SOA-based projects.

How would implementing an SOA benefit my company?

A key benefit of adapting an SOA approach is the alignment of IT investments with a company's business strategy. As a result, companies can preserve legacy systems by exposing them as business assets without the need for replacement or significant modifications. SOAs also facilitate a company's ability to develop new business capabilities at lower cost, thereby meeting the needs of the business in the required time frames when change is driven by required change or new opportunities.

How difficult are SOAs to manage?

Because of the loosely coupled nature of SOA, an SOA implementation does bring additional management concerns to the forefront. Among them: managing the service integrations as well as the component parts; managing security across organizational or implementation boundaries; and managing policies for deployment, execution, and life cycles. Management is critical to realizing the business benefits of SOA and addresses the nonfunctional aspects of SOA, which are where the majority of the costs lie (security, configuration, financial, and QoS). Managing the nonfunctional aspects of SOA also extends to issues such as compliance

and IT governance, which are also critical as SOA is adopted more widely across IT and reuse and integration cross organizational boundaries.

If I am already using Web services, do I need to think about SOA?

There is a relationship between Web services and SOA, but the use of Web services does not constitute an SOA or deliver its benefits. SOA is an architectural approach, while Web services are an implementation choice for adopting an SOA, in which interfaces are based on standardized Internet protocols. The service-oriented approach does not limit interoperability to Web services - it can be implemented using any kind of service-based technology. In that way, full interoperability is only possible through an SOA, and it's the only way to fully realize the benefits of implementation.

Am I going to be able to justify this adoption from a business perspective?

The key to justifying SOA adoption is to realize that the business justification is not tied to the architecture, but rather to the ability to serve customers faster, address compliance requirements efficiently, and to become a more agile in supporting the business objectives and goals using IT. One starting point is to evaluate which business processes generate the most return on investment for your company and prioritize your SOA deployment to make these processes more efficient and more customer-aware. Also, because SOAs are not built upon a "rip and replace" strategy, investment is not geared toward buying more and more technologies, but instead leveraging existing assets and investments for better leverage, standardization, and governance across the business.

Now that I'm planning for SOA, can I incorporate legacy applications into the plan?

Because of the cost and inflexibility associated with legacy systems, they usually need to be "modernized" to be incorporated into an SOA. An evaluation will be needed to determine if the code should be refactored, modularized, or if the application could be replaced with a commercial application. Additional design work will then need to follow to build out a modernized application that is agile enough to integrate with other applications within the SOA.

Can SOAs solve all of my integration problems?

No. Ultimately, it is the information integration strategy underlying an SOA that solves integration problems. The SOA itself is not the true solution; rather, it's a better way to achieve a desired goal. Integration problems are solved through a number of actions related to adopting an SOA: carefully planning for an SOA, assessing business requirements, establishing governance, and enacting an underlying information integration strategy. It is only when these steps have been taken that an enterprise can expect to have its integration problems solved.

Are all SOA solution components completely reusable?

One of the major benefits of the set of architectural principles that define an SOA drives organizations towards a greater level of reuse and consistency. The amount of achievable reuse increases over time once newer services are designed using existing services in an organization. Determining which services are the most appropriate for reuse is where having insight into which business processes align the business goals most closely with IT is crucial.

In the end, IT professionals need to be armed with the most accurate information available to determine how to begin the SOA journey. While the complexity of SOA can raise questions,

it's worth the time to investigate how this architectural solution can bring new power to your enterprise.

What is SOA anyway?

Getting from hype to reality
By Arnon Rotem-Gal-Oz

Service Oriented Architecture or SOA for short has been with us for quite a while. Yefim V. Natiz, a Gartner's analyst, first talked about SOA back in 1996. However it seems that only in the recent year or so SOA has matured enough for real systems based on the SOA concepts to start to appear – or has it? There is so much hype and misconceptions surrounding SOA that we first have to clear them all up before we can explain what SOA is – let alone identify who really uses it.

This aim of this paper is to try to clear some of the fog surrounding SOA and provide a clear definition of the term

Service Oriented Ambiguity

Martin Fowler best described the confusion surrounding SOA in his Bliki in a post called Service Oriented Ambiguity. Martin says that the question "what is SOA?" is impossible to answer because SOA means different things to different people. While it is true that the term has been overly used, at least few of these "definitions" are plainly wrong. Let's take for example the definition that says SOA is exposing methods through Web-services.

SOA ═ Web Services?

A common misconception about SOA is that using web-services technology makes whatever you are using SOA. The core reason for that is the poor naming choice for methods that are exposed through http which were named web-**services**. For example one of the popular books on SOA "Service oriented Technologies: a Field guide to integrating XML and Web-Services" by Thomas Earl gives the impression that SOA equals WS* (even though that buried in the book, there's an explanation hat WS* is not the only possible technology for SOA). There are many other sources that give the same impression.

Nevertheless, Just a bunch of web-services (JBOWS as Joe McKinderik named them) does not an SOA make – in fact that's isn't really different from Remote Procedure Calls (RPC) using any other technology be that CORBA, DCOM or anything else. The way I see it saying that SOA is using web-services is just plainly wrong and not away to define SOA.

SOA ═ EAI?

Another point Martin mentions is that a widely used definition for SOA is "EAI without all the expensive EAI vendors locking you in". EAI emerged as an attempt to solve the Enterprise integration spaghetti. While it managed to solve the problem of connecting applications and transferring data between them through a semi-standard way.

I agree with Sandeep Arora who said that basically EAI failed to deliver its promise as it is

- o Data centric and not process centric.

- o Can't keep up with business process change.

- o Does not address the business process.

- o And that EAI solutions are technically very complex, need specialized skills and are very expensive to maintain.

Whether Arora is right or even if EAI is the best thing since sliced bread - If we do EAI using web-service technology to get "EAI without the expensive EAI vendors locking you in" (which I also seen called SOI – or Service Oriented Integration) – we are again not doing anything new, we are just using a new technology to achieve an existing way of thinking or architectural approach.

Ambiguity – is everything lost?

Martin claims that the SOA acronym is beyond saving and that the (sometimes) good ideas that are all called SOA need to have their own name and independent life. However the way I see it the examples Martin provides are "just" misuses of the term. I think that the fact that someone uses a term wrongly does not invalidate the correct term. Nevertheless, SOA troubles do not end here – since the hype also looms behind the corner to dilute the SOA term as well.

SOA Hype and Myths

I am sure many of you have seen that happen before – You are working on v0.9 version of project, making nice progress. Then, almost out of nowhere, steps out this marketing wiz-kid or wonder sales person who meets a client and tells him all about the wonderful features of the product (you were planning for v3.5), signs a deal and leaves you pick up the pieces o somehow try make all that work. Hype is like the power of thousand such wiz marketers working against all of us – making it almost impossible to live up to these "common truths" which are actually hollow promises.

Let's just take a quick look at few of the more common hype-originated myths surrounding SOA

SOA will make reuse easy

The claims that SOA will increase reuse and/or will make it easy is very common- so much that it has been adopted by serious technical people – for example you can see an article entitles "SOA: Separating myth from reality" (!) by Mark Potts who is a CTO in one of HP's divisions which states:

"One of the major benefits of the set of architectural principles that define an SOA drives organizations towards a greater level of reuse and consistency. The amount of achievable reuse increases over time once newer services are designed using existing services in an organization... "

Well, as object orientation thought us reuse is harder the larger the components we want to reuse. Since larger components have a more context they carry with them which make it hard to reuse them elsewhere. SOA is not different. Services should encompass a meaningful business capability, if you build it right you can *use* it to solve your business needs and you can integrate it to fulfill your business processes. The chances are however, that you will *not be able to reuse* elsewhere. SOA is more about business agility and the ability to change rather than reusing services out of their original context.

There is some substance to the reuse claim, if we take a broader definition of the term reuse. As an organization moves to SOA, it can make the transition an evolutionary process - and

still use existing assets in the transition phase. The evolutionary process approach, contradict another SOA hype generated myth which states that SOA requires an all or nothing approach.

SOA requires a big-bang approach
Consultants and technology vendors would have you believe that you either have to fully embrace SOA or it just wouldn't work. I don't think so. I agree with Sohel Aziz that it is actually the other way around, if you want to succeed with SOA you need a pragmatic step-by-step approach.

When you think about it, a Full Enterprise SOA initiative would mean replacing *all* the enterprise's systems. Do you really think the business will just freeze or halt while the SOA initiative is underway? Not quite. Rather, a comprehensive SOA initiative is like building a new interchange on a highway. You need to provide ways for the traffic to continue to flow (business to continue to operate) while work is underway. More so, you will make as much progress as you can without disturbing traffic.

Also SOA characteristics (which I'll expand more about later) can even allow you to take an incremental way for removing legacy systems by exposing existing functionality using new SOA interfaces before you actually replace the underlying system. Using SOA for

SOA will make integration easy
Another SOA myth is that it will make integration easy. SOA puts a lot of emphasis on the interface , which makes it "easy" to just make services talk to each other. On top of that SOA is (usually) based on standard protocols like XML, WS*, http and the like, so it is easy to just tie services together and integrate them. However, Fred Brooks told us ages ago.:

"The essence of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms, and invocations of functions. This essence is abstract in that such a conceptual construct is the same under many different representations. It is nonetheless highly precise and richly detailed. I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation. We still make syntax errors, to be sure; but they are fuzz compared with the conceptual errors in most systems".
SOA does not change these basic rules.

To properly integrate services you need to model their contracts in a way that will make them usable for multiple business processes – this is not a small feat and that is where you'd spend most of the time – this is still hard and SOA does not solve it. Once you solve that, SOA does help you lift some of the burden of doing the actual integration - but that's just making integration easier and not making it easy.

From myths to reality
Yes, it seems SOA is indeed in dire straits. In fact, I've almost managed to convince myself that Fowler is right and SOA **is** beyond saving. However as mentioned at the beginning over the past year or two we start to see systems –The reasons for that is that in fact we start to see some consensus on what SOA is and isn't. It seems that today there are two prevalent ways to look at SOA– from a business point of view and from a technical one.

And then they were two

Looking beyond the hype and misconceptions it seems we can narrow the field into two points of view for the SOA concept and both of them seem to be valid– one from the business perspective and another from the technical one.

Service Oriented architecture

At the enterprise architecture level, it is always about the business. This is not a bad thing, on the contrary, the enterprise architecture perspective should be focused on the business needs in order to make sure IT serves the business and not vice versa.

The emphasis from the business perspective is on "service orientation". Consider for example the SOA definition from Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology - Eric A. Marks, Michael Bell:

"SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. "Services" in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages."

Looking at other "business oriented" definitions of SOA we can see they follow the same reasoning. In a nutshell, they can be summarized as follows: from the business point of view SOA is about analyzing the business to identify business areas and business processes. Followed by defining services to represent these "areas" .Services expose their capabilites through message interfaces. The services can then be choreographed or orchestrated to realize the business processes. The goal of SOA is to increase the alignment between business and IT and achieve business agility – the ability to respond to changes quickly and efficiency. And then there's the other face of SOA – this time from the technical point of view

service oriented Architecture

While the on the business side of the fence the emphasis is on" Service Orientation" or SO - on the technical front the emphasis is on the A of SOA – Architecture. True, there isn't a single unified definition for SOA; however, just like the many definitions of software architecture, there are several characteristics that are more common and frequent than others. Looking at definitions of SOA such as the ones from Wikipedia, O'reily's, JavaWorld, Windley, Microsoft etc. you can see that SOA is commonly thought of as an architecture or an architecture style that builds on loosely coupled, interoperable and composable components or software agents called services. Services have well-defined interfaces based standard protocols (usually web-services but most definitions mention that it is not the only possible implementation) as well as QoS attributes (or policies) on how these interfaces can be used by Service Consumers. SOA definitions mentions the basic communication pattern for SOA is request/reply but many definitions also talk about asynchronous communications as well.

If we look at the business and technical approaches for SOA we can see that there's still hope to achieve convergence as there are some common grounds - the next section will try to do just that.

SOA defined

In order to be able to converge between the technical and business viewpoints we first need to differentiate between an architectural style and its application – once we define what SOA is we can apply it at an organization level to get an SOA initiative where services will encapsulate business function. However we can also apply SOA on a single project and get services whose content revolves around technical issues like security or management. We also need to differentiate between design goals such as loose coupling or business alignment and architectural building blocks and constraints like coarse grained services or policy based interactions

Lastly, if we look at definitions of other architectural styles like Client/Server, Layered or REST we can see that we can see that architectural styles are defined in terms of components, their attributes, their relations and the rules or constraints that govern them.

Based on that we can define Service Oriented Architecture as an architectural style for building systems based on interacting coarse grained autonomous components called services. Each service expose processes and behavior through contracts, which are composed of messages at discoverable addresses called endpoints. Services' behavior is governed by policies which are set externally to the service itself. Figure below shows the SOA componets and their relations:

Service The central pillar of SOA is the service. Merriam Webster defines service as "a facility supplying some public demand". A Service should provide a high cohesion and distinct function. Services should be coarse grained pieces of logic. A Service should implement at least all the functionality promised by the contracts it exposes. One of the characteristics of services is service autonomy. Autonomy means the services should be self-sufficient, at least to some extent, and manifest self healing properties.

Contract The collection of all the messages supported by the Service is collectively known as the service's contract. The contract can be unilateral, meaning a closed set of messages the service chooses to provide. A contract might also be multilateral or bilateral, that is, between a predefined group of parties. The contract can be considered the interface of the Service akin to interfaces of object in object oriented languages.

End Point The Endpoint is an address, a URI, a specific place where the service can be found and consumed. A specific contract can be exposed at a specific endpoint.

Message The unit of communication in SOA is the message. Messages can come in different forms and shapes, for instance, http GET messages (part of the REST style), SOAP messages, JMS messages and even SMTP messages are all valid message forms.

The differentiator between a message and other forms of communication such as plain RPC, is that messages have both a header and a body. The header is usually more generic and can be understood by infrastructure and framework components without understanding, and consequently coupling to, every message type.
The existence of the header allows for infrastructure components to route reply messages (e.g. correlated messages pattern) or handle security better (see Firewall pattern).
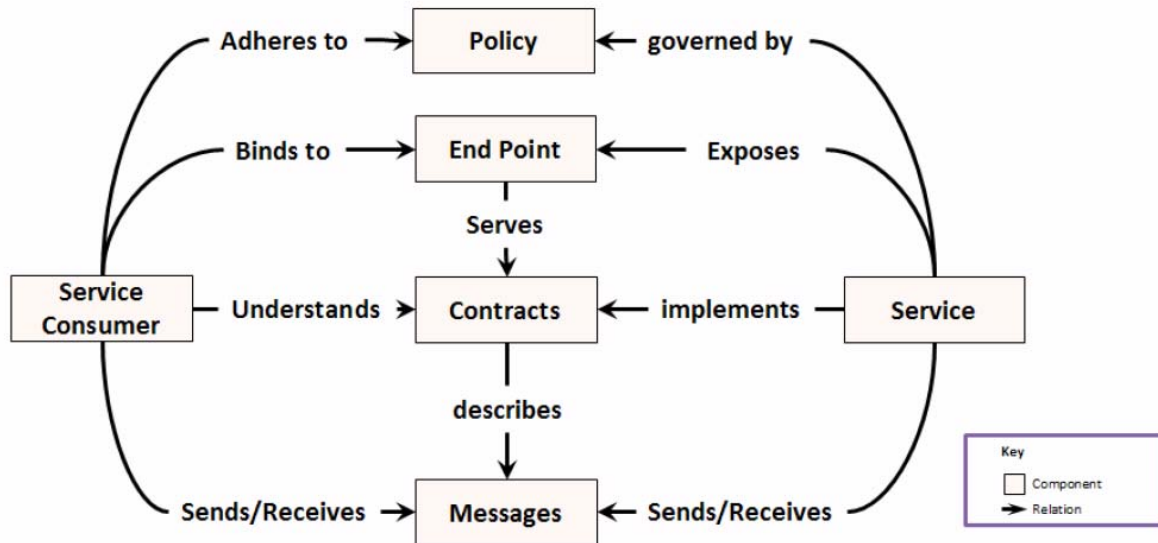
SOA components (Arnon, 2006)

Let's take a look at each of these components (excerpt from my upcoming SOA Patterns book)

Policy One important differentiator between Object Orientation or Component Orientation and SOA is the existence of policy. If an interface or contract in SOA lingo, separates specification from implementation. Policy separates dynamic specification from static/semantic specification. Policy represents the conditions for the semantic specification availability for service consumers. The unique aspects of policy are that it can be updated in run-time and that it is externalized from the business logic. The Policy specify dynamic properties like security (encryption, authentication, Id etc.), auditing, SLA etc.

Service Consumer A service doesn't mean much if there isn't someone/something in the world that uses it. So to complete the SOA picture we need Service Consumers. A service consumer is any software that interacts with a service by exchanging messages with the service. Consumers can be either client applications or other "neighbouring" services their only requirement is that they bind to an SOA contract.

Summary

Looking at this SOA definition we can see SOA has a lot of emphasis on interface. Starting from the messages which are the parts of the interface, the contract which is the collection of the messages, the endpoint where the contract is delivered and the policy which governs the behavior of the endpoint. Thus SOA has a total of four different components that deal with the interface vs., for example, OO which only has one.

The focus on interfaces is what gives SOA the ability to create loose coupling, composable components, reuse and achieve the various design goals. Another nice attribute of this definition is that we can use as a base for both the technical and the business perspectives of SOA as the common elements of both perspective are used in this definition. Thus, even though there are a lot of misconceptions and hype surrounds SOA There is value in the term. multiple definitions does not have to translate to ambiguity if they are just different workings for the same concepts. We do however have to be careful not to be fooled by the hype and misconceptions. I hope that the definition provided here helps achieve this goal.

Bibliography

Aziz, S. (2006). *Service Oriented Architecture: Look beyond the myths to succeed.* Retrieved from InfoSys: http://www.infosys.com/services/systemintegration/white-papers/SOA-look-beyond-myths-to-succeed.pdf

Brooks F. (1987, April) *"No Silver Bullet: Essence and Accidents of Software Engineering," Computer*, Vol. 20, No. 4 pp. 10-19 Retrieved from http://www.inst.eecs.berkeley.edu/~maratb/readings/NoSilverBullet.html

Fowler, M. (2005) *Service Oriented Ambiguity*. Retrieved from http://www.martinfowler.com/bliki/ServiceOrientedAmbiguity.html

He H. (2003) *What is Service Oriented Architecture*, O'reilly XML.Com Retrieved from http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html

Kodali R. R., *What is Service Oriented Architecture?* An introduction to SOA, Javaworld.Com Retrieved from http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html

Marks, E. A; Bell M., (2006 June) *Service-Oriented Architecture* (SOA): A Planning and Implementation Guide for Business and Technology, Wiley.

Microsoft (2006 Dec.) *Learn about Software Oriented Architecture* (SOA), Retrieved from http://www.microsoft.com/biztalk/solutions/soa/overview.mspx

Natis, Y. V. (2003, April 16). *Service-Oriented Architecture Scenario.* Retrieved from Gartner: http://www.gartner.com/resources/114300/114358/114358.pdf
Potts, M. (2006, Jan 28). *SOA: Separating Myth from Reality, SOA* World Magazine, Retrieved from http://webservices.sys-con.com/read/175375.htm

Rotem-Gal-Oz A. (2006). *SOA Patterns Chapter 1 (Draft).* Retrieved from Manning Publications : http://www.manning.com/rotem/free.html

Sandeep, A. (2005, Dec. 9*). EAI, BPM and SOA*.
SOA Institute Ord. Retrieved from http://www.soainstitute.org/articles/article/article/eai-bpm-and-soa/news-browse/2.html

Thomas, E. (2004). *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services.* Prentice Hall.

Wikipedia (2007) *Service Oriented Architecture*. Retrieved from http://en.wikipedia.org/wiki/Service-oriented_architecture

Windley, J. P. (2004). *Service Oriented Architectures.* Retrieved from The Windley Group: http://www.windley.com/docs/2004/SOA.pdf

Is SOA better kept in the dark, or in the spotlight?

Joe McKendrick

January 28, 2007

Should SOA be regarded as a fungus, growing out of sight in a dank, subterranean environment, or should it be brought into the spotlight, accorded the attention of a Hollywood movie?

I was pleased to join one of Dana Gardner's SOA recent [podcasts](#) ("the Gardner Gang"?), in which a number of fitting analogies for SOA were bandied about.

Perhaps the most amusing, but entirely appropriate, analogy was put out there by Jim Kobielus, who liken SOA to the world's largest living organism, a 50-square mile mushroom in Michigan.

As part of a discussion of SOA being similar to a "root system" for enterprise IT, Jim took the analogy a step further, observing that SOA is more like "a very complex hyper-mesh:"

"In other words, like a root system, where you have tendrils going hither and yon, the tendrils being simply interactions among services and client… SOA becomes this ubiquitous root system from which new sprigs can pop up, without needing to lay down their own root system. Rather they are simply branches on a huge underground system. In Northern Michigan, where I'm from, scientists have discovered the world's largest organism, as a mushroom or a fungus of some sort that spans 30, 40, or 50 square miles. They determined though DNA analysis that it's the exact same individual and has got the largest biomass in the world. In essence — and it's all underground pretty much. That's what SOA is all about, essentially all the services in an SOA sort of share a common DNA."

Not a pretty picture, and probably not the best way to sell SOA to the organization. (Kind of like packaging sushi as "cold, wet, dead fish.")   But it does make a lot of sense.

You could go to Michigan and lop off large chunks of the mushroom, but the other parts will continue to thrive, as it exists as a collection of parts that aren't dependent on one another.

Isn't that what SOA is all about?
Now for a somewhat brighter analogy for the SOA business model — the movie industry. Many smaller movie producers rely on a network of partners to deliver product. As Dana pointed out, a big monolithic enterprise with a command-and-control structure — such as MGM — can have its own infrastructure, and all of its own developers. "On the other hand, if you want to be a small company, a green-field software-as-a-service organization, you want to only be in an ecology play where you're acquiring things, and you're going to plug them in and then rip them out, reuse as much as you can," Dana pointed out.

"Then, you can be like a Miramax instead of an MGM. You're going to do just independent films. You're going to go in and spend a little bit of money, and maybe you'll still come up with some great product that will be right fit for your market."

Just as technology has lowered the barriers to entry for film production, the barriers to enterprise IT are also being lowered by SOA, Tony Baer added:

"The technology for producing films and videos has become much more accessible, and therefore you have now an independent film industry. So, that's analogous to the infrastructure that has become much cheaper, much more open, which is what SOA can do. In turn, you look at the business model. It's become more interchangeable, so that a studio might collaborate. Let's say that Universal might work with Sony — collaborate on a picture — because it's so costly that neither one of them could do it themselves. Well, they have the infrastructure and the business process in place that enables that to happen. Meanwhile, an independent film like "Crash" comes out of nowhere last year and wins the Academy Award for best picture of the year, and it's all possible because you now have this more accessible infrastructure. The moral of the story for SOA here, trying to bring this back on topic — is that if you get the plumbing down right — you don't have to be an SAP to introduce a killer app anymore."

What is Service-Oriented Architecture?

Service Oriented Architecture or SOA for short is a new architecture for the development of loosely coupled distributed applications. In fact service-oriented architecture is collection of many services in the network. These services communicate with each other and the communications involves data exchange and even service coordination. Earlier SOA was based on the DCOM or Object Request Brokers (ORBs). Nowadays SOA is based on the Web Services.

Broadly SOA can be classified into two terms: *Services* and *Connections*.

Services
A service is a function or some processing logic or business processing that is well-defined, self-contained, and does not depend on the context or state of other services. Example of Services are Loan Processing Services, which can be self-contained unit for process the Loan Applications. Other example may be Weather Services, which can be used to get the weather information. Any application on the network can use the service of the Weather Service to get the weather information.

Connections
Connections means the link connecting these self-contained distributed services with each other, it enable client to Services communications. In case of Web services SOAP over HTTP is used to communicate the between services.

The following figure is a typical example of the service-oriented architecture. It shows how a service consumer sends a service request to a service provider. After accepting the request, service provider sends a message to the service consumer. In this case a service provider can also be a service consumer.
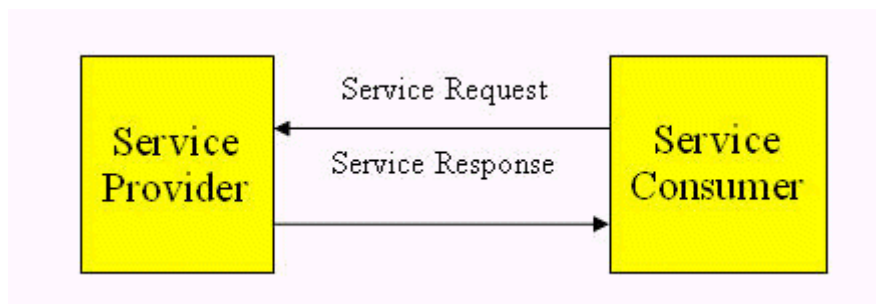
Different Technologies Used

SOA is much different from point-to-point architectures. SOA comprise loosely coupled, highly interoperable application services. These services can be developed in different development technologies (such as Java, .NET, C++, PERL, PHP), the software components become very reusable i.e. the same C# (C Sharp) service may be used by a Java application and / or any other programming language. WSDL defines a standard, which encapsulates / hides the vendor / language specific implementation from the calling client / service.

SOA definitions

(From http://en.wikipedia.org/wiki/Service-oriented_architecture)

Term	Definition / Comment
service	(Ideally) a self-contained, stateless business function which accepts one or more requests and returns one or more responses through a well-defined, standard interface. Services can also perform discrete units of work such as editing and processing a transaction. Services should not depend on the state of other functions or processes. The technology used to provide the service, such as a programming language, does not form part of this definition.
orchestration	Sequencing services and providing additional logic to process data. Does not include data presentation.
stateless	Not depending on any pre-existing condition. In a SOA, services should not depend on the condition of any other service. They receive all information needed to provide a response from the request. Given the statelessness of services, service consumers can sequence (orchestrate) them into numerous flows (sometimes referred to as pipelines) to perform application logic.
provider	The function which performs a service in response to a request from a consumer.
consumer	The function which consumes the result of a service supplied by a provider.
discovery	Service oriented architecture relies on the ability to identify services and their capabilities. Therefore, a SOA depends on a directory which describes the services available in its domain.
binding	The relationship between a service provider and consumer is dynamic; it is established at runtime by a binding mechanism.

Why SOA?

SOA architecture enables seamless Enterprise Information Integration. Here are some of the Benefits of the *Service Oriented Architecture*:

- o Due to its platform independence, it allows companies to use the software and hardware of their choice
- o There is no threat of vendor lock-in
- o SOA enables incremental development, deployment, and maintenance.
- o Companies can use the existing software (investments) and use SOA to build applications without replacing existing applications

The training costs are low, so the available labor pool can be used for running the applications

'Give businesspeople a reason to care about SOA; give them BPM'

Joe McKendrick

February 25, 2008

That's the advice given by Kaushal Mashruwala in an article just published in Financial Express. Kaushal makes a lot of sense, because BPM is a strategy that business executives and managers identify with very closely (as the success of their jobs depends upon it).

Will business process management make SOA more digestible?

SOA, as discussed many times at this blogsite, has issues with business acceptance — or even awareness, for that matter. As Jack van Hoof put it not too long ago: "I haven't meet one single business manager who begged me to please deliver him an SOA-based solution."

All too often, Kaushal points out, SOA is seen as a buzzword, and, to a large degree, "just another way to implement an application."

However, add BPM to the mix, and infusing it with SOA, business managers will have more power to change, through technology, the way their businesses are run, he observes.

"The management philosophy of BPM empowers business people to think about the processes that affect their day-to-day lives and operations. It gives them a new role in defining requirements, on their terms, and creates a common language for business and IT to address real implementation level concerns. This role of BPM as the business face of SOA is not just a possibility. It's happening now."

Perhaps we won't have to force the issue of fusing SOA and BPM, it may be occurring naturally. As posted a couple of weeks back (with a rousing talkback discussion), BPM, SOA, and Enterprise Architecture may be all the same thing underneath in the long run anyway. As Richard Lendvai commented, "It's architecture, period." He goes on to add that "architecture in general is about the enterprise's capability to work in an orderly manner, be specific in designs end to deliver results to the business. It has little to do with modelling, paradigms and other hype-stuff."

Appendix C: What is SOA Governance?

Wanted: new term for SOA 'governance'

Joe McKendrick

February 13, 2008

Here's a quote that's too good to pass up:

> *"Let's be honest, the term 'SOA governance' sucks. It reeks of someone else telling you what to do, hectoring you over every little detail of a project. It sounds about as desirable as a colonoscopy with an IMAX camera." -Michael Meehan*

Does 'SOA governance' sound too much like 'SOA politburo'?

I wish I could have come up with an analogy like this. In a new post, Mike looked at all the attention being heaped upon the topic of SOA governance and wonders if the term — not to mention the concept it represents — is too overbearing for our business culture?

"It's a particularly sticky term here in the U.S.A. We don't like a lot of governance. In fact, we get uppity when we think we've been placed under the yoke of too much governance. We'll dump your tea in the harbor when that happens. In fact, you can be sure many project teams have formed some unprintable thoughts about governance without representation."

In other words, does SOA governance have too much of a ring of "SOA Politburo"? (How's that for a better term?) The trouble is, that may be the perception some organizations have.

Mike's main beef is the term "governance" itself, of course, but I'm wondering if governance is being sold as a panacea for fixing any and all dysfunctional SOA attempts. Granted, ungoverned SOA would not be SOA at all - Just a Bunch of Web Services tangled up in a Spaghetti Oriented Architecture, with no clue as to what's being used and what value it's delivering. But with too much governance, as we've seen in the past, end users end up doing end-run around the rules with either sneaky approaches or all-new technologies altogether.

I've even heard of cases where SOA governance itself has tended to go too far, strangling the innovation that service orientation and loose coupling is supposed to promote. One vendor executive I recently spoke with said he saw customers pull back on governance when they realized that the restrictiveness stifled the ability to effectively deploy and reuse services. Besides, sometimes rogue services can desirable, and even profitable, too.

I have always thought that simply calling it "SOA management" fills the bill, anyway.

SOA in Context

Kåre Kjelstrøm

May 3, 2005

Most of the literature out there on Service-Oriented Architectures (SOA) addresses the issue at a level of abstraction, which implies that the author has not spent a day's honest work in actually applying principles to practical problems. For those of us who are faced with the challenge of not only talking the talk … this is rather frustrating.

In short what you can read out there is that SOA is a design principle for distributed systems, which governs the way peers expose functionality, connect, and interoperate. A SOA is composed of services, which are exposed by its peers, and that good, well-behaved services follow certain design principles.

- o **Stateless** - The service does not keep any state on behalf of the callee. This implies that all needed input is delivered in one shot and leads to the next principle.
- o **Coarse-grained** - Services deliver coarse-grained functionality. A service is typically not a simple getter or setter, but rather a high-level function.
- o **Loosely coupled** - Tight coupling makes versioning difficult tying the communicating peers so strongly that a change in one easily means a change in the other.
- o **Well-defined** - The interface of a service is well-defined, and ideally also immutable.

What you cannot read is what happens when you try to apply SOA principles to an organizational context as well as an inter-organizational context. At least, I haven't found a decent treatment of the topic, so here's a first shot.

SOA inside an organization

The IT infrastructure of a large organization, which has been around for some time and accumulated a bunch of legacy systems often has at least the following characteristics:

- o **Islands of functionality** - The IT infrastructure is composed of systems, some of which work independently, are not connected to common repositories, and which contain redundant information. Examples include local databases, ERP systems, specialized tools, even spread sheets.
- o **Not service oriented** - Depending on the system, it may be able to expose data or functionality through well-defined protocols, but this is often not the case. To integrate, you may have to write to files in proprietary formats, pump SQL through ODBC connections, open sockets and the like. And if you are really unlucky, the legacy system is sealed tight, integration-proof.
- o **Heterogeneous** - Systems were built over a stretch of time on different platforms, in different programming languages by different companies.
- o **Not supportable** - Once upon a time there was a programming language called COBOL. Unfortunately most of those who knew it are no longer among us, and the COBOL source for the system is only available on some obscure storage media for which there are no longer any functional drives …
- o **Centralized Control** - The IT department is typically in charge of running all the systems and hence it is easier to introduce supporting components like an Enterprise Services Bus (ESB) to the system.

SOA across organizations

When the organizational SOA needs to talk to the rest of the world or expose business services, it takes on the logical role of a single system. Just like the organizational SOA is composed of systems internal to the business, the inter-organizational SOA is composed of services external to the business. You may say that the inter-organizational SOA can be composed of organizational SOAs. The term business refers in this context to a company, an organizational unit or any entity with its own set of systems that can expose functionality.

The inter-organizational SOA is very different from the organizational one:

  o **Service-Oriented** - Most likely the services that are exposed to other businesses were provided with service-orientation in mind.
  o **Homogenous** - Web Services more often than not provide the plumbing for this kind of SOA and businesses are standardizing on this technology.
  o **Supportable** - If services can be published, chances are each organization is also capable of publishing new services, or alter existing ones (based on some contract).
  o **Decentralized control** - Businesses are autonomous entities that require contracts to cooperate. While one organization in such a SOA may take on the role as registry, service-police, etc. it is much more cumbersome to establish this inter-organizational infrastructure than it is for a local IT department to make decisions for all systems in the organizational SOA.

Conclusion

It seems to me that in order to qualify the discussion on Service-Oriented Architectures we need to extract lessons learned from applying this design principle to various distributed systems. What we will learn from going through the exercise within an existing organization, connecting legacy systems is very different from what we can do in an external SOA, built with Service-Orientation in mind. One interesting topic, which I'll pursue shortly is the role of the Enterprise Services Bus within the two SOA types and the challenges in applying such a beast.

Value dimensions of SOA explained

Architectural approaches such as Service Oriented Architecture (SOA) are transforming the way IT systems are designed by bringing in a high degree of reuse and loose coupling of applications. This opens up avenues for organizations to deliver their services in new and effective ways to their internal and external partners.

Value dimensions and measurable sub-dimensions of IT architectural paradigms such as SOA in enhancing business value is a key area of interest. Value dimensions are the ways in which value is instantiated. In the context of our discussion, value dimensions are the values or benefits which IT architectures present. Understanding the value dimensions of IT architectures is important as they typically involve large scale changes and such decisions need to be supported by the benefits which they generate.

Based on our research, we have identified six value dimensions of a services based architecture. These value dimensions are *organisational, business process, technology, standards, re-use and people*.

Explaining the value dimensions and examining the measurable sub-dimensions.

In the previous post we started off with defining the value dimensions of SOA and the need for understanding the value dimensions of IT architectures. In this post we shall go a little bit deeper into each of the dimensions listed. Expansion of the high level value dimensions into measurable sub-dimensions is important to capture some of the benefits.

Organisational value dimension has *time to market* and *support for emerging scenarios* as the key sub-dimensions.

Business Process value dimension has sub dimensions such as *modifying existing business processes* and *cost and time of introducing new business processes.*

Technology value dimension has sub dimensions such as *inter-organisational collaboration, loose coupling, reduced complexity and ease of integration, reduced cost and time of internal integration, partner integration (customers, vendors), reduced cost and time of introducing new applications, modifying existing applications, reduced cost and time of introducing new IT infrastructure, modifying existing IT infrastructure* and *scalability of systems.*

Standards value dimension has *benefits from low vendor lock-in* and *platform and technology independence* as sub dimensions.

Re-use dimension has *re-use of infrastructure, re-use of business models, re-use of processes* and *re-use of applications* as sub dimensions.

People dimension has *IT personnel efficiency* and s*tandardised employee skill-sets* as sub dimensions.

Appendix D: The differences between IT Governance and SOA Governance

Characteristics of IT Governance

According to Dr. Brad Wheeler [WHEE2005], IT Governance specifies the *decision rights* and *accountability framework* to encourage *desirable behaviors* in the using IT. It is not about making specific IT decisions but determining who systematically makes and contributes to those decisions. IT Governance also encourage and leveraging the ingenuity of the enterprise's people in IT usage and ensuring compliance with the enterprise's overall vision and values. Effective IT Governance means an actively designed set of IT Governance mechanisms that encourage behaviors consistent with the organization's mission, strategy, values, norms, and culture.

Another definition of IT Governance is addressed by TOGAF 8.1 [OPEN2007], IT Governance provides the framework and structure that links IT resources and information to enterprise goals and strategies. Furthermore, IT Governance institutionalizes best practices for planning, acquiring, implementing, and monitoring IT performance, to ensure that the enterprise's information technology assets support its business objectives.

Effective IT Governance must answer the following questions;

- What decisions must be made to ensure effective management and use if IT?
- Who has the authority to make these decisions?
- How will these decisions be made and monitored?

These decisions should cover the area of IT principles (role, models, behaviors and funding), IT architecture (technical issues and standards), IT infrastructure (shared services and resources), Enterprise applications (requirements and decisions) and IT investments (spending, approval and monitoring).

Characteristics of SOA Governance

- Flexible authority structure
- Management incentives
- Full operational lifecycle
- Decentralize vs. Centralize
- Local vs. Global governance
- Strategic vs. operative decision

The differences between IT Governance and SOA Governance

According to the characteristics of IT Governance and SOA Governance mentioned above, IT Governance is broader and covers all aspects of IT Governance, for instance, it includes data governance and IT security. While SOA Governance addresses aspects of the service life cycle such as planning, publish, discover, versioning, management and security. The difference between IT Governance and SOA Governance is that while SOA Governance contains aspects of IT Governance, it also covers business aspects that are not captured in IT Governance such as the linkage required between business and IT [IBMa2006].

Appendix E: Reusability vs. Business Agility

Is SOA still of value if nothing gets reused? How about if everything gets reused?
Joe McKendrick
March 4, 2008

What if you built a service-oriented architecture and nothing got reused? Is it still of value to the business, or is it a flop?

Reuse may be the means, but not the end

Ask many experts, and the answer will be a straightforward, yes, SOA will deliver value to the business in multiple ways beyond reuse of services. (Others will say it doesn't, but that's a subject for many other posts.) Decreased infrastructure redundancy and increased time to market are two big areas where SOA has potential to deliver.

However, there are three sticky questions around reuse: First, *should reuse be a goal in itself, or does it play more of a supporting role to more business-focused higher-level benefits?* Is it like trying to measure the number of times employees open up Excel spreadsheets through the day, versus measuring the insights and actions they take as a result of the data they get out of their spreadsheets?

The second question is even more vexing: *is reuse even essential at all to SOA success?* Suppose there are services that are only used by one application each? These services may offer streamlined interfaces that provide independence from the application underneath, saving countless hours of toil and disruption when the app changes.

Third, suppose *business units across the enterprise go wild with reuse, to the point where it gets difficult to track who is using what and how often?* The SOA appears to be a raging success, but how is it helping the business? Is it amounting to anything? Is so, how can that be measured?

AMR's Ian Finley said too much emphasis is being put on reuse as a value driver, as observed in a follow-up interview on [AMR's recent SOA study](#): Another danger seen from the SOA survey is that the main benefit that the vendors sell around SOA — code reuse — is not the real benefit that early SOA adopters have gotten. Often the code from project A is irrelevant to project B…. That focus on reuse can cause organizations to dismiss SOA's benefits because they're looking at the wrong metric."

Dave Linthicum has been warning companies against adoption of reuse as a value metric for some time. Most recently, in [response](#) to Finley's statement, he bluntly added:

"The core issue is that reuse, as a notion, is not core to the value of SOA…never has, never will. Not that you won't achieve reuse, and that there is benefit, but that the value of agility, or creating an architecture that's changeable around the needs of the business is far more valuable than any services you can share.

To the point of this post, people chase SOA understanding that reuse is the core value. Thus, when it's not they consider SOA a failure…. We need to stop selling reuse as a core benefit of SOA."

James Taylor, who has been blogging prolifically from Dialog 08, reports that he had a chance to break bread with IBM's Sandy Carter and ILOG's Pierre Haren, and the subject of service reuse came up.

James notes that Pierre said that *the main value of SOA today is in collaboration and understanding not in reuse.* Sandy, who had some examples of customers getting a lot of reuse, "generally supported Pierre's point, that reuse is not essential to the SOA value proposition. After all, many previous architectures promised reuse and it never seems to get delivered."

The consensus appears to be that "reuse may well come, whether through reused services or reused rules between services, but the power of SOA to bridge the business and IT is key," James observes. The question becomes how to measure the impact of that bridge.

Survey: companies investing millions in SOA, but don't exactly know why
Joe McKendrick
February 26, 2008

AMR Research just released snippets of its latest survey on SOA spending trends, and finds big money is flowing — but many of the companies spending the money may not exactly know what they're investing in.

The typical company adopting SOA spent $1.4 million on software and services in 2007, AMR estimates. AMR also said it found that SOA adoption is broad based and growing rapidly—China, Germany, and the United States all showed adoption growth rates of over 100%.

However, while the money for SOA will keep flowing through 2008, an interview with the survey's author reveals that there may be little rhyme or reason to the spending. "Hundreds of millions of dollars will be invested pursuing these markets in 2008, much of it wasted," said AMR analyst Ian Finley, quoted in InfoWorld.

Why is the money being wasted? Finley says there is not single driving focus for SOA. Instead, companies end up investing in SOA for a range of reasons, often unrelated individual priorities.

The survey found that the primary drivers for SOA investment were to meet the need to change investments faster, cheaper, and with less risk (22%), to meet requirements of individual projects (18%), and to reduce IT costs through reuse (17%).

While code reuse ranks as a reason to go with SOA, Finley doesn't see it as the ultimate advantage of SOA. Rather, the changed mindset that SOA brings to development and management is the real value — a value hard to quantify, of course. In addition, agility — through faster time to market — is the benefit early adopters are discovering. However, improved agility is also hard to quantify.

Appendix F: Failure of SOA

[Mike Kavis](#) SOA for the sake of SOA means failure

SOA for the sake of SOA means failure. SOA to assist with business process reengineering is where most of the value is. I agree that 80% will fail. 80% of all types of projects fail because of bad project management.

[iTech-Ed](#) (CEO): SOA - Same Old Architecture
September 4, 2007

Last week I blogged about a session at a legacy application modernization session I attended. This week I'd like to tell you about another presentation I saw later that same day. This second one was by Gary Barnett, Research Director at Ovum Consulting.

His approach was less one of telling us what to do, but rather raising our consciousness to stop us making the same mistakes that other people have made in the past. He is responsible for defining [SOA](#) as Same Old Architecture - which, although intended as a joke, made the point that this isn't all new. He reminded us that Web services weren't the first type of services that we'd come across. He suggested that we'd looked at work in terms of services before, with things like CORBA services and Tuxedo services (from BEA).

Gary also confidently predicted that 80% of SOA projects would fail. He based this prediction on the fact that they relied on ASCII and XML and that 80% was probably the number of projects that failed anyway.

He also had some important thoughts on re-use. He suggested that it wasn't enough simply to have nice interface. He insisted that if re-use was to occur it had to have been planned since the design phase. There is no way to retro-fit re-use! He also insisted that "best practice" only worked when it really was "practised"!

Gary likened many IT projects to building a bridge. IT people know how to build metaphorical bridges, so when someone says let's have a bridge the IT people start building. The reason so many projects fail is because it is not until they are half way across the river that anyone from IT stops to ask the questions, "just how wide is this river?" or, "do you really want the bridge here?".

Gary said that most presentations show large coloured squares joined by thin lines and warned that the reason the lines were so thin was that people didn't want anyone to notice them and ask questions. However, he stressed, it is often the links between applications or services that are the most difficult to modernize.

On a serious note, Gary insisted that the focus for change should be on [business processes](#). He said that in any successful company there would be no such thing as a legacy system modernization project, there would only ever be business modernization projects.

Definitely a "make you think" session, and well worth seeing for anyone contemplating modernization (i.e. all of us!).

David Linthicum: Why SOA Fails
Living service oriented architecture, one day at a time
November 18, 2005

I ran across this blog...a recent study that shows that about 1 in 7 SOA projects are not working out. Why? I think I know.

In doing research for my next book, I attempted to determine the top 5 reasons SOAs fail. Here's what I found:

      1. The enterprise considers SOA a project versus what it is; a more holistic notion. Thus, management thinks they can implement an ESB or other SOA technology, and they're done. I've been ranting about this enough that I won't do it here. However, just to recap, you need to determine your requirements first, and then the technology that works to solve the problem. Common sense, one would think, but more enterprises are failing this way, or will fail this way.

      2. They use 2nd tier talent. They attempt to implement SOA using people who really don't understand the concept of SOA, and perhaps never will. We know who these people are. They must be stopped. It takes education to get this stuff, and those who lack the education yet are making critical decisions are down right dangerous.

      3. They are under-resourced. "Go make huge sweeping changes within our IT infrastructure, and do so with about 10 percent of the resources you really need." You can't do this on a budget, unfortunately. You have to lubricate SOA projects with money.

      4. They allow the vendors to define their solution. The vendors don't understand your core business issues, and really have a conflict of interest. When you sell a hammer, everything looks like a nail. Work with vendors, make them understand your requirements, but you're ultimately responsible for the solution.

      5. Requirements are not fully gathered. There is no domain understanding before the solution is developed. I've been ranting about this as well; I'll leave it alone. Check out my 12 Steps. Use them, please.

Tom Sullivan: Top 5 reasons why SOA fails
November 18, 2005

SOA: Sixty percent of respondents in a new survey agreed that their SOA efforts have failed and David Linthicum knows where IT's pain originates. Linthicum, in fact, lists the top 5 reasons SOA fails as determined by his own research.

Notes from the field: Giving new meaning to the compound word "housewife," Robert X. Cringely opines about Denverite Deborah Hale who is tapping eBay as a means to auction her house, replete with everything -- furniture, appliances, games, exercise equipment, tanning bed, lawn tools and even Deborah herself. Of course, for almost half the cost of the house you, like Cringester Tom M., could instead invest in a Dell Axim X50 case from DealTime.com. That's $266,250. No Norwegian blonde included, though. And there's more from Cringe in Murdoch auctions a snack, MSN rejects IE for Mac.

Best of the blogs: Jon Udell celebrates [screencasting's first anniversary](#), and Bob Lewis offers up [three tips for CIOs looking to engage contractors at a fixed price](#).

The news beat: A Sun Microsystems executive calls for [a Net summit of engineers](#), the city of [Paris accelerates its move to open source software](#), and Yahoo appoints [a new head of 'emerging search.'](#)

Steve Jones: [Why SOA won't fail except for people who miss the point](#)
April 03, 2006

Reading [Dan Creswell's weblog](#) and his recent post on "[SOA definitely doomed](#)" and a couple of replys (including me) I think SOA is heading towards the "trough of dissolution" which is great news. Why? It means that the shiny happy technology wave is going to crash soon and we can get on with what SOA is really about this means

1. SOA is *not* about technology, not ESBs, Web Services, EJBs, Spring, .NET, Java
2. SOA is *not* about RPC, Messaging, Events or anything else
3. SOA *is* about changing the way you *think* about applications and enterprises

First off Dan goes after the "we don't know what SOA is" - We now have SOA RM, not everyone is onboard yet but it's a start.

This is the shift that SOA represents and is the thing that *will* continue on. Undoubtedly there will be people who don't do SOA in the future, hell there are people who keep doing waterfall today (and failing as a result), and there will be people who *say* they do SOA in the same way as people *say* they do agile today when all they are doing is traditional iterative.

SOA is an important shift that has been used successfully in massive systems for many years, and which is leaking through to the big, middle and small mainly via vendor product marketing, in a similar way that OO really made the break through in the 90s, with various companies pushing "their" take on OO, particularly OOD.

The rest of Dan's issues are focused around the development of SOA systems, which is part of the challenge for anyone trying to understand SOA. The vendors' products are NOT about architecture they are about delivery, but SOD of IT was never going to catch on. Dan's points on the confusion in this technology area are very similar to those in the 90s where people argued C++ v Smalltalk v Eiffel etc. and lost the fact that the important shift was the change in thinking, not in technology.

His point about complication is well made, SOA is aiming at a more complicated challenge, its aiming to create a common language between diverse stakeholders. But then at the moment we have nothing with which to communicate between these groups effectively and that cannot be allow to continue. But I remember only 5 years ago being told that business people would never understand Use Cases, and now it becomes a natural way to discuss requirements.

SOA is about a simple principle, making your IT look like your company and enabling it to change in the same way as your company. This isn't about STOPPING small agile projects, it's about enabling them, but enabling them in a way that doesn't make them the maintenance nightmare of tomorrow.

OO has failed in the large, its time to try an approach that works. For the record Dan could definitely kick my arse. And for an example of the OO wars of the 1990s, here is a [cracker from the inventor of C++](#) and a [troll from 1997](#) and back in [1994 people were not sure what OO was about](#)

Gartner: Why SOA deployments fail
June 27, 2007

Organisations that embark on service-oriented architecture (SOA) initiatives aimed at enterprise wide deployment must pay equal attention to technical and governance issues. Gartner said that although the risks of SOA project failure are initially associated with bad technical implementations, risks of failure due to insufficient governance are becoming increasingly significant, as SOA scope expands. Gartner analysts identify hit lists of the top tech and organizational SOA mistakes -- a total of 11 errors that can signal long-term failure.

Organisations that embark on service-oriented architecture (SOA) initiatives aimed at enterprise wide deployment must pay equal attention to technical and governance issues. Gartner today said that although the risks of SOA project failure are initially associated with bad technical implementations, risks of failure due to insufficient governance are becoming increasingly significant, as SOA scope expands.

"Actual implementations are showing that SOA requires more investment in service design governance and application integration best practice than current levels in most organisations," said Paolo Malinverno, research vice-president at Gartner. "At the beginning, risks of project failures are small but as the SOA project develops the risk curve increases. For this reason, organisations should never think of SOA without establishing a set of governance processes around service definition, implementation and maintenance."

However, enthusiasm for SOA and its anticipated benefits results in some companies taking risky shortcuts in establishing robust governance, service development disciplines and staffing. Gartner predicts that by 2010, less than 25 percent of large companies will have the sufficient technical and organisational skills necessary to deliver enterprise wide SOA.

"Technical risks should not be under estimated either," said Massimo Pezzini, vice-president and distinguished analyst at Gartner. "The ease of use of modern SOA enabling tools hides the technical complexity of implementing a reliable SOA technology platform, but developing an enterprise-wide reliable, scalable, high performance, secure and manageable SOA infrastructure requires a level of technical command that few organisations have been able to develop."

According to Gartner these are the areas where mistakes are being made by IT operations and application managers when planning SOA implementations.

Gartner's 'hit list' of the most common technological errors includes:
- Underestimating the technical complexity of a large-scale SOA

- Bad selection of application infrastructure components (ESB, orchestration and adapters)

- Insufficient validation of the SOA enabling technical infrastructure implementation (for example, no proof of concept and no stress tests)

- SOA infrastructure, services and consumer applications are insufficiently instrumented for security/management/troubleshooting

- Too-coarse/too-fine service granularity

- Insufficient/not up-to-date documentation

Gartner's 'hit list' of the most common organisational errors includes:

- Overlooking governance

- Thinking an SOA project should be organised just like any other application development (AD) project

- Not anticipating service number explosions in a maturing SOA

- Giving up on an integration competency center or SOA center of excellence

- Outsourcing architects (or not having them at all)

"In order to avoid the most common technical implementation mistakes, we recommend that organisations design their SOA technical infrastructure on the basis of their real functional and non-functional (e.g., performance, availability and security) requirements and not on the basis of theoretic models. Selecting proven and referenced SOA infrastructure products is also vital," said Mr Pezzini. Organisations must also architect their SOA infrastructure so that it can be easily monitored and provide all the information required to debug SOA applications.

"Finally testing is critical and at least 25 percent of the effort in a SOA project should be dedicated to this activity," he added.

From an organisational point of view, there is no "one size fits all" approach governance. "Too little or too much governance will kill an SOA project, companies need just enough governance," Mr Malinverno said.

When looking at their governance arrangements, organisations need to ensure that their governance arrangements are not too sophisticated and disproportional to their company size, organisation and culture. They also need to realise that they can not do without an integration competency centre (ICC) or SOA centre of excellence (CoE).

Joe McKendrick: [How do we really know when SOA 'fails'?](#)
August 13th, 2007

There's no place to hide when a huge software project hits the skids. Lots of money is invested, but things just don't work. Fingers of blame are pointed in all directions. The business kept on changing the specs for the application. The software was crapware. The vendors acted like schmucks once the ink was dry on the contract.

In the meantime, end users are ready to break down the doors to the data center. Vendors are called on the carpet. Rollbacks to previous versions are made. There are plenty of legendary stories of painful ERP, CRM, and operating system installations that didn't quite make it.

Such is the way of IT failure. *But what happens when an SOA project 'fails'? For that matter, how do we even know when a SOA project fails?* Frankly, I haven't heard about any big failures… yet. But for most companies, it's still too soon in the evolution to SOA to make the call.

In one of our recent discussions on Dana Gardner's SOA Insights panel, we chatted about the [meaning of SOA failure](). Of course, when a project doesn't meet projected ROI, it can be considered to have failed. But SOA is a long-term journey that may not deliver ROI for years.

What does SOA failure look like? Lots of money spent on development with no return to date? No or little reuse or sharing of services, so they end up sitting as virtual shelfware? (Failure by obscurity.) What happens then? Does the SOA get ripped out of the infrastructure and everyone starts over? Do the previous stovepipe systems get rolled back in? Probably not.

In our podcast discussion, I posited that components of the SOA still may stay in place and just be sent off in a different direction. I might add that perhaps the major failing of SOA thus far is when an organization throws some Web services in place, and thinks it has a service-oriented architecture.

Fellow panelist, the insightful Jim Kobelius, recently posted his ruminations about the [meaning of SOA failure](), evoking a metaphor Miko Matsumura introduced, comparing a failing SOA to "airplanes in danger of failing hence going into tailspins." Miko also cited an example of a CIO that outsourced his company's applications via SOA-style interfaces, and ended up being canned because the outsourcer evidently screwed up. But this sounds like more of an outsourcing management issue, rather than an SOA failure. Such is the case on many occasions — there are other factors at work.

Jim defines SOA failure thusly:

"SOA failure is the failure of SOA as a set of practices that a company adopts, the company's failure to realize the grand claims made for SOA…. An SOA project or initiative is a failure if it increases the complexity of your environment, if it increases cost, if doesn't make much of a dent in the incompatibilities among different platforms, or if it locks you into a given vendor."

Since SOA is a long-term undertaking, it's unlikely ROI will be pronounced within the first few years, and costs will be higher. The organization may be spending more money than it saves or makes during the formative period, when there may be just a few services developed, shared by a limited number of business units.

Of course, the opposite could occur, with SOA delivering nice ROI its in early years, then seeing diminishing returns. When, exactly, can an SOA project be called a success or failure? Dave Linthicum talked about this last year, in an effort to [pinpoint the productive lifespan of a particular SOA project](). (He even provides a formula for making this determination.)

But, again, SOA success or failure is probably harder to pinpoint than other types of projects. Here is a brief summary of some SOA points of failure. Note that these are not fatal career-terminating failures. No major ripping and replacing required; simply a change in direction.

- *Lack of a true SOA*. An organization may only *think* it has SOA. Having a primordial soup of services across the organization and hoping an SOA may evolve from it at

some point. Worse yet, it ends up as a "service-averse architecture," which is an architecture that is built without having first consulted the people who will use it, or is so secure and complex that it discourages people from using it comfortably. (Or, conversely, so insecure that its data is compromised, corrupted, lost, or stolen.)

- *Lack of reuse or sharing by multiple business units.* Such services may languish until a more effective governance strategy can bring them into the light of day. Or, sharing may actually be happening, but the business is not tracking these metrics.

- *More money spent than gained — over the long run.* Increased costs are to be expected in the early phases of SOA. The other issue may be that the business may not be adequately tracking where gains, if any, are taking place. So SOA proponents may not even know if the SOA is delivering, beyond anecdotal evidence.

- *More, not less, vendor lock-in.* The purpose of SOA is independence from vendors, so components or services can be swapped out as needed. If you still have to wait for a vendor to improve upon a particular part of an application, or are forced to upgrade, then SOA is not doing what it's supposed to be doing. (Or it's not a true SOA.)

Joe McKendrick is an author and consultant with deep knowledge and insights regarding trends and developments in the technology industry. See his full profile and disclosure of his industry affiliations.

Dana Gardner: BriefingsDirect SOA Insights analysts explore SOA's role through failure
Governance, policy and politics
March 25th, 2007

Follow service oriented architecture (SOA) to its logical conclusions and you recognize that modern corporations will soon be operated on the equivalent of aviation's "fly by wire." The traditional governance and management means of running a business — the levers, pulleys, cables, and brute muscle — will through SOA become more automated, rules- and event-driven, self-service, pre-programmed, policy-orchestrated … agile.

Instead of directing a business on how to function from the board-room megaphone, with explanations and edicts, countless meetings, and then reviews and crass incentives, there may soon be policy-driven decisions on how to execute made on a more federated basis — collaborative and productive by making IT not just the means of operating the computer applications, but making IT the means through which to operate the very business itself.

Sound far-fetched? Consider that whomever controls the full-fledged SOA to a large extent controls the company. So how should that control actually work? Will companies take a lesson from world history on how to run the business and allow for federated and balanced power? Or will mismatched control over business elements, exacerbated by IT that can not reflect the will or wills of the controlling factors, drag productivity and the company down? To fail at SOA is to fail at modern business?

Our assemblages of analysts and guests have some unconventional and startling conclusions, as well as thoughtful insights. With that, welcome to the latest BriefingsDirect SOA Insights Edition, Vol. 11, a weekly discussion and dissection of SOA-related news and events, with our panel of noted IT industry analysts. Join experts Steve Garone, Joe McKendrick, and Jim

Kobielus — along with guest Miko Matsumura, the vice president of SOA products at webMethods — for our discussion, hosted and moderated by me, Interarbor analyst Dana Gardner.

We start the discussion on what contributes to SOA failures, and the need for operational cultural transformation to accompany any meaningful advancement in SOA. We then examine federated approaches to balancing governance and control for business as well as … yikes! … governments. We conclude that SOA "failure" is probably necessary and a good, as it shows momentum — as long as the organization deals with failure maturely and constructively. No quitters!

Listen in: There are lessons from world history on how to run your business, and on how to use IT to define, balance and invoke power. Roll over Clausewitz.

Here are some excerpts:

On Failure in SOA
In 2007 we're as likely to see catastrophic failures as we are limited success. There are a huge number of moving parts within SOA, and I'm going to use that almost as a handout point to this very well-considered group of folks. We need to categorize for the listener which moving parts are more dangerous than other moving parts, because those are the things that eventually cause the thing to kind of wiggle the wrong way, and send it to a tailspin.

Corporate backing … is more focused on the people-oriented things and the collaborative issues associated with deciding what to build and how to build it [than technology].

The most dangerous moving parts — are people. From our perspective, the system is sort of cybernetic, half-human, half-machine. The human pieces of SOA are the parts that we've seen in failure mode. It's not necessarily just the human beings themselves [but] … the interfaces between the human world and the machine world, whether those interfaces are the specifications used to design applications, or the mechanisms used to manifest constraints … [Do] people, when they do fight each other, fight each other in a way that's productive, as opposed to destructive.

When projects are pitched … [as if] we're going to totally clean up our development practices and our integration practices … you're just setting up the SOA project for failure.

A Battle for Control Over Who Controls SOA
The people who control the SOA are the people who essentially control the policies. The policies include metadata, repository, and registry — the kind of policies that are machine-enforceable, but also involve human factors. In a way, the model is more of an equal partnership now. On the other hand, system integrators (SIs) like to control policy as a way to permanently set up a base-camp inside an account, pour people through the door, and take over. It's something that we know they're salivating about.

A CIO had this mistaken impression that the service interface abstraction allowed him to outsource completely the operational concerns and the implementation concerns, and eventually to treat this service interface as something like a child's car seat, where really mom is driving.

It's important to treat the interface abstraction layer like a saddle on a horse, which means that the only people who can successfully get from Point A to Point B are the people who

have the skill of riding and controlling the horse, which is the service implementation. It's really an abstract or complicated metaphor. It's not hard to lose control.

[Customers said:] "We don't want a single vendor to come in with a product and a set of services, because we don't want them to control everything. We want an independent to mix things up." There is a very significant danger of the inmates running the asylum or the integrators taking over the whole account from the inside.

SOA Fails When Control Over Change is Monopolized?'
Is SOA a democratization type of an effect, or is it really giving command-and-control through policies that you could think of as a governor or an accelerator — a brake-pedal/dashboard type of an affair — where suddenly those in the organization that may not have had power before gain it? Is the failure when the control doesn't go to the right people?

This question is basically The SOA Question, because the people who control the policy metadata are the people who are running the show. The thing that we're trying to establish here is that the SOA success model is essentially a model where there are federated controls and delegated controls. The reason why this term "federation of control" is so significant is because we're trying to achieve a balance between the central function, the IT function, and the distributed function, or the business function.

If you want to balance these things, you need a mechanism that enables some amount of control by the people who are on the periphery, in the business units, trying to create agility. Then, [there comes] some amounts of control by the people in the center, who are trying to create more orthodox standardization and security and orthogonal cross-cutting concerns. Having the wrong people controlling the wrong things is exactly the pattern that causes things to go a little nuts.

The extent to which your SOA initiative and your SOA governance are totally centralized and totally rigid — but your business environment and the challenges and threats and so forth are constantly changing — then your SOA failure will ultimately become a business failure, a failure to adapt.

The world is about finding that midpoint, where control and governance is centralized enough to keep things safe and secure, and to be able to take advantage of business opportunities — where consolidation makes sense — while at the same time staying agile.

Federated Approaches Balance Governance and Control
If you look at it from a metaphorical perspective, for example, the federal government of the United States is a very interesting model. You have essentially a bunch of business units called states, that each have their own legislation, their own competency centers called state legislatures, and even their own executives called governors.

Those look a lot like business units to me. If you look at the notion of federation and the federal government model, what you see is this whole principle of jurisdiction. Ultimately, competency centers become the legislative bodies within these organizations. All of the efforts that I've seen to codify methodologies around SOA tend to focus on these competency centers or centers of excellence, primarily because there needs to be an inclusive organization for adjudication and jurisdiction, as opposed to having a model, where it's just a single iron-clad dictator that controls all policy.

You can actually look at the failure modes of failed states. If you look, for example, at how you establish and foment democracy, there are some models, some really good, real-world

cases about how not to establish democracy. Not to get too overly abstract, but there are a lot of practices and principles around establishing policy federation. The interest in doing so is the interest in establishing a controlled paradigm that actually serves the common good in a way that enables agility, but also enables this centralized capability of control.

But governance is an abstract concept, and you don't necessarily want to dictate one governance model that's applicable or should be applicable to all organizations and industries. Everybody has their own pressures, market pressures and so forth. In terms of SOA governance, there are radically centralized models in a given organization.

You might want to do the equivalent of a [Myers-Briggs test](#) and figure our what kind of company it actually is. Then, figure out in what way to approach governance, so that we don't try to overstep what's possible on a linear basis. I suppose it's also evolutionary. Some companies might need to start out as strict dictatorships, and then perhaps the government withers away and it becomes a democracy. We've seen the example of Eastern Europe over the last 20 years.

The U.S. Constitution, which has some key design patterns in it. If you actually look at the separation of power declared in the preamble, it says that the purpose is, "… in order to form a more perfect union." So, there's this notion of the intent of the formation of this governing entity, which is the goal of a more perfect union, which essentially means that there's a distribution of power and that the consent of the governed essentially be the overriding principle.

The idea that comes out of that, though, is the clause "provide for the common defence." That's really talking about the security domain, whether it's physical security or technical policies associated with the current data. The idea is that it actually should be a federated concern. In other words, security is everybody's business. You can't just delegate it to one unit and say, "It's your business."

Technology so permeates how a company operates, particularly if you're Internet-facing and if you're using and exploiting the Internet for more and more of your supply chain, your distribution, your transportation, for the way in which you attract sales and customers, and so on and so forth. So technology now is at an intersection with the corporation as an organization, and perhaps that's what's forcing this need for a different look at how to organize in general, and, therefore, on how to govern.

Deal with Failure Maturely and Constructively
We talk about the post-modern corporation. Where are these companies going to get their IT? Where are they going to get their technology? We're seeing more and more instances of companies going outside, not wanting to get involved with the bits and bytes of managing a technology infrastructure. We call it "software as a service," "managed hosting," and various types of acronyms and terminology.

The metaphor of nations and the competition between nations has typically been along the lines of warfare in our history. Look at the metaphor of business at war, which is essentially competition for the survival of the integrity of your company against all others. It's not on the battlefield, but it's for customer value, for creating services that people treasure. In the history of warfare between governments and nations, what we found is that the organizations that leverage technology to their advantage are the ones that come out ahead.

Abdicating the responsibilities of the management of technology to a commoditized provider creates an extreme vulnerability because your competitive differentiation should not be held or embodied by some generic provider. … Control over how your organization behaves and controls your assets and resources strikes me as something that you would never want to commoditize.

Clearly we've defined here that a successful SOA is a lot about politics, power, and moving beyond traditional norms of organization. How you do that probably is going to involve failures. If the Unites States is a good model, it had to fail a couple of times. It failed with the Articles of Confederation. It failed in dealing with slavery up until the Civil War, and perhaps for a hundred years afterward in terms of how it was dealt with in practice, if not in law.

Perhaps we should look to failures as a necessary set of learning activities, in that SOA is not going to just happen and spring up like a fungus or a mushroom after a spring rain, but it's going to have to be something that's hard-earned.

The way I want to respond to is that having maturity in the way that you deal with failure is essential. If you look at the way that our policy system functions within the United States, what you have is you have a set of policy assertions about what it is people can and can't do. But then, you actually have a policy enforcement mechanism that's heterogeneous and distributed. You have the FBI, the CIA, the state and local law enforcement, the Army and the National Guard.

You have all these different policy enforcement points everywhere, manifesting these policies. I think that having a learning engine that monitors, adapts, and revises policies, and having a competency center, an adjudication point that's deliberately there for the purpose of making those adaptations — that is an essential function.

[David Tyler](#): SOA - Is It Right For You? - Part I
September 20, 2005

This is part 1 of a three part series. The intention of this blog is to guide you through SOA and the many areas that SOA covers. This blog is not the final answer, but a set of questions, thoughts, examples, and guides to help you find the true value that SOA can create for you as well as help you determine if SOA is right for you. Managers and decision makers often ask me, "Is SOA right for us? Would a project like that ever be successful here?" Well, that is actually a hard question to answer, and usually cannot be answered honestly without some investigation. As I have often stated to my clients, it depends on what you are trying to achieve. Because SOA is an approach, technique, set of principles, it often seen as intangible and therefore not real. This also tends to allow for "interpretations" of what SOA is and how it works, leading to "SOA in a box" type solutions. However, no matter what you are trying to achieve, there are some questions you can ask yourself, and your company, that should help you in making this determination. In addition, as you go down this road, you might discover your true goals for looking at SOA and hopefully a solution on what to do.

Why SOA?
The first question I have for you is why SOA? How did you hear about it? Why did you pick it? Did you choose it because it sounded cool or techie? Did you feel that it could solve a problem for you? How did you learn about SOA? Of all the questions I just asked, the last one is the most important. How you and your company learn about SOA will probably

determine your first implementation of it. Companies that tend to learn about SOA from an EDI/EAI company, tend to focus on data integration projects that connect all of their systems to a new "Enterprise Service Bus (ESB)" (whatever that is). Companies that learn about SOA from reporting software type companies tend to focus on dashboards and scorecards as their first endeavour into SOA. The amazing thing is that all of those ARE SOA projects, at least in principle. So depending on how your company learned about SOA is probably what they will expect as a first project. This is critical because if your company expects dashboards and scorecards and you deliver an integration project, SOA will fail, because your company does not see the connection between a dashboard and an ESB.

When you look at SOA projects, what or how to implement them, you should be asking yourself what is the business group expecting from this, and what are they gaining from this. There has to be some business value generated and there has to be some business gain or else you are simply implementing an IT project and any future SOA projects will probably not get off the ground. I can never stress enough with my clients or friends about reading up on SOA. Make sure to educate your business groups. I had one client whose business community was supposedly educated on SOA. However when it came time to pitch in dollars and resources into the project, they refused because "SOA is just an implementation of web services", from their point of view, and therefore no need for the business community to get involved. Educating the business group will also help educate them on what they can expect and gain from SOA projects. This will help you in the long run by allowing you to take the "baby steps" you are going to need to take to truly implement SOA. This will also help you from the standpoint that the business community can help you define the business value, versus you trying to figure it out for yourself. You cannot implement SOA overnight; it will take you several projects to get a solid SOA foundation implemented. In addition, post implementation, you are going to make changes, both in business and technology, and those changes need to roll into your SOA implementation. We will cover how you do that in a later part of this blog.

Therefore, understanding why you choose SOA, and how you became educated in SOA, can help you determine when your company is ready for SOA. Without a proper education, you and your company will be ill prepared to support a true SOA endeavour. Moreover, you and/or your company will have the wrong expectation of what SOA can deliver.

Why not SOA?
One of the other questions you should ask yourself and your company is, "what if we do not do SOA?" This question is very important because it underlines what you or your company perceives as benefits from SOA. If the only reason to do the project is to integrate technologies, or to create a dashboard, you can implement those things and still not implement SOA. Your answer to this question should have technical ramifications as well as business value attached to it, like "will not reduce costs of operations by 10%" or "will not improve customer satisfaction by 20%". Applying these types of metrics, business metrics, allows you to keep the end goal in mind as you delve into the technical aspects of the implementation. Also with that, the business value is usually greater than the project itself. In other words, you maybe starting an integration project to service enable applications, but your business value is visibility into customer purchasing habits. This signifies both the real value that SOA brings as well as the fact that achieving SOA is not immediate. Instead, you do "baby steps", implement an ESB, then create a Composite Application, and then assign KPI metrics, and so on; each step being a project on to itself, and with the idea to deliver a specific business value that gets you to your eventual goal.

As you look at the projects that you classify as SOA, they should be small, focused, and above all deliver business value at the close of each project. If you are deciding to implement SOA, make sure to understand what the impact is if you do not implement SOA.

What did you leverage?
One of the biggest questions clients ask me is "buy or build". My response, "what did you leverage?" One of the major principles of SOA is leveraging, if you are not leveraging your current infrastructure, processes, or organizations, then you are not reaping the full value of SOA. Taking what you and your company has already invested in and expanding its capabilities, enhancing your visibility, or connecting silo'd business functions into an heterogeneous Business Service is the power of SOA. This is also critical when it comes to the education of yourself and your company in SOA; making sure that everyone, especially the business, understands that leveraging what you have today keeps the costs down and shortens the time in gaining value. One of the things that clients sometimes miss is leveraging processes and organizations. All companies have processes, look to see where those processes are the most developed, and leverage them. Leverage how they were produced or the procedures themselves, this helps you when looking at creating Business Services (which we will cover in this blog later), and creating the ability to adapt your infrastructure to your business. Leveraging your organization is also critical in creating a solid SOA enterprise. Companies tend to have the organizational components for SOA already in place, but used for a different capacity. One of the other major principles of SOA is governance, making sure that all parties contribute and monitor, both technically and from the business, on aligning the business and technology goals for the delivery of business value. You will need structuring organizations like a Program Management Office (PMO), standardization committees, compliancy departments, or change management organizations. You can use these types of organizations, no matter what their current charter is, to govern your SOA implementations and maintain a high value proposition for your company.

When you can answer these three questions, "Why SOA?", "Why not SOA?", and "What did you leverage?" you will have a solid footing on why you are looking at SOA and a beginning on whether or not SOA is right for you. You are now ready to look into something else, your infrastructure. In part 2, we will cover infrastructure and IT services, and how they play a role in determining if SOA is right for you.

Dan Creswell: SOA definitely Doomed
March 31, 2006

I made a prediction a while back that SOA wouldn't make it and that certainly seems to be the case but not for the kinds of technical reasons I outlined. In fact, there's a lesson to learn in that the real reasons for failure are almost entirely human.

We Can't Agree On What SOA Is
Is it a design discipline? Is it a technical approach? Is it business process re-engineering? Do we deal in services at all or is it a justification for document processing?

Chances are that a small group of people will have coined this term and had a specific meaning for it. We as an industry have taken and abused that term to the point where it has no useful meaning at all. We have lost information and lost what might have been a useful thought, that's bad, bad, bad. Worst of all, we repeat this cycle of term abuse over and over with all the associated costs.

We Can't Agree How To Implement It
It's XML-based messaging. It's SOAP. It's REST. It's JMS. It's an enterprise service bus (does anyone really think you can send all messages across the entire enterprise through one centralized dispatching/routing thing? Imagine trying to manage the flow of traffic across the US from one centralized point of control - scary huh?) Do it with an application server. You need an uber application server. Do it with a database. Do it with 3-tiers, do it with n-tiers. You can do it with that out-of-date CORBA thing that failed (yeah failed, just take a look around the telco's, the military and the banks).

Everyone Has a View
And worse than that, they all think they're right! If SOA really is the fundamental shift all these dignitaries say it is, it's surely too big for any of them to understand it all and thus believing that they know enough to be right is simply arrogant.

It's Way Too Complicated
Everybody and his dog has a stake in SOA - business people, analysts, architects, vendors. It's bad enough managing all of this for a single internal project in one business unit let alone on a cross-company scale.

If it's going to happen at all, ever, it's going to have to be done a little bit at a time in a loosely coupled (oh, the irony) fashion with as few dependencies as possible between groups of people, systems etc. You simply aren't going to change your accounting, processes, architecture and teams in one big bang overnight. And please don't tell me this is about agile, it's about little steps, that's it, no need for flash buzzwords.

Focus on Fundamentals
And in a galaxy far, far away but close to Web 2.0, there are some businesses that are focused on good customer interaction, delivering focused minimalist solutions and generating value whilst avoiding overly weighty processes and running screaming from obsessive ever-decreasingly effective cost cutting (which ironically seems to add costs elsewhere).

Shamus McGillicuddy: SOA Projects fail to deliver as expected
September 05, 2007

Most service-oriented architecture (SOA) projects fail to deliver ROI, according to a new report from Nucleus Research Inc.

Nucleus Research surveyed 106 organizations and found that only 37% of SOA projects demonstrated ROI.

"People do succeed some of the time in getting benefits from SOA in the form of improved developer productivity," said David O'Connell, senior analyst at the Wellesley, Mass.-based research firm. "What we found, though, was that adoption tended to be somewhat narrow. People tend to get into SOA in an ad hoc or departmental basis."

O'Connell said adoption of SOA tends to get "stranded." He said a major value of SOA is the concept of reuse. In SOA, developers create software that performs general functions or "services" that can work in different business contexts. These services can be strung together in an architecture to perform business processes. A major advantage of such architecture is

that many of the software services can be reused, saving time and resources and speeding deployment in application development.

O'Connell said organizations often fail to delve deeply enough with their SOA adoption. They will see some benefits in the form of improved productivity for software developers and testers, but adoption of this reusable software isn't broad enough. For instance, developer productivity increased by an average of 28% in the Nucleus Research survey.

But companies don't go deep enough with the technology. O'Connell found that SOA touches only 27% of current IT projects in the average organization. And only 32% of published software services get reused.

Jeff Kristick, vice president of marketing at Tibco Software Inc., a Palo Alto, Calif.-based vendor of SOA software and consulting services, said it was possible that many of the organizations surveyed by Nucleus are on their way towards an ROI. They just haven't reached it yet.

"It would be interesting to overlay the maturity of projects in this survey," Kristick said. "I think the companies that have positive returns have higher numbers of projects."

Kristick pointed out that companies often don't see a return on their first couple of SOA projects. It's only after several projects are in place and companies are reusing software code that it truly starts to see positive results.

O'Connell said Kristick's point is valid, but he added that something else is at work here.

"You'll be lucky to get ROI on the first couple projects," he said. "You have got to do a few projects to get a positive return. However, SOA has been around for a while, and I think the lack of ROI has more to do with a lack of breadth rather than a lack of projects."

Organizations fail to get that breadth of adoption because they fail to broadcast the advantages of SOA reuse internally. O'Connell said too many organizations fail to have an internal champion to push the reuse of software services.

"It's something we emphasize," Kristick said. "How to promote reuse and to encourage developers to think outside the project they're working on and the service they're creating. We talk to our customers a lot about that and how people underestimate that."

Many companies invest in registries and repositories where developers can find software services that can be reused. But these applications aren't enough to guarantee reuse. Human resources departments should enact policies that require developers to adopt it.

But O'Connell said it's most important that these companies have an internal SOA champion.

At Con-Way Inc., a $4.2 billion freight and logistics company based in San Mateo, Calif., that champion is lead enterprise architect Maja Tibbling.

"It's absolutely necessary that senior IT executive management be on board," Tibbling said. "If it isn't approached from an architectural perspective it won't be successful. Just doing random and spotty hit-and-miss SOA doesn't give any value."

She said a company needs an evangelist for SOA.

"People can talk about governance all they want," Tibbling said. "They can have a wonderful repository sitting there, but if no one knows to look there they're not going to find it. You have to get the word out in touting the benefits of SOA and making it known what there is to reuse.

"The key is that services need to be built in such a way that they can get leveraged in different contexts, and that you provide an agile platform for business processes," she said.

Tibbling, who has been working with SOA since 1995, said she's in the process of extending the technology from Con-Way's freight division to the entire enterprise. She said the biggest barrier to adoption she hears about within her company is understanding reuse.

"I hear people say, 'We don't have time to consider other uses [of services],'" Tibbling said. The key is to build software services that are general enough that reuse is easier to achieve, but not so general as to make them useless.

[Charles Badcock](#) [InformationWeek](#): The SOA Gamble: One in three companies are disappointed, our survey finds
September 8, 2007

Increased complexity and high costs is a culprit. BT offers lesson in doing service oriented architecture right.

Revamping the it infrastructure for [service-oriented architecture](#) is well under way at many companies, but the mostly positive results are dogged by uncertainty, including an unsettlingly high number of projects falling short of expectations, an *InformationWeek* survey shows.

Ten percent of respondents say SOA/Web services have exceeded their companies' expectations, and 58% say they've met expectations, in our survey of 278 professionals. But another 32% say the technology has fallen short of expectations.

The SOA Gamble: [More Poll Results in the IMAGE GALLERY](#)

Asked if SOA/Web services have achieved the business goals used as the rational for adopting the technology, 69% say they're meeting "some" of their goals, while 15% say they aren't.

The up-and-down results are striking when one considers that SOA isn't all that new. It first gained credence as a design principle and dominant architecture around the beginning of the decade, as IBM and others championed it as a new way to design business technology. It's had ample time to wend its way into the thinking and planning of IT staffs, and still the outcomes show a persistent thread of disappointment.

In part, it's because the goals for SOA are so high. No longer is SOA confined to its early goals of application integration and service reuse. Now it's about an expanding litany of business goals, such as software reuse and business alignment, that companies have historically struggled to achieve.

Business Case For SOA

- Standardization — 63%
- Operational savings — 62%
- Business process automation — 56%
- Improved application performance — 43%
- Increased customer satisfaction — 38%
- Better integration with the operations of our business partners — 33%
- Our biggest customers demanded it — 29%
- Increased sales — 19%
- Better visibility on how goods move through supply chain — 15%
- Other — 6%

Data: InformationWeek Research SOA/Web services survey of 278 business technology professionals; 229 companies using SOA/Web services

The big goals include flexibility in application development, by 77% of survey respondents; increased software modularity (70%); lower costs (59%); better integration with business partners (53%); and bringing new products to market faster (40%).

So why did SOA disappoint almost a third of companies? Complexity's the biggest culprit: 58% say SOA introduced more complexity into their IT systems rather than resolving it; 30% say SOA cost more than they expected.

Those findings have shown up in recent research by Sajay Sethunath, chief architect for BearingPoint's Wall Street consulting business. For reuse to pay off, the service must be designed for reuse by a potentially large variety of systems, he says. That's because designing and implementing for reuse costs more--35% more at the Wall Street firms that BearingPoint advised on SOA, compared with a straightforward, single hardware platform design.

Of those falling short of expectations, 27% say SOA failed to provide the expected level of integration.
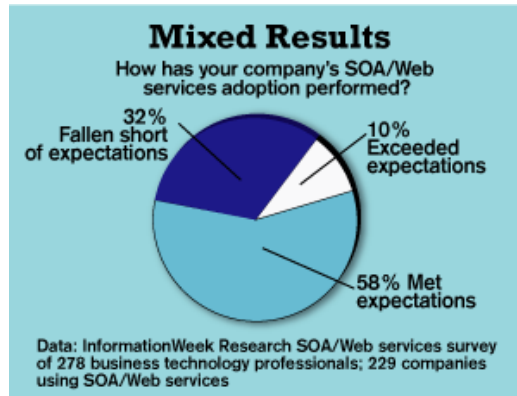
There's great disparity among companies in terms of how widely they're putting SOA to use. Based on results from 400 companies, the Aberdeen Group concludes that companies building out a full SOA middleware infrastructure are getting much better results than those just deploying Web-services-based applications--or what analyst Perry Donham calls "SOA Lite." Companies focusing on SOA infrastructure have "lower application life cycle costs, better throughput for projects, and higher levels of user satisfaction," Donham says in a recent report.

Lesson from BT
George Glass, chief architect at BT, the U.K. telecommunications company, agrees with the approach of building SOA that sits atop a fully functioning set of middleware. BT builds services around Web standards, including WSDL, UDDI, and SOAP. But those services are closely linked to the company's BEA WebLogic middleware, its IBM WebSphere MQ messaging service, and other elements of its conventional infrastructure.

Some services are new, and some are produced by "wrapping" sections of code from BT's existing 3,500 applications to capture an individual service, Glass says.

Mixed Results

How has your company's SOA/Web services adoption performed?

32% Fallen short of expectations

10% Exceeded expectations

58% Met expectations

Data: InformationWeek Research SOA/Web services survey of 278 business technology professionals; 229 companies using SOA/Web services

But he has a warning: To get cost savings, mining Web services from old applications only pays off if IT follows through and shuts down the legacy applications. Three years into developing a services architecture, BT shut down 205 systems its first year, 710 its second year, and 260 in the first quarter of its third year.

Failure to take Glass' advice could explain some of the results: Adding complexity and failing to get cost savings could result from companies adding new Web services without ridding themselves of old applications. Thirty-six percent of respondents didn't replace older technology with SOA.

Another problem with SOA can be alignment--making sure IT departments are building the services the business units most need.

IT staffs tend to pick the low-hanging fruit and convert the most easily transformed applications into services, says BearingPoint's Sethunath. Then they have no metrics to show the gain from their effort, and business-unit managers may see minimal improvement. That robs the technologists of a strong argument to expand SOA's use.

SOA has won many converts, but its popularity should not mask the challenges of delivery
March 22, 2007

The development of a service-oriented architecture (SOA) has become one of the key pillars of the IT strategy in companies across the globe. IT leaders, analysts and vendors all agree that SOA has the potential to reinvigorate IT's corporate reputation, as it ushers in a new era of agile, responsive, business-driven IT. Amid the fanfare, it can be easy to overlook that such a revolution will be no cakewalk; progress will be hard won.

The claims being made for SOA are bold, in some cases even overblown. One SOA expert and author, JP Morgenthal, goes as far as saying that SOA has ramifications far beyond the world of technology. "It is a design pattern that can be applied to any type of system in the world, including purely human-based systems, such as order taking at McDonald's."

But for most organisations, SOA has a more mundane, yet still profound purpose: to ensure that IT becomes the engine room of business agility, rather than the anchor it is often characterised as today. And there are now sufficient numbers of CIOs energised by SOA that its momentum looks unstoppable. *Information Age*'s own Effective IT research indicates that more than 50% of enterprises have adopted web services strategies.

The reasons for SOA's popularity are well documented. Traditional enterprise application integration techniques are unwieldy, difficult to code and absorb significant cost without adding much in the way of competitive advantage. SOA offers new hope: the ability to quickly compile new applications from component services should make for an IT operation that is reactive to fluctuating demand, making the enterprise more agile.

SOA is also predicated on open standards, which should make integrating applications simpler and faster. And the ability to reuse code in establishing new services, and to reuse services in multiple applications, should reduce the amount of development work that is required in any project, and therefore reduce costs.

Hidden dangers
The allure of SOA is palpable and, according to IT advisory group Forrester Research, adoption is gathering pace. As of late 2006, 20% of European organisations have a fully formed SOA strategy, Forrester reports, up 6% from 2005. A further 20% are applying SOA technologies selectively (again an increase on the 2005 figures – up 4%). The key drivers, Forrester notes, are increased business and application agility, followed by lower software development time and reduced costs.

But not all companies that have begun to service orient their IT infrastructures have found the switch to be as edifying as expected. One in five US companies that have taken the SOA plunge has found that the move introduced unexpected complexity, reports IT analyst group Ovum.

Annraí O'Toole, Cape Clear: Unless you really understand what services you are going to create, and why, there is no point in doing SOA

Back in 2005, now defunct analyst organisation SOA Pipeline conducted a survey of SOA satisfaction. Nearly a quarter of respondents (24%) said that their SOA project had run into 'serious difficulties' while 14% described their SOA efforts as having failed. In the time since that survey was conducted, expertise and experience have of course been accumulated, and technological standards have improved. Nevertheless, these findings are indicative of an uneasy transition to SOA.

One of the chief criticisms levelled against SOA is that the anticipated return on investment (ROI) will not materialise for many years. Researchers at market watcher Saugatek Technology believe that SOA will not substantially reduce IT costs for those organisations already adopting it until after 2012. The desired payback will be stalled by the investment in training that is necessary to deliver SOA and the lack of genuinely open standards that will allow competitive products to interoperate. Those expecting a quick return, the report's authors conclude, will be sorely disappointed.

None of this calls into question the wisdom of SOA. Indeed, given its status as a 'design principle' rather than a technological product set, it is almost impossible to question its wisdom: a failed SOA implementation is a failure in the application of the principles, not of the principles themselves.

But it certainly highlights that, despite the mounting excitement around the model, there is a real danger that SOA projects will make IT more complicated and more expensive, not simpler and cheaper.

So what is it that stands in the way of businesses in their attempts to realise the benefits of SOA? Examined here are a number of pitfalls that can either undermine the chances of gaining a return on investment from SOA, or even result in systems that simply fail to work. They can be roughly grouped into technological concerns and organizational concerns, as the task of becoming a service-oriented enterprise demands both new technological skills and a new approach to technology.

Design flaws
The ideal end-point for an SOA implementation may be regarded as a collection of discrete web services, well-documented, engineered to be reliable and scalable, and easily linked to business functions. However, each of these aims present its own hurdles to be overcome.

Creating this collection of autonomous web services throws up the first problem. Ideally, by the end of the SOA transformation, business architects, with a deep understanding of operational processes, will be able to rapidly compile a number of these services to automate a discrete process. IT role in this is to make sure the services work and can be integrated satisfactorily.

To date, however, the task of creating these services has been almost too easy, says Steve Craggs, president of Saint Consulting, an SOA services company. "I have seen a lot of what I call the 'right click syndrome'. This is where developers use tools to turn every single piece of application functionality into services. It is often a complete disaster."

A simple function – say that of retrieving a customer's address – might depend on a collection of legacy systems and a number of operations. For example, the accessing of the customer name from a database, the conversion of that name into a unique identifier, and then a third function that looks up the customer number and returns address details. The order in which these are conducted is vital to the successful completion of the task.

"In the 'right click syndrome', each one of these functions is turned into its own service," explains Craggs. "That means that any application which needs to retrieve a customer address must be programmed to understand the order in which these services are called. And if you are to change the way the service works, the calling application will again need to be updated."

Five common SOA mistakes
- Building unnecessary services
- Hard-coding finely-grained service level agreements into services
- Building synchronous, request-response type services
- Inadequate performance and scaling testing prior to launch
- Insufficient reuse of code

Such examples demonstrate how the inappropriate creation of services can make more integration and coding work necessary, instead of less. What should be created instead is a discrete service that conducts all three functions, so that the service reduces complexity rather than adding to it, says Annraí O'Toole, CEO of enterprise service bus vendor Cape Clear. "You should be able to bring a business person in and explain to them what the service does," he says, otherwise it is an unnecessary service.

Unless CIOs are fearless in evaluating what services are absolutely necessary there is a danger that the new architecture will simply repeat mistakes made before, he adds. "You have got to remember that that the incumbent vendors have a lot to protect; they want you to

keep using their stuff," he says. "But unless you really understand what services you are going to create, and why, there is no point in doing SOA. You must have a revolutionary approach."

The way in which services interact with one another must also be addressed. Developers grounded in Java coding may be tempted to build synchronous services, which, having issued a call, wait for responses before continuing. "When you are building large-scale applications, such as an SOA, if you build them in a tightly-coupled, request-response fashion then the result will be very brittle and the whole thing will inevitably break down," says O'Toole.

It may also be tempting to include service level agreement provisions into services. For example, a company with a subset of customers to whom it promises a higher availability of services, may prioritise traffic associated with those accounts at peak times.

A common error, says O'Toole, is to design this traffic control into services. This means that any time the service level agreements are changed, all services that relate to the SLAs must be recoded. "If you build SLAs into services, you are coding in years of misery," he says. Instead, an over-arching SLA governing service that can be applied to all other services is the appropriate solution.

Pitfalls
The implementation of SOA also demands careful consideration of the infrastructure that supports it, warns Ravi Kalakota, VP for strategy and solutions management at system integrator Unisys. "We have seen numerous examples where the software architects move the SOA into production, only to find that it has worse performance than the old mainframe."

Companies instigating SOA often discount the significance of the underlying infrastructure, says Kalakota. "If your infrastructure is fragmented, so for example a service is based on 25 different servers, consolidation is the first step. That message is not getting through."

"If you don't fix the infrastructure, but you have a great SOA, you will have to over-compensate with more servers. And that kills your ROI," he explains.

Virtualisation technology may help businesses to consolidate their server infrastructure, but it also important to keep a close eye on how hardware resources are being consumed by services, says Kalakota. The field of business service management is a complicated area, he says, and achieving a comprehensive map of how services relate to infrastructure is a massive undertaking that might not always deliver a satisfactory ROI. At the very least, though, "you have to have a feedback mechanism constantly monitoring what is going in and coming out of the service bus."

Ravi Kalakota, Unisys: We've seen the software architects move the SOA into production, only to find that it has worse performance than the old mainframe

The impact of SOA design upon infrastructure performance is often untested until the systems go into production, which Cape Clear's O'Toole reports is the most common cause project failure. "Failed SOA projects all share the characteristic that scaling and performance testing have been left to the end."

Mike Scott, the former head of innovation at BT, now at TCS Global Services, experienced the dangers of poor scalability testing first hand while at the telco. "We had an ERP system, which we wrapped as a service so we could push it out to suppliers at one end, and BT at the other. When we went into production the whole system locked, costing us two weeks fixing it."

"We argue that testing should begin in the first week," says O'Toole. "Most problems can be solved with some simple architecture redesign, but it has to be done early."

Scott also attests to potential complexity of keeping track of the link between services and hardware: "The configuration management [the process that links hardware assets to application functionality] must really be under control; but you only need two or three configuration products. At BT, we had everything going."

Service culture
Finally, SOA dictates a fundamental mind-shift for IT practitioners and business leaders alike. Developers, says Saint Consulting's Craggs, must be actively encouraged to engage in the reuse of code if any ROI is to be realised. "Vendors will tell you that agility is the return for SOA, but most companies have justified their investments in SOA by demonstrating that it will reduce development costs and the time to market for new services. Both of these depend on reuse," he says.

"But most developers worth their salt are loath to reuse code; by their nature they want to find new solutions to old problems. So you need cultural training to make them do it."

At US financial services giant Wachovia Bank, this reuse has been encouraged through the use of bonuses – the more code developers reuse, the more they get paid.

Reuse is also endangered by the differing requirements of various departments within the organisation, says Craggs.

"Often a developer will publish a service, and then a certain department will say it doesn't do what they need it to, so the developer will create a new one," he explains. "So instead of reuse, you get a proliferation of services. You need cross-departmental agreement when defining services, or there is little change of ROI."

Deciding how service requirements are to be defined in the face of differing departmental requirements – and establishing which departments are responsible for which services – calls for strong leadership from the top.

"You need to decide where the locus of control is going to reside," says Bruce Graham, senior VP at middleware vendor BEA's global SOA practice. "That decision impacts the way services are going to developed, so you need to take it early on. And because that involves a lot of company politics, it can take months to reach agreement."

Graham says that the design of an SOA calls for the CIO to become unusually engaged in the nitty gritty of technical design and departmental ownership. "In the most successful projects I've seen, the CIO has been directly involved in governance. That's a significant shift; they need to step down and get close to the process."

But others believe that CIO buy-in alone is not enough; SOA success requires the business to be involved from the outset. And therein lies an irony: For many organisations, in terms of SOA, the carrot on the end of the stick is the closer alignment of IT and business. But if any SOA project is to be successful, that alignment must already have been achieved before any work can be successfully undertaken.

Bibiography

[ACKO1967] Ackoff, R. L., *Management Misinformation Systems*, Management Science, Vol. 4, No. 4, 1967

[ACKO2003] Ackoff, R. L., *Iconoclastic management authority, advocates a "systemic" approach to innovation*, MCB UP Limited, Vol. 31, No. 3, 2003

[ARSA2005] Arsanjani, A., *How to Identify, Specify and Realize Services for your SOA (Part I)*, IBM, 2005

[AFSH2007] Afshar, M., *SOA Governance: Framework and Best Practices,* An Oracle White Paper, Oracle, May 2007

[ALBE2005] Alberts, D. S. and Hayes R. E., *Power to the Edge*, Command Control in the Information Age, CCRP, 2005

[ANSI1975] ANSI/X3/SPARC Study Group on Data Base Management Systems, *Interim Report. FDT*, ACM SIGMOD bulletin. Volume 7, No. 2, 1975

[ASHB1956] Ashby, R.W., *An Introduction to Cybernetics*, Chapman and Hall, London, 1956

[ATKI2005] Atkinson, S. R. and Moffat J., *The Agile Organization*, From informal networks to complex effects and agility, CCRP, July 2005

[BAKE2006] Baker, D. and Curran C., *The Right Way Towards SOA*, Diamond, Management and Technology Consultants, 2006

[BART2005] Barton, T., *Capgemini to Contribute First SOA Notation and Methodology to OASIS SOA Adoption Blueprints Committee,* tom.barton@capgemini.com

[BLOO2006] Bloomberg, J. and Schmelzer R., *Service Orient or Be Doomed!,* How Service Orientation Will Change Your Business, Wiley, March 2006

[BROW2006] Brown, W. A., Moore G. and Tegan W., *SOA Governance-IBM's approach*, Effective governance through the IBM SOA Governance Management Method approach, White paper, IBM, August 2006

[CHEC1981] Checkland, P., *Systems Thinking, Systems Practice*, John Wiley and Sons, 1981

[CHEC1985] Checkland, P., *From Optimizing to Learning: A Development of Systems Thinking for the 1990s*, Journal of Operational Research Soc., Vol. 36, No. 9: 757-767, 1985

[CHUR1971] Curchman, C. W., *The design of Inquiring Systems: Basic concepts of Systems and organization,* Basic Books, 1971

[CHUR1994] Curchman, C. W., *Management Science: Science of Managing and Managing of Science Interface* Vol. 24 1994

[DAFT1978] Daft, R. L. and Macintosh N. B, *A New Approach to Design and Use of Management Information*, California Management Review, Vol. 21, No. 1, 1978

[DAFT1986] Daft R. and Lengel R., *Organizational Information Requirements Media Richness and Structural Design*. Management Science, 1986

[DAVE1992] Davenport, T. H., Eccles R. G. and Prusak L., *Information Politics*, SLOAN Management Review, Fall 1992

[DAVE1997] Davenport, T., *Information Ecology,* Oxford University Press, 1997

[DAVE1998] Davenport, T., *Putting the Enterprise into the Enterprise System,* Harvard Business Review, 1998

[DRUC1988] Drucker, P. F., *The Coming of the New Organization*, Harvard Business Review, January-February: 45-53.

[DRUC1994] Drucker, P. F., *The Theory of the Business*, Harvard Business Review, September-October 1994

[ERLT2005] Erl, T., *Service Oriented Architecture, Concepts, Technology and Design*, Prentice Hall Professional Technical Reference, 2005

[EVAN1997] Evans, P. and Wurster T. *Strategy and the New Economics of Information,* Harvard Business Review, September- October 1997

[FEA 2007] *FEA Consolidated Reference Model Document,* Ver. 2.3, www.egov.gov, 2007

[FORR1961] Forrester, J., *Industrial Dynamics*, 1961

[GALB1977] Galbraith, J. R., *Organizational Design*, Addison-Wesley, 1977

[HEDB1980] Hedberg, B., Using Computerized Information Systems to Design Better Organization and Jobs. In. Bjørn-Andersen, red. The Human Side of Information Processing, North-Holland, 1980

[HEDB1978] Hedberg, B. and Jönsson S., *Designing Semi-Confusing Information Systems for Organizations in Changing Environments,* Accounting, Organizations and Society, Vol. 3, No. 1: 47-64, 1978

[HEND1990] Henderson, R. and Clark K. *Architectural Innovation: The reconfinguration of Existing Product Technologies and Established Firms*. Administrative Science Quarterly, Vol. 35, 1990

[HEWI1986] Hewitt, C. (1986). Offices are Open Systems. *ACM Transactions on Office Information Systems,* Vol. 4**,** No. 3, 1986

[HOLL2006] Holley, K., Palistrant J. and Graham S, *Effective SOA Governance,* IBM Corporation , 2006

[HUGO1991] Hugoson, M. Å., *VBS Verksamhetsbaserade System, Principer och tillämpningar*, Göteborg, Programator AB, 1991

[IBMa2006] IBM Corporation, *SOA Governance*, Speed-start Web services, 2004 – SOAs and Web services, January-April 2006.

[IBMb2006] IBM Global Service, *Extend the value of your core business systems. Transforming legacy applications into an SOA framework,* IBM Corporation, 2006

[IBMa2007] IBM Corporation, *SOA Governance and Service Lifecycle Management,* 2007

[IBMb2007] IBM Corporation, *Introduction to SOA Governance,* 2007

[IBMC2006] IBM Corporation, *SOA Governance, Service Oriented Architecture*, 2006

[IEEE1471] ANSI/IEEE Std 1471, *Recommended Practice for Architectural Description of Software-intensive Systems*, http://www.iso-architecture.org/ieee-1471/

[IFEA2008] Institute For Enterprise Architecture Developments, (IFEAD), EA and Services Oriented Enterprise (SOE)/Service Oriented Architecture (SOA) and Service Oriented Computing (SOC), 2008

[KOON1988] Koontz, H. and Weihrich H., *Management*, McGraw-Hill Book Co., 1988

[LANG1978] Langefors, B., *Analysis of User Needs*. In G. Bracchi och P. C. Lockermann, red, *Information Systems Methodology*, Berlin, Springer-Verlag, 1978

[LANG1984] Langefors, B., *The Infological Equation and Decentralization*. SPOT-3, Göteborg, SYSLAB, Chalmers tekniska högskola, Göteborg, 1984

[LANG1986] Langefors, B., Information and Management Systems. *Erhvervs Økonomisk Tidsskrift*. Særtryk. Udgivet af Foreningen af Danske Civiløkonomer, 1986

[LIBR2008] LIBRIS, Nationella Bibliotekssystem, www.libris.kb.se/english/indexeng.jsp

[LIND2006] Lindquist, M. G., *Libris som lokalt system,* Kungliga biblioteket/BIBSAM, Dur 68-211-2005, February 2006

[McLE2006] McLean, V. A., *SOA: Reference Model, Standards and Specifications,* Booz, Allen, Hamilton Consulting, 2006

[MAGO1998] Magoulas, T. and Kalevi P., *Strategisk IT management*. Ph. Doctors Dissertation, Göteborgs Universitet, 1998

[MALO1987] Malone, T. W. et al., *Electronic Markets and Electronic Hierarchies,* Communications of the ACM, Vol. 30 No. 6, June 1987

[MARC2006] Library of Congress, *MARC Standards*, Network Development and MARC Standards Office, 2006

[MINT1989] Mintzberg, H., *Mintzberg on Management,* Free Press, New York, 1998

[MORG1914] Morgan M., *Ten Books on Architecture*, Harvard University Press, 1914.

[NADH2003] Nadhan, E. G., *Service Oriented Architecture Implementation Challenges,* EDS Solutions Consulting, 2003

[NEWC2005] Newcomer, E. and Lomow G., *Understanding SOA with Web Services*, Addison-Wesley, 2005.

[NICK2005] Nickull, D., *An Introduction to the OASIS Reference Model for Service Oriented Architecture (SOA),* Adobe Systems, 2005

[OASI2006] OASIS, *Reference Model for Service Oriented Architecture*, www.oasis-open.org, 2006

[ONEL2008] *OneLook Dictionary*, http://www.onelook.com/

[OPEN2007] *SOA Governance: An Open Group SOA project*, The Open Group, 2007

[RAYD2007] Ray Daniel, *Advancing the Business/IT linkage with SOA Governance and Service Lifecycle Management*. WebSphere Americas Executive, rmdaniel@us.ibm.com, IBM Corporation, 2007

[SCHM2007] Schmelzer, R., *The Governance, Quality, Management Trifecta: Making Loose Coupling Work*, Document ID: ZAPFLASH-200765, www.zapthink.com, 2007

[SIMO1969] Simon, H. *The Architecture of Complexity.* In The Science of Arcticials, MIT Press, 1969

[SMIT2006] Smith, D. and Place P., *Governance Issues for a service-Oriented Architecture*, 2006

[SOAO2007] SOA Wichashita, *SOA Executive Forum,* SOA Organix, 2007

[SVÄR2006] Svärdström, J., Magoulas T., and Pessi K., *A Framework for Enterprise Morphology,* IT University Göteborg, 2006

[THOM1967] Thompson, J. D., *Organizations in Action*, McGraw-Hill, 1967

[WEIL2004] Weill, P. and Jeanne R., *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results,* Harvard Business School Press, 2004

[WIKI2008] *Wikipedia, the Free Encyclopedia,* http://en.wikipedia.org/wiki/IEEE_1471, 2008

[WHEE2005] Wheeler, B., *IT Governance*, Nokia NEBIC and IT Governance Program, 2005

[ZIMM2004] Zimmermann, O., Krogdahl P., and Gee C.: *Elements of Service-Oriented Analysis and Design,* IBM Library, 2004