University of Twente
Enschede - The Netherlands

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

A Stage Maturity Model for the adoption of an enterprise-wide Service-Oriented Architecture (SMM-SOA): a multi-case study research

Master thesis Business Information Technology

Maarten Veger, BSc
University of Twente, Enschede, the Netherlands
September 26, 2008

Supervisors

Dr. L. Ferreira Pires, University of Twente
Dr. M.E. Iacob, University of Twente
Ing. R.P. Bussink, Managing Consultant, Capgemini Netherlands
Drs. E.R. Bakker, Competence Cluster Manager, Capgemini Netherlands

Abstract

A Service-Oriented Architecture (SOA) is an architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA adoption can be a solution to integrate the heterogeneous applications in an organization. In a SOA, existing applications are decomposed into application services, and business processes are decomposed into business services. Application services can be reused throughout the organization because their service descriptions are published in a service registry, and they can interact using standards for the exchange of messages. Application services are loosely coupled, and can therefore be orchestrated to automate business processes. By using standards for message exchange, service description and service discovery, functionality residing in different business functions can be composed to automate business processes that span different business functions.

Adopting an enterprise-wide SOA involves the translation of the SOA principles into concrete architectures and implementations, which fit in with the existing applications. This translation requires that choices are made about which standards, technology and software to use to implement SOA. However, SOA adoption is not only a technical implementation. It requires that the importance of organizational aspects is acknowledged and managed by the stakeholders that are involved in the SOA adoption program. Experience in SOA adoption shows that it is a complex process, and that the stakeholders involved tend to oversimplify the expected effort that is required to adopt an enterprise-wide SOA. As a consequence, organizations are dissatisfied with their SOA adoption and some organizations even cancel their SOA initiative. Organizations should gradually adopt an enterprise-wide SOA to minimize the risk of failure and to gain management commitment.

Stage maturity models can support the SOA adoption process, by functioning as a roadmap for organizations. Stage maturity models break down the adoption processes in steps, and consider different perspectives to measure the progress of adoption. Some stage maturity models exist for enterprise-wide SOA adoption, but these models are not independent of technology, they lack attention for management involvement and organizational issues, and they lack an assessment tool from which organizations can derive an improvement plan.

We propose a Stage Maturity Model for the adoption of an enterprise-wide SOA (SMM-SOA). Our model is based on a review of literature, and validation with experts that are active in the field of IT architecture. The stage maturity model consists of six stages and six maturity aspects. The stages reflect the cumulative adoption within the organization in an s-curve, as observed earlier in research about IT adoption. The six stages are named siloed, experimental, applied, integrated, institutionalized and networked. The six maturity aspects can be divided in respectively three organizational aspects and three technical aspects: strategy & governance, organizational change, business architecture, information model, application architecture, and operational infrastructure.

Based on the six stages and six maturity aspects, a questionnaire was developed to assess the maturity of three organizations in a multiple-case study. Two approaches could be observed from the case studies: decomposing either the business architecture into business services, or the application architecture into application services and orchestrating business processes based on these application services. These approaches can be related to the business or IT strategy respectively.

Preface

This thesis is the end product of my study 'Business & IT' that I followed at the University of Twente. During my study I learned a lot about IT architectures and business processes, and the topic of Service-Oriented Architecture got my attention as it is considered to be the bridge between these two fields. I started reading about SOA, and wanted to know how it was adopted in practice. I wrote a concept research proposal and send it to Capgemini, a large IT consultancy organization in the Netherlands. Before I knew it, I could start my graduation in the sector Financial Services at Capgemini.

I started reading a lot of literature about SOA, and quickly got lost in the jungle of terms and concepts. It was a challenge to keep a narrow focus, and to extract useful information from the wealth of scientific and business information. I remember many fruitful discussions with IT architects and peer students, which were also involved in this topic. As my research progressed, I managed to sharpen my scope, and especially I made decisions about what should be out of the scope of my research. My initial goal was to validate the model that I constructed in practice, but this was not an easy ride. Many organizations hesitated to participate, because of the sensitivity of a maturity assessment, and the 'maturity verdict' that would follow as a consequence. Although many organizations did not participate in my research, I made a lot of observations from the intake interviews that I conducted with them. After a lot of phone calls, e-mails, and interviews, I achieved to convince three organizations to participate in this research, for which I am very grateful. Looking back at the whole process, I learned a lot of things about conducting research, but also about myself.

I could not have written this thesis without the help of others. Firstly, I want to thank my supervisors from University, dr. Luís Ferreira Pires and dr. Maria Iacob, and my supervisors from Capgemini, Ing. Richard Bussink and Drs. Erik Bakker, for their support, critical views, interesting discussions, and enthusiasm about the topic. Secondly, I want to thank the experts from Capgemini that were involved in this research, by participating in the group session, sharing their experiences and visions, and/or by making the time for an interview. Thirdly, I want to thanks all other experts from business and academia that participated in this research. Fourthly, I want to thank the graduate peer students at Capgemini for having interesting discussions, reviewing my work, and giving renewed motivation. Special thanks go to Anne Hiemstra, Bas Huiskens, and Sefan Linders. Fifthly, I want to thank my friends for supporting me along the process, and for having relaxing moments besides my graduation. Special thanks go to Marte Dijkstra, Sander Schoneville, Tijs van den Broek, Timo van der Walle, and off course to all those others. Last but not least, I want to thank my family for supporting me to grow mature, and for supporting me to finish my graduation.

I hope you find it enjoying and useful to read my thesis.

Utrecht, September 26, 2008

Maarten Veger

"To grow mature is to separate more distinctly, to connect more closely"

- Hugo Von Hofmannsthal (1874 – 1929)

Table of contents

1. Introduction

This chapter gives an introduction to this thesis. Section 1.1 discusses the problem area related to the adoption of an enterprise-wide Service-Oriented Architecture. Section 1.2 presents the research aim and research questions. Section 1.3 describes the research method used to answer the research questions. Section 1.4 explains the structure of this thesis.

1.1. Problem area

The integration of heterogeneous IT applications is an ongoing problem within and between organizations [1]. Caused by departments that want to maintain power and the fear to change large applications, different application silo's prolong to exist. To support the dynamic change to business processes that span multiple business functions, the functionality within the different application silo's need to be decomposed into components. These components can then be combined to support the business processes. Middleware is used to enable the interaction between application silo's and to decompose the functionality [2]. Middleware functions as a translator between heterogeneous applications. However, middleware is traditionally focused on the integration of applications within an organization and not between intra-organizational applications. In an intra-organizational setting, there is no obvious way to place the middleware as there is no central authority. Furthermore, there is a lack of standardization between different middleware platforms.

Adopting a Service-Oriented Architecture (SOA) is a solution to overcome the limitations of middleware. A SOA is an architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner [3]. SOA can be applied to application and business logic, respectively representing the IT and business perspective of business process automation [4]. When SOA is applied to application logic, applications are decomposed to application services that represent basic application functionality. Application services abstract underlying functionality and exchange information through a service interface that is described by a service description. Application services publish their service descriptions to a service registry where they can be discovered by other application services. Enabled by the Web Service standards, application services are interoperable and can be reused as flexible building blocks to develop new applications.

When SOA is applied to business logic, business processes are decomposed to business services that represent the basic business activities within a business process. Business services are described in terms of input and output and therefore are loosely coupled so they can be reused in different business processes. In an ideal SOA implementation, business processes are implemented by orchestrating application services. Orchestration specifies the order in which the different application services are invoked. An orchestrated business process is a chain of application services that can be changed easily because the services are loosely coupled and execute autonomous. New services can be added to the chain, services can be removed from the chain and the services in the chain can be reordered so that orchestrated business processes can easily be reconfigured to fulfill changing business requirements.

The ideal SOA that is presented in theory is mainly gradually adopted by organizations [2, 5-8]. Experiences in practice have shown that adoption of an enterprise-wide SOA is a long-term and complex transformation process [5-7, 9-11]. Although many organizations have adopted SOA, comprehensive SOA implementations are currently still scarce [6, 12, 13]. The complexity of

enterprise-wide SOA adoption not only involves many technical issues related to the SOA implementation but also a variety of organizational issues like funding of services [7, 14]. Caused by the complexity, the stakeholders within organizations have a limited understanding of the impact of adopting an enterprise-wide SOA [13]. Potential and current adopters tend to oversimplify the expected effort that is required for enterprise-wide SOA adoption [5]. The limited understanding of adopters is also caused by their imitation behavior: organizations adopt SOA because other organizations they see as competitive have already started adoption [15, 16]. This imitation behavior was intensified by the hype around SOA that created overinflated expectations. The limited understanding, unrealistic expectations and misconceptions are a risk for successfully adopting an enterprise-wide SOA. SOA adoption is a long-term transformation process in which the benefits do not show on the short term [6]. Without understanding the impact of SOA adoption, there is a risk of losing commitment in the organization for the SOA initiative as it does not showcase immediate success. This is reflected by the fact that one in three organizations are not satisfied with their SOA adoption initiative and that some organizations cancel their SOA initiatives [6, 17].

Stage maturity models can be used by organizations to improve the understanding and to create realistic expectations. Stage maturity models guide the transformation process needed to adopt IT technologies like SOA. Maturity models are widely used within the IT field to measure and improve quality. Process maturity models like CMMI are aimed at controlling and improving the quality of software development processes but these models lack attention to architectural processes and do not measure enterprise-wide adoption of technology [18]. Stage maturity models measure the maturity of IT adoption from an organization-holistic perspective and measure the quality of IT technology products like architectures and derived implementations. The construction of a stage maturity model involves three key elements: grow stages in the adoption of IT technologies, maturity aspects that reflect different perspectives on the progress of adoption and an assessment tool to assess the maturity of an organization [19]. By distinguishing different grow stages, a stage maturity model functions as a roadmap for organizations. By using an assessment tool organizations can assess their current maturity stage for each aspect. Based on the current stage, the organization can determine a goal stage that needs to be achieved. Using the results of a maturity assessment, an organization can develop and execute an improvement plan to reach the goal stages from the current situation. In addition to the assessment, the maturity model functions as a communication tool to improve the understanding of the stakeholders within the organization.

Stage maturity models not only function as a roadmap to support organizations, but also as a research tool to analyze adoption behavior and adoption success of different organizations over time. Cumulative adoption of technology typically follows an s-curve, where early adopters adopt the technology first followed by the majority [20]. Stage maturity models find their origin in Electronic Data Processing in the seventies [21] and have more recently been used for the implementation of Enterprise Resource Planning systems [22]. With the implementation of ERP systems the importance of organizational aspects was identified as a factor for IT adoption success [23, 24]. Enterprise-wide SOA adoption also has a major impact on technical and organizational aspects [14], so both aspects should be included in a stage maturity model for enterprise-wide SOA adoption. Stage maturity models for SOA adoption are currently already used: 16% of large organizations respond to SOA adoptions challenges by choosing a SOA maturity model as organizational roadmap [13].

Several stage maturity models exist for enterprise-wide SOA adoption, these are mainly developed by software vendors and consulting firms [25, 26]. As a consequence of the creators, these models function as a sales tool or as an instrument to identify clients' needs. SOA is a technology-agnostic architectural style and so a stage maturity tool must be independent of software products that implement SOA. Furthermore, current models lack attention for management involvement and organizational changes that are needed [27]. Enterprise-wide SOA adoption is a long term path and understanding and commitment of the management are required for successful adoption. Another important lack in current models is the availability of an assessment tool. Although some indicators are used in the different models to assess maturity, the indicators are not used homogeneous throughout the different stages. Earlier research about maturity of enterprise-wide SOA adoption is scarce. The CSOAMM combined two vendor models but lacked validating the combined model in practice [28]. The Service-Oriented Enterprise Maturity Framework includes the key elements of a stage maturity model and was validated in practice but the model is mainly focused on the decomposition of business services and lacks focus on application services within a SOA [19].

This thesis proposes the Stage Maturity Model for the adoption of an enterprise-wide SOA (SMM-SOA). The model was constructed based on interviews with experts and validated based on a multiple-case study with three organizations. SMM-SOA defines six maturity stages and six maturity aspects. The six maturity stages reflect the adoption process of SOA throughout the organization following the s-curve spread. The six maturity aspects cover technical and organizational aspects. The maturity aspects are further subdivided in a total of 31 key indicators to assess specific maturity elements within the six aspects. Based on the six maturity stages and the 31 key indicators an assessment tool was developed to assess the current stage and the goal stage of maturity on a general and specific level. The assessment tool was used to assess the maturity of enterprise-wide SOA adoption at three large organizations. The assessment was combined with semi-structured interviews with the persons leading the SOA adoption imitative in the organization. The results within the three cases were compared with each other to validate SMM-SOA.

This research is relevant for organizations planning to or already adopting SOA by offering a technology-agnostic roadmap for enterprise-wide SOA adoption. This research is also relevant for consulting firms to determine the current state of enterprise-wide SOA adoption at clients and to determine a goal state. This research is relevant for researchers by offering a framework for analyzing enterprise-wide SOA adoption behavior and strategic use of SOA by organizations.

1.2. Research questions

The main goal of this research is *to propose a stage maturity model for enterprise-wide SOA adoption and to validate this maturity model in practice*. The stage maturity model can be used by organizations to function as a roadmap, to improve understanding of SOA adoption by stakeholders within the organization and to create realistic expectations. The stage maturity model can be used by researchers to analyze SOA adoption behavior of organizations and to extract factors that influence the success and failure of SOA adoptions.

The main research question for this thesis is formulated as:

> ***How is an enterprise-wide Service-Oriented Architecture (SOA) adopted by organizations and how is SOA integrated in the IT and business strategy of organizations?***

The main research question is subdivided in the following sub questions:

> ***RQ1: Which stages can be distinguished in the adoption process of an enterprise-wide SOA?***
>
> ***RQ2: How can maturity of SOA adoption be measured and made operational?***
>
> ***RQ3: How is SOA integrated in the IT and business strategy of organizations?***

1.3. Research approach

The research approach used to answer the research questions is shown in Figure 1. The construction of the stage maturity model was based on a literature study and by interviewing experts. An assessment tool was developed to assess the maturity of three organizations within a case study. The assessment was combined with a semi-structured interview with the persons involved in the SOA adoption program. The three cases were analyzed individually and a cross-case analysis was performed to compare the three case studies. The results of the multi-case study were used to validate the stage maturity model in practice.

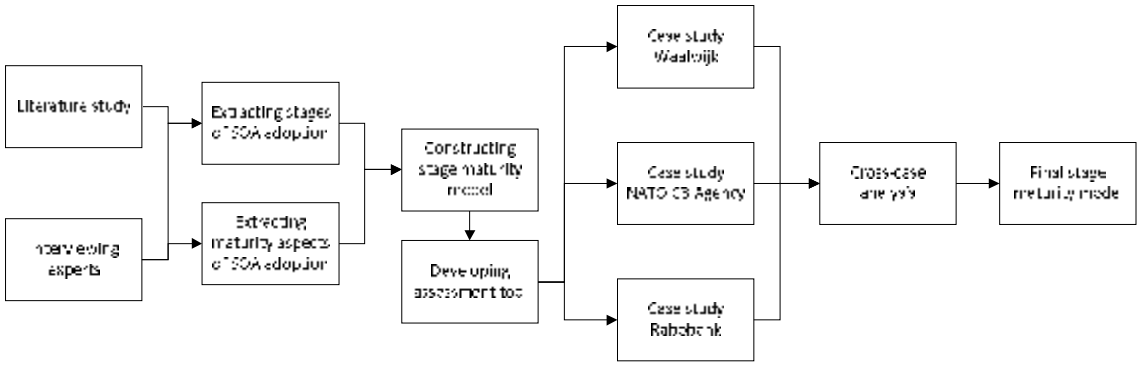

Figure 1: Research approach

Research question 1 is initially answered based on literature about SOA adoption and literature about adoption of IT technology. The former mainly comprises different case studies conducted with organizations adopting SOA and different stage maturity models for SOA adoption. The results from the literature review are extended and validated by interviewing experts. The experts that were interviewed can be categorized into three groups: IT architecture experts from a large IT consultancy organization that are involved in SOA-related implementations, experts on SOA adoption in practice and academic researchers that have published about SOA adoption. From the findings in theory and practice, six maturity stages are derived and defined.

Research question 2 is initially answered based on literature about SOA, SOA adoption, ERP implementation and general literature about IT adoption. The results from the literature review are extended and validated with experts and based on this this six maturity aspects are derived and defined. Within these maturity aspects or perspectives, 31 key indicators are identified that describe maturity on a more specific level.

Research question 3 is initially answered based on literature about SOA governance, IT strategy and business strategy. The results from the literature review are extended and validated with experts. The findings are then included into the stage maturity model as a maturity aspect and key indicators. Within the case studies semi-structured interviews were conducted to more in-depth insight about how SOA was integrated in the strategy of organizations.

The outputs of research questions 1, 2 and 3 were combined to construct the *Stage Maturity Model for SOA adoption* (SMM-SOA). The initially constructed model was validated and refined by interviewing experts. To support assessing the maturity of SOA adoption with SMM-SOA, a questionnaire was developed. This questionnaire comprises generic questions about the organization and its SOA adoption program and questions to assess the maturity stage on the 31 key indicators of SMM-SOA.

Using the methodology of Paré [29], a multiple-case study was conducted with three organizations, which were adopting an enterprise-wide SOA. Within each organization one person was selected that had a leading role with regard to the SOA adoption program. This person was asked to fill in the questionnaire. The results of the questionnaire were combined with a semi-structured interview, to interpret the results of the questionnaire in the context of the organization. The interviews were recorded, transcribed and coded to support analysis. A case study database was developed to separate the collected raw data from the analysis data. A case study database was constructed that contains the transcriptions, field notes and documents, which were relevant to the SOA adoption programs.

Relevant quotations within the interview transcriptions and other collected data were mapped to the elements of SOA-SMM (the stages, aspects and key indicators), in order to interpret the case studies individually and to analyze cross-case patterns. The coding of relevant quotations, which were derived from the transcriptions, was performed using the software program NVivo [30].

The multiple-case study as research method was appropriate to conduct this research for several reasons [29]. Firstly because enterprise-wide SOA adoption by organizations is a broad and complex phenomenon. The adoption involves many technical and organizational aspects and the people involved with SOA adoption have a limited understanding [5]. Secondly because the existing body of knowledge was insufficient [31]. SOA is a young technology and so the experiences are still limited. Thirdly because a holistic in-depth investigation was needed. SOA is an architectural style that is applied within the context of an organization, and so in-depth insight was required to answer the research questions. Fourthly because SOA adoption cannot be studied outside the organizational context in which it occurs.

1.4. Thesis structure

The remainder of this thesis is structured as follows. Chapter 2 presents the basic concepts of a Service-Oriented Architecture (SOA). It discusses the problem of integrating heterogeneous and distributed applications within organizations and how middleware is used to function as a bridge between these applications. Furthermore it discusses the SOA principles, and how these overcome the limitations of traditional middleware. The core Web Service standards are discussed to explain how they implement the SOA principles.

Chapter 3 discusses SOA adoption by organizations. It explains how the SOA principles can be transformed into concrete architectures and implementations. It discusses the importance of acknowledging organizational aspects of adopting an enterprise-wide SOA. Furthermore it explains the use of maturity models within the IT field, and compares current stage maturity models that are applicable for measuring the maturity of enterprise-wide SOA adoption.

Chapter 4 proposes the Stage Maturity Model for enterprise-wide SOA adoption (SMM-SOA), and defines its six maturity stages and six maturity aspects. The six aspects are further subdivided into 31 key indicators. A questionnaire is presented that functions as an assessment tool to measure the maturity of SOA adoption.

Chapter 5 reports the within-case analyses for each of the three case studies as proposed by Eisenhardt and Paré [29, 31]. For each case study the SOA adoption program is discussed in general to describe the organization-specific context. The results of each maturity assessment are discussed and structured based on the six maturity aspects. The results are interpreted using the semi-structured interviews.

Chapter 6 reports the cross-case analysis of the three different cases. It discusses general observations, and analyzes the cases structured according to the six maturity aspects.

Chapter 7 presents the conclusions of this research. It answers the three research questions, discusses the managerial and theoretical relevance, and discusses the results and limitations of this research according to the four general considerations for quality of research.

2. Service-Oriented Architecture

This chapter presents the basic concepts of a Service-Oriented Architecture (SOA). Section 2.1 discusses the problem of integrating application silo's within organizations. Section 2.2 discusses the use of middleware to support the integration of applications. Section 2.3 defines SOA and discusses how SOA can be applied to integrate application logic and it supports business process automation. Section 2.4 discusses how SOA is enabled by the core Web Service standards. Section 2.5 presents the conclusions of this chapter.

2.1. Integration of application silo's

The integration of IT applications is an ongoing issue within and between organizations [1]. The different applications within large organizations are typically heterogeneous: applications are developed on different computing platforms and the applications are running on different application servers [2]. The application portfolio of an organization typically includes commercial-off-the-shelf applications, legacy systems based on older mainframe technology, tailor-made applications and database management systems [27]. The IT functionality and data that together support the business are scattered throughout different applications. As the business environment is dynamic, this requires that the functionality within the different applications can be restructured to fit with changing business requirements.

In the traditional organizational model an organization is structured into functional departments like sales and procurement. These departments often build or buy standalone applications referred to as application silo's [32]. The silo's support a department within its boundaries and therefore it is customized to the departmental requirements. As each department manages its own applications, different application silo's exist within an organization and it is a difficult task to integrate these silo's [2]. From an organization-holistic perspective this is not a desired situation but the existence of silo's prolongs for different reasons. Some reasons relate to autonomy: departments do not want to lose power and project management wants self-contained projects to control for budgetary reasons. Some reasons relate to the offered technology: development methodologies are silo-based and third-party application packages often are compartmental. Some reasons relate to fear: stakeholders within the organization fear the development of large integrated applications and fear changing large existing applications. However, interaction between these silo's is needed to support automated business processes that span multiple business functions.

IT architectures are used to support structuring and integrating applications and so to manage the silo problem [32]. Architectures are used to define a high-level design of applications and their interactions. An architecture can be used as a framework to discuss implementation design and so to support architectural decision-making process. The scope of architectures differ as different levels of abstraction are needed. As a consequence, different architectures can co-exist within an organization. To keep control over these different architectures an enterprise architecture functions as a high-level overview of the different architectures within an organization. There are different stakeholders involved with defining and implementing architectures in relation to business process automation, ranging from business-oriented to IT-oriented.

There are four sub architectures included in the enterprise architecture to reflect the different stakeholders [33]:

- **Business architecture:** this describes the business strategy, governance, organization, and business processes;
- **Applications architecture:** this provides a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business processes of the organization;
- **Data architecture:** this describes the structure of an organization's logical and physical data assets and data management resources;
- **Technology architecture:** this describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services.

The business architecture represents the business perspective of business process automation. The requirements in the business domain are translated into business processes that can be supported by the different applications. The applications and data architecture represent the applications and their underlying data. Applications exchange and manipulate the data that is managed by underlying data management systems. The technology architecture functions as an infrastructure to support building and running the different applications and data management systems.

In the beginning of automation, business processes were supported by mainframe systems where the application logic and data management resided in the same system [4]. The typical architecture nowadays involves many distributed applications that interact over a network enabled by Web technologies and standards to communicate over different networks. Large organizations typically have a portfolio of hundreds of applications [27]. An important trend is the use of the Web to communicate with applications of external organizations, for example the communication with external applications in the supply chain [34]. The application logic and data resources reside in all of these internal and external applications and so functionality and data is highly distributed. As a consequence managing the applications and their interactions is a complex task.

2.2. Middleware

Different software solutions have been developed to facilitate and manage the interactions across heterogeneous computing platforms, this software is referred to as middleware [2]. Middleware functions as a bridge between applications by shielding the variety of methods to communicate and transfer data. In this way the different applications require no substantial change because the middleware software is added to act as a translator. Communication between applications is typically achieved by exchanging messages over a computer network. The middleware is used as a common channel to carry these messages between applications. In the context of application integration a message is a well-defined, data-driven text format that can be sent between two or more applications [3]. By exchanging messages applications can exchange data, invoke actions or notify another application of events that occurred within the sending application.

Figure 2 gives an overview of a situation with different application silo's that communicate using middleware. Every application silo has its own data system to manage the underlying data. The middleware acts as a liaison in the messages exchange between applications.

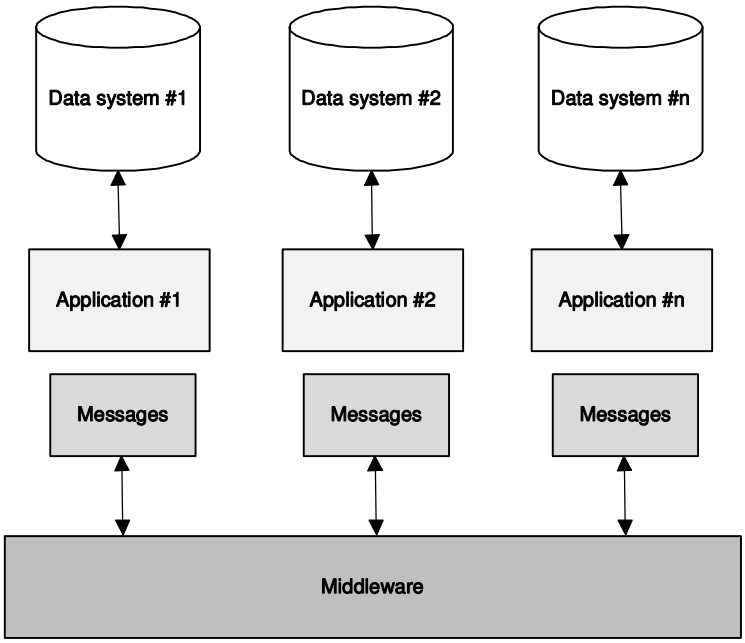

Figure 2: Application silo's communicate by exchanging messages through middleware [3]

There are two styles of message communication: synchronous and asynchronous communication [2]. When two applications communicate synchronously the sending application performs a request and waits for a response of the receiving application. In the mean time the sending application is locked in a thread and has to wait for a response before starting other activities. When applications communicate asynchronously this is performed using a message queue. A message queue can be compared to a mailbox: someone sends an email to a mailbox and the receiver of the mailbox chooses a moment to receive the email. When applications use a message queue to communicate, a sending application puts a message in a queue that can be retrieved by one or more receiving applications. In this way the sending application does not lock itself and can perform other tasks in parallel. Middleware that are based on asynchronous communication that use message queues are referred to as Message-oriented middleware [2]. Message-oriented middleware enable not only communication between two applications but also one-to-many and many-to-many communication. Asynchronous middleware is therefore more suitable to support distributed applications.

Although middleware enables the integration between applications, there are two limitations to middleware [2]. Application integration is traditionally focused on applications within the organization but the focus is shifting to integration with applications of external organizations. These are applications of external organizations like partners, suppliers, distributors and customers [34]. A limitation of middleware for using it in intra-organizational integration is that there is no obvious way to put the middleware. This means that the cooperating organizations need to agree on which specific middleware platform to use and who should operate the middleware. Although it is technically feasible to integrate cross-organizational applications using middleware it is disabled by

the lack of trust between organizations, the autonomy that each organization wants to preserve and the confidentiality of transactions.

Another limitation of middleware is the lack of standardization between different middleware platforms. Middleware platforms implement the idea of message exchange but these platforms support different technologies and standards to do this. These standards include standards about message formatting and about the protocols to exchange the messages. Middleware platforms of different organizations can be incompatible as they support different standards. The incompatibility problem can also show up in one organization that uses different middleware platforms that are incompatible with each other.

2.3. SOA principles

A Service-Oriented Architecture (SOA) is a solution to overcome the limitations of traditional middleware. The shift to integrate not only with internal but also with external applications is enabled by SOA. This shift was driven by the existence of Web technologies to communicate between these applications. The Web is characterized by its widely adopted standards like TCP/IP and HTTP. The Web Service standards emerged as complementary standards to integrate heterogeneous applications within and across organizations. SOA is enabled by using these Web Service standards. Section 2.4 discusses Web Services and how they enable SOA. This section discusses SOA from a theoretical perspective.

SOA is based on the idea of service-orientation. Service-orientation is a general way of thinking in needs and capabilities [35]. A well known example of service-orientation is a free economic market. In a free economic market services are offered by service providers to service consumers that have a need. People and organization create capabilities to solve or support a solution for the problems they face in the course of their business. The needs of one person or organization can be solved by the capabilities of another by delivering a service. For example, telecommunication providers offer communication services to consumers that want to communicate with each other. Service-orientation presents an ideal vision of a world in which capabilities are cleanly partitioned and consistently represented. In this vision service consumers are easily able to switch to other providers that deliver (almost) the same service based on differences in price and quality of service. This requires a transparent market where service consumers are aware of all services that are offered.

When applying service-orientation to business process automation, capabilities are divided into business and application logic [4]. The business logic is represented by business processes. A business process is a set of logically related activities performed to achieve a well-defined business outcome [3]. A business processes is commonly modeled as a sequence of activities. It is initiated by the request of a service by a customer and ends with the delivery of a service to the customer.

Business processes can be automated by the support of application logic. Application logic represents the functionality that is offered by the different applications within an organization. Applications process underlying data to support the execution of activities within business processes. The aim of business processes automation is to fully automate business processes. However, in practice business processes automation is a difficult task caused by the dynamic business environment and the application silo's problem [34]. In terms of service-orientation these are problems related to respectively the needs and capabilities within business process automation. The needs of the business constantly change under the pressure of competition in the business environment. As a

consequence business processes have to change and this requires flexible support of applications. Caused by the application silo's problem this flexible support is a difficult task to achieve: functionality of applications cannot easily be decomposed and reconstructed to fit with the changing business needs [34]. Especially automating business processes that span different business functions is a difficult task and is often performed manually [2].

The flexibility that is needed can be offered by a Service-Oriented Architecture (SOA). SOA is an architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner [3]. In a SOA the business and application logic are cleanly partitioned and consistently represented as services [4]. Services can then be composed to construct and reconstruct automated business processes.

Adopting a SOA requires that the existing business and application logic are decomposed into services, this is shown in Figure 3. The middle layer represents the services.

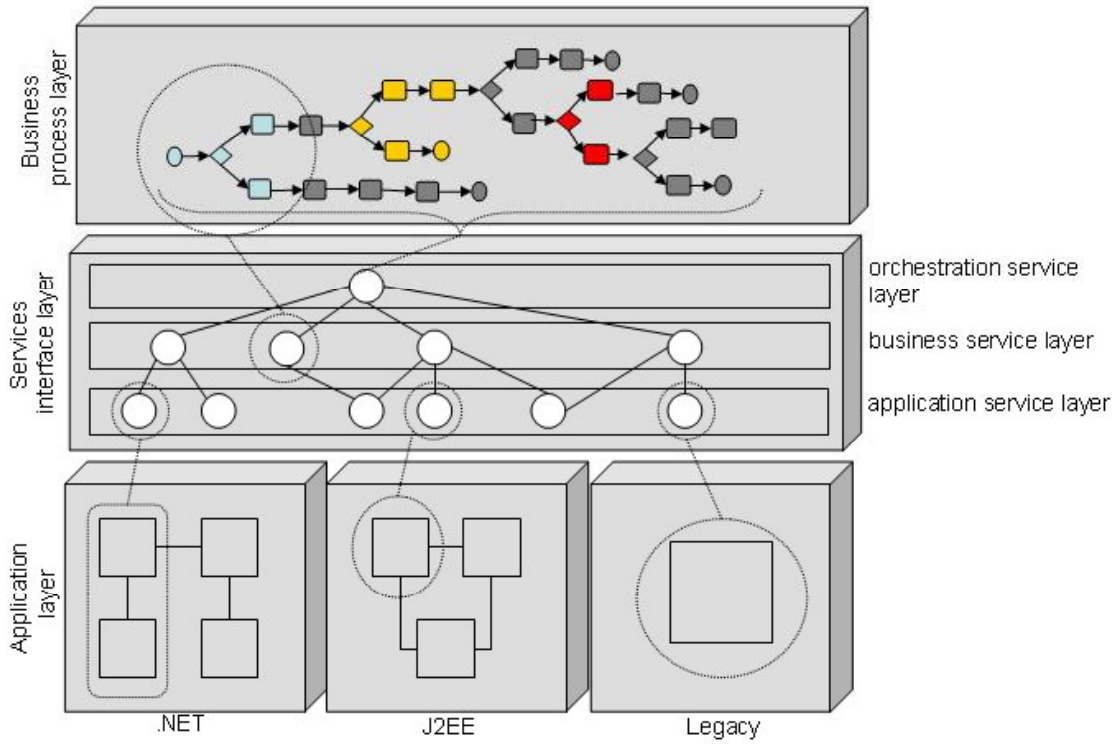

Figure 3: Applying service-orientation to business and application logic [4]

In the top layer the business logic is represented as a business processes. A business process is commonly modeled as a sequence of activities. The sequence can be decomposed into business services represented in the business service layer in Figure 3. A business service is a logical representation of a repeatable elementary business activity that has a specified outcome [36]. Because the outcome of business services are specified, they can be reused within different business processes.

In the bottom layer in Figure 3 the application logic is represented by different applications. These applications reside in different computing platforms, represented in Figure 3 by the three larger

boxes. Functionality within these applications can be decomposed into application services that are represented in the application service layer in Figure 3. An application service is defined as a logical representation of repeatable application logic [4]. An important question in relation to decomposition into services is what the granularity of a service should be [37]. To decide this, organizations have to make trade-off decisions between flexibility of business processes, reusability and performance [38]. Although service granularity is an interesting topic of interest, it is beyond the scope of this research.

Communication between application services involves service-to-service interaction. Application services communicate on the application logic layer and they are not directly responsible for the interaction with the end user of an application. The latter one is covered by the presentation layer that is positioned on top of the application logic layer of an information system [2]. Application services exchange information through the service interface and they communicate by exchanging messages as can be seen in Figure 4. The service description functions as the manual of a Web Service: it describes what it does, where it resides and how it can be invoked. Service descriptions are published in a service registry to be discovered by other application services. In this way application services are aware of each other. The application service that invokes another application service performs the role of service requestor, the service provider delivers the service. One application service can at the same time perform both roles: a service acts as a service requestor when invoking other services and as a service provider when being invoked by other services.

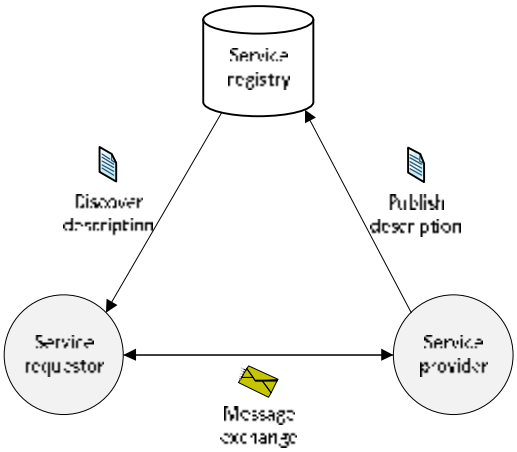

Figure 4: Interaction between two application services [4]

Application services act as a standard access layer on top of existing applications. The functionality within these existing applications is broken down into application services that can be discovered and reused by other application services. The standards for message formatting and service description enable application development based on functionality that resides in different application silo's. By using common standards the role of an intermediate translator becomes unnecessary.

Business and application services in a SOA are more extensively characterized by eight common principles [4, 35, 36]:

- Reusability: a service is designed to support potential reuse. Application functionality can be reused within different automated business processes. Standards for message exchange and service descriptions are the key enabler for this principle, so that services can be used within other internal and external applications;

- Service description: a service shares a formal description that specifies what the service does, where it resides and how it can be invoked. By using a standard for describing services, functionality within different heterogeneous applications can be accessed in a uniform way;

- Loosely coupling: a service is designed to interact with minimal dependencies on other services. In this way a service that is part of an automated business process can be replaced by another service that delivers (almost) the same functionality;

- Abstraction: the only part of a service that is visible to the outside world is what is exposed via the service interface. The service acts as a black box from which the in- and outputs are defined by the service description. The internal application logic of a service is shielded from the service requestor;

- Composability: a service may compose other services to increase the degree of granularity. A service then combines the functionality of other services into a composite service that is more coarse-grained. By composing and orchestrating services, business processes can be easily be constructed and reconstructed;

- Autonomy: a service has control within a certain boundary of logic and it has minimal dependency on other services. It has the resources available to execute its tasks and does not have to account for its activities;

- Statelessness: a service does not manage state information with regard to the service requestors. This enables services to be reused within different orchestrated business processes. The coordination of different interacting services in a business process resides in the process logic and not in the application logic underlying a service;

- Discoverability: a services allows its descriptions to be discovered and understood by service requestors. To be discovered the service descriptions are published in a service registry where they can be discovered by potential service requestors.

The eight service principles enable services to be used as flexible building blocks that can fulfill dynamic business needs. In a SOA business processes are automated by orchestrating application services. Orchestration defines in which sequence application services are invoked and how they interact to support a business process [3]. Orchestrated business processes are represented in the orchestration service layer shown in Figure 3. The process logic that defines the sequence in which services are executed resides in the orchestration of a business process. The loose coupling of services enables them to be added to, removed from or reordered within the sequence.

2.4. Web Services

The SOA principles are enabled by implementing it with Web Services. The idea of offering services using Web technologies finds it origin in the concept of Software-As-A-Service (SAAS) [39]. In SAAS, application service providers offer services on a subscription basis to customers using different Web technologies. The application, data and presentation logic then reside at the application service provider and the customer can just start using the system over the Web without having to implement an application. Web Services are a specific form of SAAS. They are not directly used by end users of a system but they reside in the application logic layer. Communication between Web Services involves service-to-service interaction. Web Services are the implementation of application services in a SOA. A Web Service is a platform-independent, loosely coupled, self-contained, programmable Web-enabled application that can be described, published, discovered, coordinated, and configured using XML artifacts for the purpose of developing distributed interoperable applications [1]. Web Services are implemented using a set of Web Service standards and by using standard Internet protocols as HTTP and TCP/IP as fundament. The purpose of the Web Service standards is to achieve interoperability between distributed applications, therefore open and accepted standards are included.

The three core standards SOAP, WSDL and UDDI make up the first-generation Web Service standards and are the fundament of basic interaction between Web Services. Over time complementary standards were added on top of the core standards to deal with different issues like addressing, routing, coordination, quality of service and security. These standards are referred to as the second-generation Web Service standards.

Figure 5 shows how the three core standards implement application services. Message exchange between Web Services is defined by the SOAP standard [40]. SOAP is a XML-based message framework that defines message formatting and message exchange between Web Services. Service description is defined by the Web Service Description Language (WSDL) standard [41]. A WSDL specification defines what a Web Services does, how to invoke and where it resides. The service registry is defined by the Universal Description, Discovery and Integration (UDDI) standard [42]. UDDI is a platform-independent, XML-based registry to publish and discover Web Services.

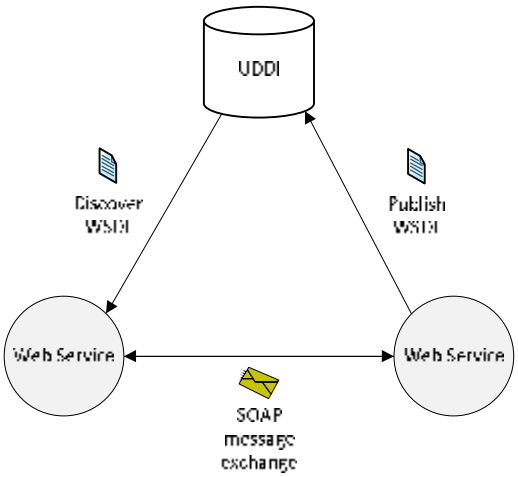

Figure 5: Implementation of application services by the core WS standards [4]

The three core standards describe the basic interaction between two Web Services but do not specify the complex interaction between Web Services that is needed to construct business processes. The Web Services Business Process Execution Language (WS-BPEL) is a complementary Web Service standard used to specify the orchestration of business processes based on Web Services.

The following sections describe how SOAP, WSDL, UDDI and BPEL enable business process automation by implementing the SOA principles. The sections provide a basic understanding of these four standards and are not aimed at giving an in-depth specification of the standards. Furthermore, there are other complementary Web Service standards that deal with a variety of issues but these are beyond the scope of this research. The first and second generation Web Service standards have already been specified and discussed in detail in literature [2-4].

2.4.1. Message exchange: SOAP

Exchange of information between Web Services is implemented using the SOAP standard [40]. The purpose of SOAP is to enable message exchange between applications regardless of their operating systems and computing platforms. SOAP is a wire protocol that defines communication between applications. A wire protocol needs an underlying transport protocol that communicates between systems facilitating the transport within a network. HTTP is the common transport protocol that is used to carry SOAP messages because it is not blocked by firewalls and is widely adopted. Figure 6 shows the different layers of communication. The TCP/IP stack is the fundament of communication between network computers, on top of this is a transfer protocol like HTTP. On top of the transport protocol are the SOAP messages that are exchanged between Web Services through their service interfaces (WSDL is explained in section 2.4.2).

Figure 6: The SOAP messaging framework [3]

The SOAP message format specifies the elements contained within a message. A SOAP message is a XML document consisting of the elements that are shown in Figure 7. The XML root element of a SOAP message is the envelope element. The envelope may include a header element and must include a body element. The body element represents the payload and contains application-specific XML-data that is exchanged between Web Services. In case of a fault in the execution of the service provider, the body element contains the specification of this fault instead of the data. The header element includes header blocks that contain detailed information about how to process the message.

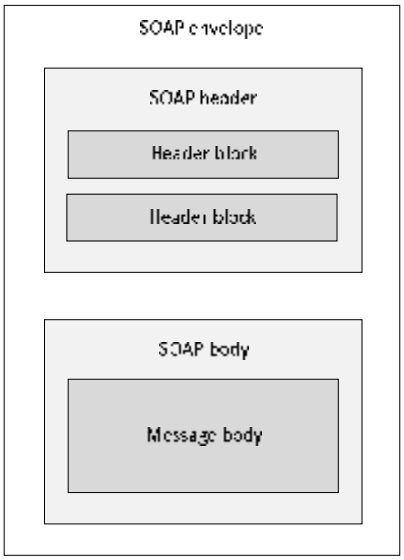

Figure 7: Elements of a SOAP message [2]

SOAP messages are sent between two endpoints but may be forwarded by intermediary SOAP nodes along their message path. The intermediary nodes can process one or more header blocks and then forward it to another intermediary node or to the ultimate receiver.

SOAP defines two message exchange patterns: one-way messaging or request-response messaging. In one-way messaging SOAP messages only travel in one direction from the sender to the receiver. In request-response messaging the sender sends a message to the receiver and the receiver is expected to send a message back. These two message exchange patterns can be used as a foundation to construct longer running dialogues between two Web Services but this is not defined within the SOAP standard.

The two message exchange patterns describe the flow of messages between Web Services. This in contrast to communication modes that describe the payload of messages. SOAP defines two communication modes: Document/Literal and RPC/Literal. The former represents asynchronous communication and the latter one synchronous communication. In the Document/Literal mode the body element contains XML-structured application data. The receiving Web Service processes the data and the sender does not have to lock but can go on executing other tasks. In the RPC/Literal mode the sending Web Service invokes a method with parameters that are contained in the body element. The receiving party can send back return values after processing the parameters and invoking the method call.

2.4.2. Service description: WSDL

The Web Service Description Language (WSDL) is a standard that defines how to describe Web Service and how data may be exchanged between interacting services [41]. A WSDL specification is an XML-based specification schema for describing what a service does, where it resides and how to invoke it. WSDL specifications can be published by service providers to a service registry to be discovered by service requestors as shown in Figure 5.

A WSDL specification can be divided into two parts: an abstract part that defines the service interface and a concrete part that implements the abstract interface. The abstract part consists of four types of elements as shown in Figure 8. A Port Type element is a logical collection of related operations. Each Operation element defines the exchange of SOAP messages between the service requestor and the service provider. The operations offer the functionality of the Web Service and they are invoked by service requestors. Types elements are used to specify the data types that describe the data send by the messages.

The abstract part does not include the details of the implementation, this is done by the concrete part. The concrete part tells at what address the service is located. A Binding element binds the abstract Port Type to an existing implementation and provides information about the protocol and the concrete data formats expected. The Service element specifies a specific network-addressable location so that a requestor can bind to the service. A Service element has one or more ports that represent the endpoints of the communication.

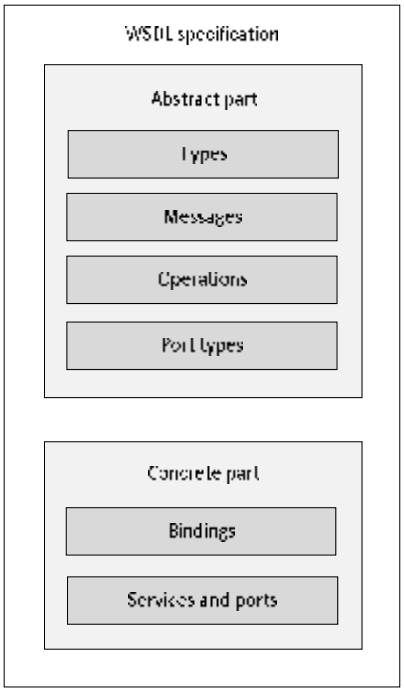

Figure 8: WSDL specification [2]

The functionality of a Web Service is offered through its operations. An operation can exchange SOAP messages with another Web Service in four different ways:

- One-way operation: the service requestor sends a message to the service provider and no response is needed;
- Request-response operation: the service requestor sends a request message to the service provider and a response is send back;
- Notification operation: the service provider sends a message to the service requestor and no response is needed;
- Solicit-response operation: the service provider sends a request message to the service requestor and a response is send back;

These basic interactions are the fundament for more complex interactions involving many operations and Web Services.

2.4.3. Service registry: UDDI

The Universal Description, Discovery and Integration (UDDI) is a standard that defines a platform-independent, XML-based registry for Web Services [42]. The UDDI registry is used by service providers to publish WSDL specifications and by service requestors to discover Web Services. UDDI functions as a phone book for organizations to discover Web Services that are offered within and beyond the organization. The initial goal with UDDI was to support worldwide registries where every organization could publish and discover service descriptions. As of version 3, the UDDI standard started to include support for private registries within organizations and for semi-public registries between partner organizations. There are three types of information that can be found in the UDDI registry analogue to a phone book [2]:

- White pages: listing of organizations, their contact information, the Web Services they provide and their known identifiers;
- Yellow pages: classifications of the organizations and the Web Services according to standardized or user-defined taxonomies;
- Green pages: technical information about how Web Services can be invoked.

UDDI includes an XML schema to store the information of these three categories. There are four main data entities within the XML schema as shown in Figure 9. The businessEntity describes an organization that provides Web Services by general information about the organization. Organizations can also be assigned to a category like the type of industry. The businessService describes a Web Service that is offered by a businessEntity. The technical information needed to use a Web Service is described by a bindingTemplate. A bindingTemplate defines the address at which the Web Service is available along with references to tModels. A tModel is used to describe dynamic information like the identifiers, the categories and the reference to service descriptions. Actual service descriptions are not stored within the UDDI registry, only the tModels that refer to service descriptions that reside at the service provider. An actual service description like a WSDL specification is referred to as an overviewDoc. It is not mandatory to use UDDI in combination with WSDL for service specification but WSDL is the most widely adopted standard.

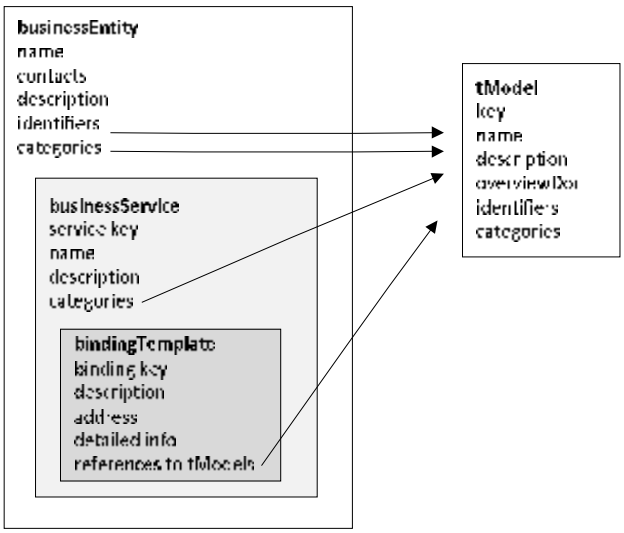

Figure 9: Entities in an UDDI registry entry [2]

UDDI offers APIs to service providers and service requestors to respectively publish and discover Web Services. The Publishers API includes operations for service providers to add, modify and delete entries in the registry. The Inquiry API includes operation for service requestors to find registry entries based on search criteria and to get information to connect to the Web Services that are discovered in the registry. To interact with an UDDI registry service providers and requestors need to exchange SOAP messages specified by the APIs.

2.4.4. Business process orchestration: WS-BPEL

The Web Service Business Process Execution Language (WS-BPEL, or in short BPEL) is a standard for specifying automated business processes based on Web Services [43]. As Web Services act loosely coupled and do not include process logic, the execution of their operations must be coordinated to compose the Web Services into a business process. BPEL is used to coordinate different Web Services and so to compose them in a coherent whole. Although BPEL is not the only standard for composing Web Services, it is the most widely adopted standard.

A BPEL process is a XML document that specifies the orchestration of Web Services [2]. Orchestration defines the order in which different Web Services are invoked. The different Web Services that are involved in a BPEL process are invoked by executing BPEL activities. A BPEL process specifies the flow of these activities and is layered on top of WSDL, with WSDL defining the specific operations allowed and BPEL defining how the operations can be sequenced. BPEL supports primitive and structured activities [44].

Primitive activities are the simplest form of interaction with a Web Service. The main primitive activities include:

- Invoking other Web Services, using the tag <invoke>;
- Receiving a request from another Web Service, using <receive>;
- Sending a response, using <reply>;
- Manipulating data variables, using <assign>;
- Waiting for some time, using <wait>;
- Terminating the business process, using <terminate>.

The primitive activities can be combined into more complex activities using structured activities. Structured activities describe how an orchestrated business process can be created by composing the basic activities it performs into structures. The main structure activities include:

- Invoking a set of primary activities in a certain order, using <sequence>;
- Invoking a set of primary activities in parallel, using <flow>;
- Case switching based on a certain condition, using <switch>;
- Looping an activity, using <while>;
- Acting on received events, using <pick>.

The primitive and structured activities are used to coordinate the operations of the different Web Services that are involved. The Web Services that are involved in the BPEL process are modeled using the tag <partnerLink> as shown in Figure 10.

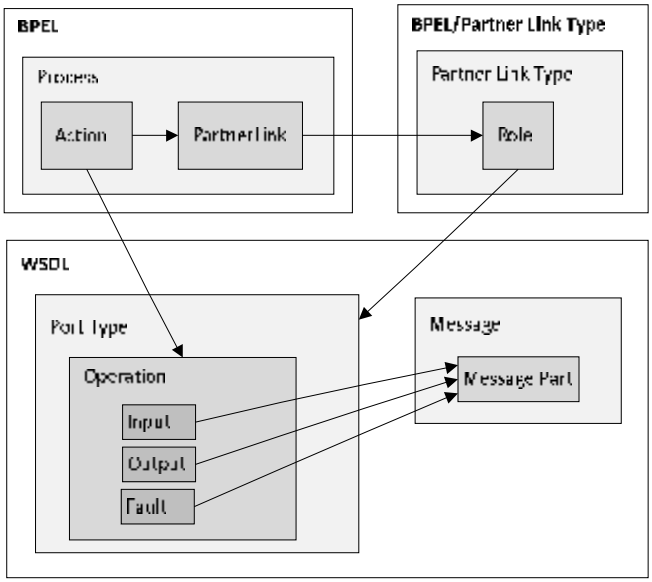

Figure 10: Partner Link Types in BPEL [2]

A partnerLink defines the Role of the BPEL process and the involved Web Service and the service interface that each provides. Each partnerLink can be assigned one or two Roles. Invoking an asynchronous operation requires that two Roles are defined, one for invoking the operation of the Web Service and one for a callback invocation of the BPEL process by the Web Service that was invoked. Invoking a synchronous operation requires that one Role is defined because the BPEL process waits for the response of the Web Service. A Role refers to the Port Type that contains the operations that are invoked by the BPEL process.

BPEL can be used to specify abstract and executable business processes. An abstract business process defines the SOAP message exchange between Web Services but reveals the internal business logic of the BPEL process. Specifications of abstract business processes include information like when a Web Service should wait for messages and when a Web Service should send messages. An executable business process implements an abstract business process and includes detailed information to do this. This detailed information comprises the interactions and behavior of the involved Web Services.

2.5. Conclusions

This chapter discussed the basic concepts of a Service-Oriented Architecture (SOA). Adopting a SOA is a solution to integrate the heterogeneous applications in an organization. SOA is an architectural style used to decompose application silo's into application services, and business processes into business services. By using standards for message exchange, service description and a service registry, application services can be reused throughout and outside of the organization. As services are reusable, business processes that span different business functions can be automated. Business processes can be automated by orchestrating application services. Business processes are then implemented as chains of services, in which services can be added, removed or switched.

SOA is enabled by the Web Service standards, which are open and accepted standards. Using Web Services, application services are interoperable and can be reused as flexible building blocks to develop new applications, and so to cope with ever changing business requirements.

3. SOA adoption

This chapter discusses SOA adoption by organizations in practice. Section 3.1 discusses how the abstract SOA concepts are translated into SOA implementations. Section 3.2 presents a literature review of earlier field research about SOA adoption and discusses the importance of organizational aspects for successfully adopting an enterprise-wide SOA. Section 3.3 discusses the use of maturity models for the adoption of IT technologies. Section 3.4 compares four stage maturity models that can be used as stage maturity models for enterprise-wide SOA adoption. Section 3.5 presents the conclusions of this chapter.

3.1. From concepts to implementations

SOA is an architectural style that can be applied to the business and application architecture. To do this, organizations need to translate the abstract SOA concepts into concrete architectures and implementations. They need to define the granularity of services and they need to choose standards, technology and software to support service description, a service registry, message exchange between services and business process orchestration. SOA implementations are typically build on top of existing applications and not in a greenfield situation. As a consequence, implementations are environment-specific and constrained or enabled by that environment [36]. To guide the SOA adoption in the organizational context, different policies are needed to deal with expected and unexpected issues. These policies need to guide the SOA lifecycle ranging from the design, development and operation of a SOA.

Architectures with different levels of abstraction are used to support the translation of the basic SOA concepts into concrete SOA implementations. Figure 11 shows these architectures and other related entities, which are used for translation. The SOA reference model that describes the SOA principles is positioned on the top. The concrete SOA implementations are positioned on the bottom.

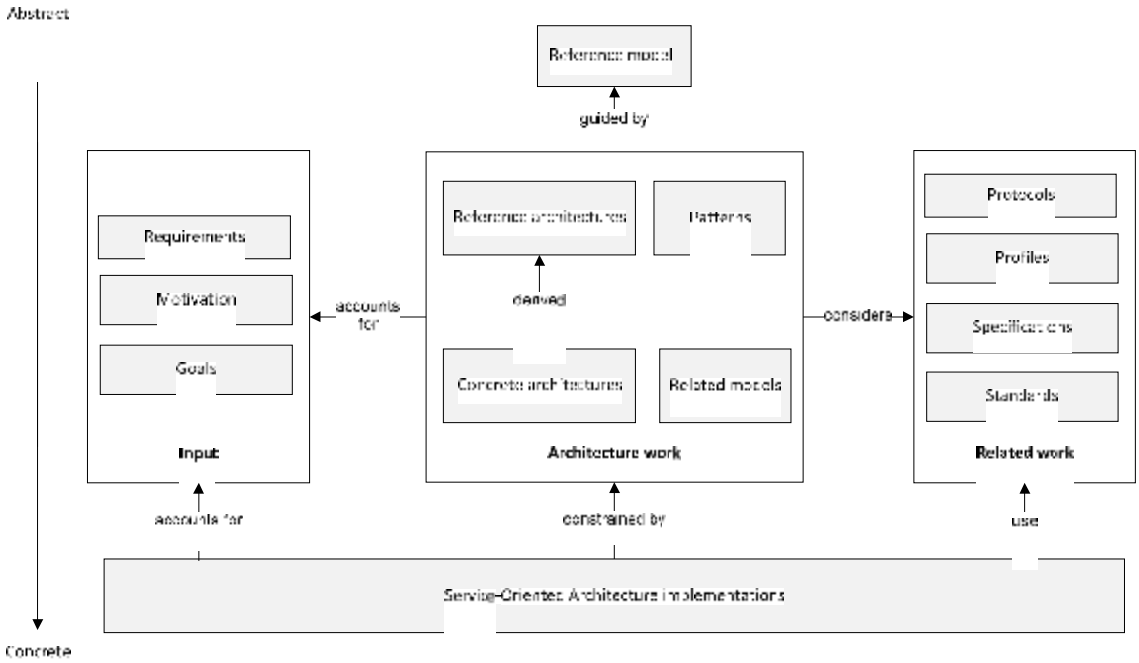

Figure 11: Translating the SOA principles in concrete implementations [35]

A SOA reference model is not directly tied to any standards, technologies or other implementation details. As a reference model is generic, it can be used by different types of organizations as a generic design template. With regard to the architectures two levels of abstraction are shown in Figure 11: reference architectures and concrete architectures. A SOA reference architecture is guided by a reference model and is used within an organization as a design template for concrete architectures. Typically one or more reference architectures guide concrete architectures within specific projects. SOA implementations are developed based on concrete architectures.

To better understand the entities and their interrelations shown in Figure 11, the reference architecture of the Dutch public sector, called NORA, is taken as example [45]. NORA provides guidelines for more than 1.600 public administration bodies, which use NORA as a SOA design template to develop concrete architectures. The *requirements* for NORA originated from citizens, business and politics. For example, one requirement is that citizens and businesses have to register their data only once at a front office. The *motivation* for NORA originates from plans of the Dutch government that wants to act decisively, efficiently and customer oriented based on IT. The NORA *Goals* originate from different guiding sources like the Dutch government program "Andere overheid" (English: Other public administration). An example of a goal is that citizens are free to choose which channel they use to communicate with the public administration. NORA considers the context of a pre-defined environment with the *protocols, profiles, specifications* and *standards* that are pertinent. An example of a predefined environment is the use of DigiD, which is the Dutch citizens authentication system for communicating with the public administration. Public administration bodies can use NORA and other *reference architectures* together with *architectural patterns* and *related models* to develop *concrete architectures*. Specific SOAs implementations are developed based on concrete architectures. This example shows the difference between the SOA principles and the translation to architectures and implementations in organizations. While the reference model mainly describe components in a general way, architectures and SOA implementations require that choices are made about which standards, technology and software to use.

3.2. Technical and organizational issues

Adopting an enterprise-wide SOA is a large undertaking for organizations. Many architectural decisions have to be made to translate the basic SOA concepts into SOA implementations. The implementations are typically developed on top of the existing heterogeneous applications which are distributed over different departmental boundaries. The aim of SOA is to integrate these applications and to support business processes with distributed applications that span different business functions. Technology is not the only factor to successfully adopt an enterprise-wide SOA. There is consensus in literature that organizational issues have a significant influence on the success of SOA adoption.

Table 1 gives an overview of our literature review, it presents the field research that was conducted by different authors. The Research Methods (RM) that these authors use, are divided into Action Research (AR) and Case Studies (CS) [29, 46]. As can be seen in Table 1, most authors used a case study as research method.

Table 1: Field research about SOA adoption

Research	Year	Case(s)	RM	Research goal
SOA in the real world [10]	2005	Consumer lending organization	AR	Evaluating experiences in building an enterprise business application based on SOA
Public services integration & interoperability [47]	2006	4 Dutch municipalities	AR	Determining contribution of an ESB and SOA to interoperability for the government and deriving a roadmap for ESB and SOA adoption
SOA and the enterprise [9]	2006	US city government	CS	Evaluating experiences of three years of SOA adoption
BPM on top of SOA [7]	2007	Danske bank	CS	Evaluating experiences from four years of BPM and SOA adoption
Common misconceptions about SOA [5]	2007	Organizations (potentially) adopting SOA	CS	Identifying misconceptions about SOA adoption
SOA adoption in practice [11]	2007	Deutsche Post Brief, Credit Suisse, T-com, Zuger Kantonalbank	CS	Evaluating how the SOA concepts are applied in practice
SOA in Nederland [6]	2007	12 Dutch organizations	CS	Evaluating applicability and benefits of SOA adoption
Where to start with SOA [48]	2008	5 German chemical organizations	CS	Defining decision supporting criteria for successful SOA projects

Although the scope of each work differs, evaluation of experience generally acknowledges the importance of organizational changes that have to be made to successfully adopt an enterprise-wide SOA. Implementing SOA on an enterprise-wide scale influences the applications that are distributed over the different departments and the business processes that span multiple business functions. The different departments have to strongly cooperate to enable the reuse of application services throughout the organization and to orchestrate business processes based on services. Changes have to be made within the organization to support SOA adoption. For example, another way of funding is needed to support the development and operation of application services. Popular services increase the costs of operation and this needs to be billed to the service users.

The management of the variety of organizational issues related to SOA adoption is referred to as SOA governance [4]. When experimenting on a small scale with SOA, organizations typically do not pay much attention to the organizational aspects of SOA and different issues are solved ad hoc [49]. But when the scope of SOA becomes enterprise-wide and application services are reused throughout the organizations, the management of technical and organizational aspects becomes essential.

The variety of technical and organizational issues involved create a complex transformation process. Caused by this complexity, the stakeholders within organizations typically have a limited understanding of the impact of SOA adoption [5]. Furthermore, adopters tend to oversimplify the expected effort that is required for an enterprise-wide SOA adoption. Earlier field research shows that SOA is not just a simple and quick solution for the application integration problems [7, 9]. Enterprise-wide SOA adoption is a long-term transformation process and the SOA benefits do not

showcase on the short run. Without understanding the impact of SOA adoption, there is a risk of losing commitment in the organization for the SOA initiative as it does not showcase immediate success [48]. This is reflected by the fact that a significant part of organizations have been dissatisfied with their SOA initiatives [17], and in some cases organizations cancelled their SOA initiatives [6].

The consensus in literature is that organizations can best take a gradual approach to successfully adopt an enterprise-wide SOA. The gradual approach allows an enterprise to mitigate the risks and to minimize the impact on the end users of applications [5]. Furthermore, when taking a gradual approach, SOA can showcase itself within projects to gain management commitment. A lack of management commitment is seen as a major risk for successful adoption [48].

3.3. Maturity models in the IT field

To support a gradual approach, maturity models can be used. Maturity models guide the transformation process needed to adopt IT technologies like SOA. In the literature there is no generally agreed definition of a maturity model. The use of the term maturity is derived from maturity of humans and organisms. Maturity is defined as the state, fact or period of being fully grown or physically developed [50]. Humans grow up and walk through the different development stages (e.g., infancy, childhood, adolescence) to fully grow up. Although the actual process differs from person to person, there is a high level of commonality between humans in their development.

Maturity models are popular within the IT field. Two types of maturity models can be distinguished, namely process maturity models and stage maturity models. In a process maturity model, the maturity of organizational processes is considered. The goal of a process maturity model is to improve the quality of processes. The most well known process maturity model is the Capability Maturity Model Integration (CMMI) [18]. CMMI considers the maturity of software development and software maintenance processes within organizations. The maturity of processes ranges from level 1, where processes are performed ad hoc and chaotic, to level 5, where processes are continuously being improved. The premise underlying CMMI is that if the quality of processes is high, the quality of the produced software system is also high. Although process maturity models focus on improvement of software systems, they lack attention for architectural processes and do not measure enterprise-wide adoption of technology [51]. Furthermore, they lack attention for the automation of business processes using IT.

Stage maturity models focus on the adoption of IT technology by organizations. Their goal is to give a quick understanding of the adoption process from different perspectives. This is done by breaking up the total process in different stages and considering different aspects of maturity. Stage maturity models are applicable within this research because they measure the maturity of IT adoption from an organization-holistic perspective and measure the quality of IT technology products like architectures and derived implementations. The process of adoption of IT technology is typically long-term and has a risk of failure. The main function of a stage maturity model is to function as a roadmap for organizations and to improve understanding.

A stage maturity model is constructed based on three key elements:

- Stages: growth steps in the adoption process ranging from the initiation of adoption to full development;
- Maturity aspects: relevant perspectives that reflect a specific maturity of IT adoption. For each maturity aspect the current maturity stage can be assessed. This includes aspects that focus on the technology and aspects that focus on the organization;
- Maturity assessment tool: an instrument to assess the maturity of an organization in terms of stages and maturity aspects.

In the first stage, an organization has initiated the adoption while in the last stage full development (maturity) is reached. A stage maturity model supports in positioning an organization in its current stage, and determining the target stage to reach. A stage maturity model is commonly presented as a matrix model. A representation of the stage maturity model for electronic data processing is shown in Table 2. On the horizontal axis the six maturity stages are shown ranging from initiation to maturity. On the vertical axis the four maturity aspects are shown. The cross point of one aspect and one stage, the cells of the matrix, represent the progress on one maturity aspect.

Table 2: Stage maturity model for electronic data processing represented as matrix model [20]

	Stage 1: initiation	Stage 2: contagion	Stage 3: control	Stage 4: integration	Stage 5: data administration	Stage 6: maturity
Applications portfolio	Functional cost reduction applications	**Proliferation**	Upgrade documentation and restructuring of existing applications	Retrofitting existing applications using database technology	Organization Integration of applications	Application integration "mirroring" information flows
Data processing organization	Specialization for technological learning	User-oriented programmers	Middle management	**Establish computer utility and user account teams**	Data administration	Data resource management
Data processing planning and control	**Lax**	More lax	Formalized planning and control	Tailored planning and control systems	Shared data and common systems	Data resource strategic planning
User awareness	"Hands off"	Superficially enthusiastic	Arbitrary held accountable	Accountability learning	Effectively accountable	**Acceptance of joint user and data processing accountability**

The progress on specific maturity aspects of an organization may differ as can be seen in Table 2 by the underlined cells. The underlined cells represent the current maturity stage on a specific maturity aspect. As a consequence of an assessment, an organization can develop a plan for improvement on specific aspects. For example, an organization targets to improve the applications portfolio aspect from the current stage (proliferation) to stage 4 (retrofitting existing application using database technology). A stage maturity model should support diverse organizations, and therefore cannot be

adjusted to the specific situation of every organization. Some maturity aspects may not be relevant for a specific organization. An organization can derive a roadmap that translates a stage maturity model to the specific situation and priorities of the organization. In this translation maturity stages can be skipped and complementary stages can be added to the organization-specific roadmap.

Stage maturity models find their origin in research on the adoption of Electronic Data Processing (EDP) in organizations in the seventies [20, 21]. At that time, EDP was introduced as alternative for paper work. The stage maturity model that was constructed for EDP adoption is shown in Table 2. An interesting analysis within the research was that cumulative adoption of EDP typically followed an s-curve, which has been inspired by the diffusion of innovations theory [16]. The s-curve of cumulative adoption was also observed with the adoption of Enterprise Resource Planning systems (ERP) [22].

Early adopters select the technology first, followed by the majority, and ending with laggards until the technology is common. this can be seen in the right curve in Figure 12.

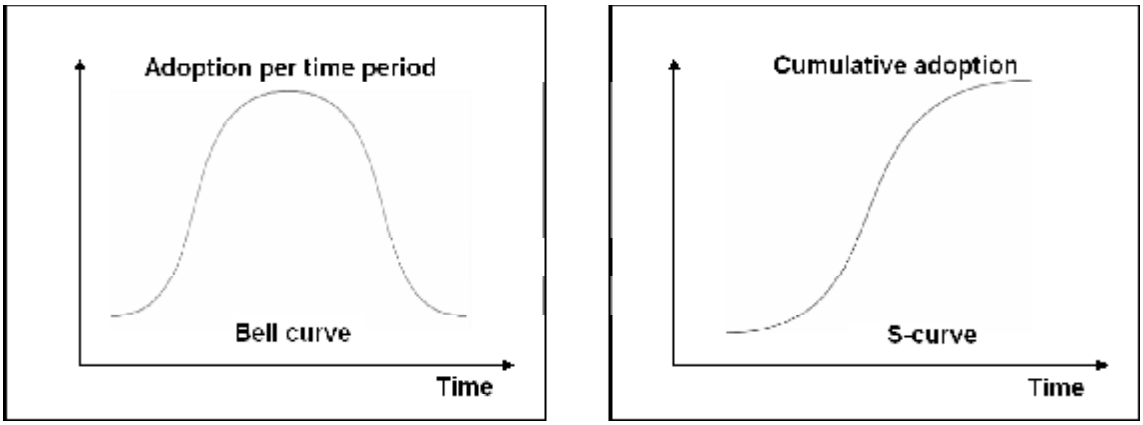

Figure 12: Curves of technology adoption and cumulative technology adoption [52]

The left curve in Figure 12 shows the adoption of the technology over time with a peak in the middle representing the adoption of the technology by the majority. The right curve represents the cumulative adoption, where the rise in the middle represents the adoption by the majority. People, departments and organizations are more likely to adopt a technology if others they see as an example adopt it. This is confirmed by research on acceptance of information technology, which stated that social influence is an important factor in the acceptance of information technology by employees [53]. The imitation behavior also applies to SOA, not only between organizations but also within organizations between departments. This is reflected by the fact that 55% of organizations run SOA pilots to showcase the success within the organization [13].

Different stage maturity models for measuring the progress of IT adoption emerged over time. A stage maturity model for the implementation of Enterprise Resource Planning (ERP) systems appeared in 2001 [22]. With the introduction of SOA as technology, different SOA stage maturity models were proposed mainly developed by software vendors and consulting firms. As a consequence of the creators, these models function as a sales tool or as an instrument to identify clients' needs.

The benefits of SOA maturity models can be split up in two categories, benefits for organizations that are planning to adopt and benefits for those who are already adopting [28]. These benefits are:

- Planning benefits: offering support on four points namely defining SOA, creating a SOA vision, identifying needs and creating awareness of the business impact;
- Adoption benefits: providing guidance, showing different paths to maturity, offering key practices, supporting in defining an architecture and measuring adoption progress.

The existence of the stage maturity models for SOA adoption reflect that organizations are adopting SOA without a limited understanding of the SOA adoption process. This lack of understanding can be explained by the imitation behavior explained: they are adopting SOA because other organizations they see as more competitive adopted it. The effect is that organizations enter an uncertain path and that they do not know understand the impact with time.

The degree of SOA adoption between organizations differs. A survey of the Aberdeen Group showed that in 2005, 58% of organizations had no experience with SOA projects, 12% had conducted 1 SOA project, 16% had conducted 2 or 3 projects, and 14% had conducted more than 3 projects [13]. A survey of AMR research showed that in 2005, 21% of organizations was using SOA, 53% was considering SOA and 26% was not considering SOA. Legner and Holland conclude that in 2007 comprehensive SOA implementations continue to be scarce in practice [11]. This confirms that the adoption process is a long-term process.

3.4. Maturity models for enterprise-wide SOA adoption

In this section, four stage maturity models are analyzed and compared for their applicability in measuring maturity of enterprise-wide SOA adoption. The criteria for choosing these models are that they can measure the maturity of SOA adoption, include an organization-wide perspective and cover the technical and organizational aspects of SOA adoption. The models explained below are the stage maturity model for ERP systems use (section 3.4.1), the Service-Oriented Enterprise Maturity Framework (section 3.4.2), the Open group Service Integration Maturity Model (section 3.4.3) and the CBDI SOA maturity model (section 3.4.4). Each of these models distinguishes different stages in the path to maturity and consider different aspects of maturity.

Table 3 lists the four stage maturity models with the number of stages that they distinguish and the number of aspects considered. The questionnaires of the first and the last two models were not publicly available.

Table 3: Stage maturity models relevant for measuring enterprise-wide SOA adoption

Stage maturity model	Author	Stages	Aspects	Assessment tool
Stage maturity model for ERP systems use	Holland and Light (2001)	3	5	Questionnaire/ Semi-structured interviews (n/a)
Service Oriented Enterprise Maturity Framework	Meijborg (2006)	5	5	Questionnaire / Semi-structured interviews
The Open group Service Integration Maturity Model	The Open Group (2007)	7	7	Questionnaire (n/a)
CBDI SOA maturity model	CBDI (2007)	5	7	Questionnaire (n/a)

3.4.1. ERP stage maturity model

Holland and Light published a stage maturity model for ERP systems use in 2001 [22]. This model is relevant for our work because ERP precedes SOA as a technology for the integration of applications within an organization. Furthermore, the model includes technical and organizational aspects.

Within the ERP model, three stages were distinguished based on qualitative data:

- Stage 1: organizations are managing legacy systems and starting an enterprise-wide ERP project;
- Stage 2: implementation is complete and the functionality of the ERP system is being exploited across the organization;
- Stage 3: organizations have institutionalized the ERP system into the organization and are engaged in the process of obtaining strategic value from the system.

The first stage represents the initial phase where the organization has started implementing the ERP system. In the last phase the ERP system is fully institutionalized and strategically used in the organization. Assessments of organizations were performed by means of case studies, on base of which a maturity score was determined. This maturity score was calculated from scores on the following five maturity aspects:

- Strategic use of IT: the importance of the IT function in the business;
- Organizational sophistication: how the organizational structure has evolved as a result of the ERP system implementation.
- Penetration of the ERP system: how extensively the system is used within the organization;
- Vision: the identification of the strategic potential by management for the use of the system;
- Drivers & lessons: the drivers behind and the lessons learned during the implementation and the use of the system.

These aspects reflect the technical aspect and organizational aspects of ERP implementation. The penetration of the ERP systems is considered on the technical side, while the way in which the organizational structure has evolved is considered on the organizational side. Underlying this model is the premise that business and IT should be aligned. The aspects focus on the strategic use of the ERP system by the organization and not only on the progress of the implementation of an ERP system.

The overall maturity scores of the different organizations are shown in Figure 7.

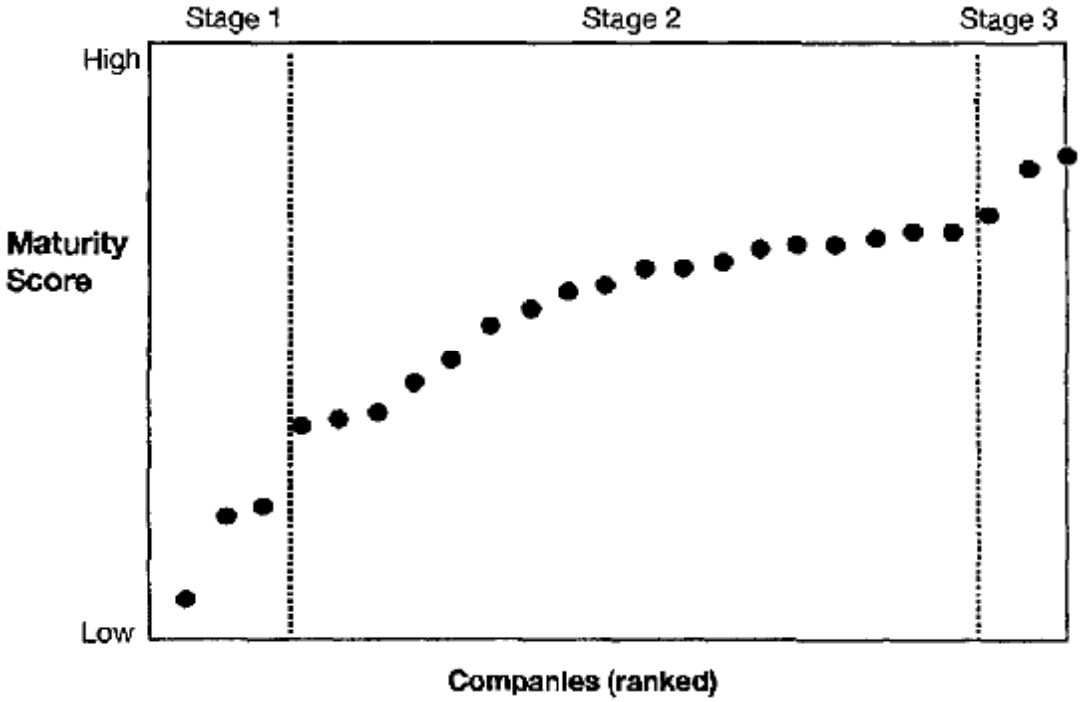

Figure 13: Maturity of ERP implementations at 24 organizations [22]

Most organizations have a relative average maturity. The horizontal axis in Figure 13 is not the time dimension as in the model of Nolan, but the 24 organizations that were assessed. Based on the results the distribution of the relative maturity *between* organizations follows the s-curve of Nolan.

3.4.2. SOE maturity framework

Meijborg developed a maturity model for the Service-Oriented Enterprise (SOE) in 2006 [19]. A SOE is defined as an organization whose business functions are implemented as composite business services. This means an organization where the concept of service-orientation is applied to the business architecture, and departments interact as service consumers and service providers. Meijborg describes this as applying SOA to the business functions. The maturity that is measured is the degree to which the organization is service-oriented following the eight service principles [4]. The most mature situation is the extended enterprise, where service-orientation and SOE are seen as opportunities for the complete value chain and ecosystem. In this maturity stage, the use of business services that use contracts between service consumers and service providers is then available for every organization working with business services and in- and outsourcing are easily applicable and used.

The SOE model is represented as matrix model in Table 4.

Table 4: SOE maturity framework [19]

	Stage 1: Some people and some projects	Stage 2: Departmental	Stage 3: Enterprise	Stage 4: Extended enterprise
Strategy & policy	Unstructured	Department related	Enterprise internal related	Internal & external related
Monitoring & control	Cost control	Financial control	Management control	Chain control
Organization & processes	Unstructured	Department related	Enterprise internal related	Internal & external related
People & culture	Personnel administration	Personnel policy	Human resource management	Human capital management
Information technology	Island automation / Isolated services	Department integration / Structured services	Enterprise integrated / Integratable services	Ecosystem integrated / Adaptive services

The maturity of the SOE adoption is split up in five aspects that are based on the Business IT alignment model [54]:

- Strategy & policy: the processes and activities that contribute to the creation and maintenance of strategy and policies;
- Monitoring & control: procedures about monitoring and control;
- Organization & processes: the way the organization handles internal processes and in how far these are documented in formal descriptions and how they fit each other;
- People & culture: how the organization handles employees and the organizational culture;
- Information technology: all information and communication facilities in the organization.

The five aspects are further broken down in a total of 37 key indicators with corresponding questions. The model mainly focuses on applying service-orientation to business functions, and creating composite business services. Although there is one aspect about Information Technology, no explicit attention is paid to application services and orchestration of business processes with application services. In the SOE maturity model, the aspects were not ranked with a number as in the ERP model but for each maturity level there is an option in the questionnaire that corresponds with one of the five maturity stages. These stages are:

- Stage 0 (No SOE): nobody is responsible for service-orientation and no vision is available for SOE;
- Stage 1 (Some people and some projects): management considers service-orientation and launches some projects to assess its potential benefits for the organization;

- Stage 2 (Departmental): some departments have made agreements between each other. The organization develops a vision about SOE.
- Stage 3 (Enterprise): SOE is realized in some departments that are centralized and business services are offered to the other departments. The whole organization works in terms of services;
- Stage 4 (Extended enterprise): service-orientation and SOE are seen as opportunities for the complete value chain and ecosystem. Cooperation and in- and outsourcing are easily applicable and used.

Underlying these stages is the assumption that the adoption of SOE follows an s-curve through the organization, starting on a small scale and followed by the majority in the organization. Validation of this model in practice was performed by assessing the maturity of two organizations in case studies.

3.4.3. The Open group Service Integration Maturity Model

The Service Integration Maturity Model (SIMM) was published by IBM in 2006 [55]. After that, it was adopted with minor changes by The Open Group as the Open group Service Integration Maturity Model (OSIMM) [26]. The Open Group is a consortium of organizations in the IT community and so the underlying focus of OSIMM is the adoption of SOA and application services for supporting business goals. More specifically, the mission of The Open Group is enabling access to integrated information within and between organizations based on open standards and global interoperability. OSIMM measures the maturity of integration with application services. The OSIMM matrix model is shown in Figure 8, with seven maturity stages on the horizontal axis and seven maturity aspects on the vertical axis. Each cell represent a maturity stage on a certain aspect.

	Silo	Integrated	Componentized	Services	Composite Services	Virtualized Services	Dynamically Re-Configurable Services
Business	Isolated Business Line Driven	Business Process Integration	Componentized Business	Componentized Business offers Services	Processes through service composition	Geo-graphical Independent Service centers	Mix and match business and context-aware capabilities
Organization	Ad hoc LOB IT Strategy &Governance	Ad hoc Enterprise IT Strategy & Governance	Common Governance processes	Emerging SOA Governance	SOA and IT Governance Alignment	SOA and IT infrastructure Governance Alignment	Governance through Policy
Methods	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling for Infra (CDSP)	Business Grammar Oriented Modeling
Applications	Modules	Objects	Components	Services	Process Integration via Services	Process Integration via Services	Dynamic Assembly; context-aware invocation
Architecture	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Re-Configurable Architecture
Information	Application Specific	LOB or Enterprise Specific	Canonical Models	Information As a Service	Enterprise Business Data Dictionary and repository	Virtualized Data Services	Semantic Data Vocabularies
Infrastructure	LOB Platform Specific	Enterprise standards	Common Reusable Infrastructure	Project-based SOA Environment	Common SOA Environment	Virtual SOA Environment; S&R	Dynamic Sense, Decide & Respond
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7

Figure 14: OSIMM matrix model [26]

The vertical axis shows the seven aspects of maturity. The Business, Organization and Methods aspects focus on the organizational side of SOA adoption. The Applications, Architecture, Information and Infrastructure aspects focus on the technical side of SOA adoption. The seven aspects can be described in the following way:

- Business: focuses around management of business architecture, and how business processes are developed and executed;
- Organization: focuses on how the organizational structure has evolved as result of the SOA adoption and on SOA governance;
- Methods: focuses on the methods and processes that are used for IT and business transformation and the software development lifecycle;
- Application: focuses on the application style and functional decomposition of application domains into application services;
- Architecture: focuses on architectural topologies, architecture decisions, and standards and policies;
- Information: focuses on information modeling including ontology, taxonomy and semantic interoperability;
- Infrastructure: focuses on the technical architecture and integration platforms.

The horizontal axis shows the stages. The path to maturity starts with three stages in which service-orientation is not started yet. In these stages the foundation for services is build by organizations by decomposing application systems into components. In the following stages application are decomposed as application services and are orchestrated in business processes. The stages or levels are described as follows:

- Service foundation levels
    - Stage 1 Silo: individual parts of the organization are developing their own software independent of each other with no integration of data, processes, standards, or technologies;
    - Stage 2 Integrated: technologies have been put in place to communicate between the silo's and to integrate the data and interconnections;
    - Stage 3 Componentized: the IT systems in the silo's have been analyzed and broken down into component parts within a framework in which they can be developed into new configurations and systems;
- Start of service-orientation:
    - Stage 4 Service: composite applications can be built from loosely coupled application services. Services may be invoked based upon open standards and independent of the underlying application technology;

- Stage 5 Composite services: it is possible to construct a business process for a set of interacting application services by using of a composition language to define the flow of information and control through the individual services;

- Stage 6 Virtualized services: business and application services are provided through a virtual interface as virtual service. The infrastructure converts the virtual invocation into a physical call of a real service with, for example, an address, a network and a protocol. The virtual service becomes more loosely coupled from the infrastructure on which it is running, permitting more opportunities for the composition of business services;

- Stage 7 Dynamically re-configurable services: the business process composition can be performed at runtime, instead of development time, under the guidance of business analysts using suitable tooling. In this situation business processes can be assembled.

The model starts in stage 1 with a situation where there is no integration between application silo's. Integration between these silo's is then achieved by decomposing the application silo's into components, and then to application services, which ultimately can be used by business analysts to orchestrate business processes at runtime. The scope of services development and use within the organization is not made explicit in the OSIMM model. The implicit idea behind the model is to measure the spread of service development and service use within the organization. This is reflected by descriptions like *individual parts of the organization are developing their own software* (description of stage 1). Spread throughout organizations is what is missing in this model. What should be included is that organizations start some pilot project to showcase success with a narrow focus, and other departments follow if they see it is successful. The model does not include the possibilities of using application services between organizations in business collaborations. A possible scenario in the situation of level 7, or maybe even in earlier stages, is that organizations invoke application services of other organizations.

3.4.4. CBDI SOA maturity model

The CBDI SOA maturity model was published in 2005 and updated in 2006 and 2007 [25, 51, 56]. It measures the maturity of enterprise-wide SOA adoption and so considers the spread of SOA throughout the organization. In the most mature stage, called ecosystem, organizations can share their application services between each other to support business collaboration.

The SOE model is represented as matrix model in Table 5.

Table 5: CBDI SOA maturity model [25]

	Stage 1: early learning	Stage 2: applied	Stage 3: integrated	Stage 4: enterprise	Stage 5: ecosystem
SOA management	Funding for pilot/PoC projects	Services are managed as an IT architecture concept	Funding systems facilitate provisioning of shares services	Services are managed as business assets	Services facilitate inter business collaborations
Service architecture	Architecture is fragmentary & experimental	Project architectures are service-oriented	There is a standard for rich service specification	The enterprise has a service portfolio plan	There are agreed business process and data architectures for business collaborations
Operational infrastructure	ESB pilot or PoC	Project ESB	Common ESB framework	Common framework for enterprise service management and security	Services are managed as federated resources
Life cycle infrastructure	Services are not managed assets	Services are project level deliverables	Enterprise level registry and repository provides consistent life cycle governance of the run time service asset	Enterprise registry and repository provides design and runtime service asset life cycle governance and asset dependency horizon analysis	Ecosystem registries provide governance over collaborative business processes
Framework and processes	Frameworks and practices extended in ad hoc manner	Project specific architecture frameworks	Common SOA reference architecture and process	Convergence of business and IT practices around service concept	Collaboration reference architecture
Organization	IT architects sponsor SOA	SOA is a project level responsibility	There is a single point of accountability for integration	Services are owned by the business	Services defined and managed on inter business collaborative basis
Project & programs	Pilot and PoC projects	Service delivery and usage integrated into LoB projects	Specialization of service provisioning, implementation and assembly projects	Service product management	Collaborating parties act as provider and consumer

The maturity stages are described as follows:

- Stage 1 (Early learning): the organization is experimenting with SOA. The SOA initiative is likely to be characterized as pilot project or proof of concept projects;
- Stage 2 (Applied): SOA is employed within some business critical projects to deliver improved structure;
- Stage 3 (Integrated): SOA is used to deliver integration between projects and or application silo's;
- Stage 4 (Enterprise): SOA is institutionalized at the enterprise level;
- Stage 5 (Ecosystem): SOA is inherently federated supporting virtual business.

The stages reflect the spread of SOA and services throughout the organization. It starts in stage 1 as pilot projects, and in stage 5 application services are institutionalized on inter-enterprise level. In contrast to the SOE maturity model and OSIMM, there is no initial stage in which SOA is not applied. The different aspects of maturity reflect the technical and organizational side of SOA adoption. The aspects are:

- Management: the focal point for management capabilities spanning visioning, strategy, funding, chartering, governance, measurement and management of the SOA adoption process;
- Service architecture: the creation and maintenance of the service and associated architecture;
- Operational infrastructure: the architecture and capabilities to support the run time service environment.
- Life cycle infrastructure: the architecture and capabilities to support the entire life cycle of service states, ranging from 'planned' to 'retired' and 'archived';
- Framework and process: the reference architecture framework detailing the layering, policies, patterns, models, deliverables;
- Organization: The roles and responsibilities required to establish, operate and maintain a service-oriented business;
- Project & programs: the project capabilities necessary to plan, provision, implement and assemble services.

These aspects also cover a broad perspective on SOA adoption. Less emphasis is given in this model to the orchestration of business processes based on application services.

3.4.5. Comparison

Before discussing the differences of the four models, first the similarities are identified. All four maturity models consider the three key concepts of stage maturity models namely stages, maturity aspects and a maturity assessment tool. For OSIMM and the CBDI model the assessment tools were not publicly available. Except for the ERP model, the maturity models were represented by a matrix model.

An important difference between the models is that the ERP stage maturity model is based on empirical data, while the three SOA maturity models mainly reflect expert opinions of a mature situation of SOA adoption. This is because SOA is still a young technology and comprehensive SOA implementations are still scarce [11]. The ERP stage maturity model also focuses on the use and benefits of the ERP system, not only on the progress of the implementation of the ERP system. The SOA maturity models focus only on the progress of the SOA adoption.

Furthermore the regarded maturity differs in these models. The SOE maturity model and the CBDI SOA maturity model focus respectively on the scope of business services and application services use inside the organization and between organizations. The scope starts on a narrow scale and increases to enterprise-wide and inter-organizational use. OSIMM focuses on the maturity of integration with application services in the context of organizations, but does not address the increase of scope. The use of application services for business collaboration is not made explicit in this model. The stages of the ERP stages maturity model reflect a more realistic view of use. When applying the ERP model to SOA, stage 2 should then be renamed to *SOA adoption is complete and the functionality of the SOA is being exploited across the organization*. Organizations have not progressed that far with SOA adoption and so most organization would be positioned in stage 1 [11, 13].

The starting point of the maturity models also differ. OSIMM starts with application silo's which are not integrated using services, and in the first three stages of this model service-orientation has not been started. The SOE maturity model calls the first stage *no SOE*. In the ERP model and the CBDI model SOA adoption has already been started in the first stage.

Considering the applicability of the models to measure SOA adoption, the ERP stages maturity model is an empirical model but the stages do not reflect the situation for SOA adoption in the current situation. Further, it does not include technical aspects relevant for SOA, like how the application architecture changed, simply because it is a stage maturity model for ERP systems and not for SOA.

The SOE maturity model mainly focuses on applying service-orientation to business services. No explicit attention is paid to the development and use of application services. Although it misses the technical aspects of SOA with application services, key indicators within the organizational aspects are relevant and can be translated to fit these aspects.

The OSIMM model and the CBDI model both have applicable key indicators, and their maturity aspects cover the technical and organizational side of SOA adoption. They both focus on application integration with the use of application services, but their perspective of maturity differs. Two maturity aspects of the ERP model are covered by the aspects of the OSIMM and CBDI model. The Organizational sophistication aspect, which captures the way in which the organizational structure has evolved, is covered by the aspects they both call 'organization'. The *penetration of the system* is covered by most aspects of the CBDI model, because the aspects consider, for example, the spread of

an Enterprise Service Bus (ESB) and the spread of the service architecture throughout the organization.

The stages in the OSIMM model reflect the focus on services. In the mature situation, services are virtualized and can easily be used in different business processes. The scope of services use within the organization is not considered explicitly. At level 5 of the OSIMM model business processes can be assembled from loosely coupled services at design time, which is necessary to support shared services at enterprise level and so that services can be used organization-wide. The stages in the CBDI level explicitly focus on the scope of services use within the organization. Services are first used on project level and are then centralized into shared services at the enterprise-level, where institutionalization takes place. This better reflects the s-curve of technology adoption.

3.5. Conclusions

Adopting an enterprise-wide Service-Oriented Architecture involves the translation of the SOA principles into concrete architectures and implementations that fit in with the existing applications. This translation requires that choices are made about which standards, technology and software to use to implement SOA. However, SOA adoption is not only a technical implementation. It requires that the importance of organizational aspects is acknowledged and managed by the stakeholders that are involved in the SOA initiative. Organizations should gradually adopt an enterprise-wide SOA to minimize the risk of failure and to gain management commitment.

Stage maturity models can support the SOA adoption process by functioning as a roadmap. Several stage maturity models exist for enterprise-wide SOA adoption but these models have different lacks. Firstly, they are not technology independent because they are created by software vendors or consulting firms. SOA is a technology-agnostic architectural style and so a stage maturity tool must be independent of software products that implement SOA.

Secondly, current models lack attention for management involvement and other organizational issues. SOA adoption is a complex and long term path, and understanding and commitment of the management is essential to facilitate the adoption throughout the organization.

Thirdly, current models lack an assessment tool. Using an assessment tool, an organization can assess the current and desired maturity stage and derive an organizational roadmap and develop an improvement plan.

Although the four stage maturity models that were considered contain lacks, their elements can be used as a basis to construct a stage maturity model for the adoption of an enterprise-wide SOA. Reviews of these models with experts are needed, combined with evaluation of adoption experiences in practice, to identify the most important maturity aspects and stages.

4. Stage Maturity Model for enterprise-wide SOA adoption (SMM-SOA)

This chapter present the Stage Maturity Model for the adoption of an enterprise-wide SOA (SMM-SOA) we propose. Section 4.1 presents SMM-SOA as a matrix model. Section 4.2 discusses the maturity stages. Section 4.3 discusses the maturity aspects. Section 4.4 discusses the maturity assessment tool. Section 4.5 presents the conclusions of this chapter.

4.1. Matrix model

The matrix model is shown in Table 6. The stages ranging from initial to full development are shown on the horizontal axis, and the aspects of maturity are shown on the vertical axis.

Table 6: Matrix of the maturity model for SOA adoption

	Siloed	Experimental	Applied	Integrated	Institutionalized	Networked
Strategy & governance	Functional strategy	Ad hoc SOA strategy	Best practice SOA strategy	Formalized SOA strategy	Enterprise SOA strategy	Network SOA strategy
Organizational change	Silo's owned by functional IT teams	Services owned by architects	Services owned by project teams	Central services owned by central IT department	Enterprise services owned by business units	Network services owned by business units
Business architecture	Process automated with components	Ad hoc process orchestration	Best practice process orchestration	Formalized process orchestration	Process orchestration with enterprise services	Process orchestration with network services
Information model	Domain specific information models	Ad hoc domain-crossing information model	Best practice domain crossing information models	Common information model	Enterprise information model	Network information model
Application architecture	Componentized application silo's	Experimental services	Best practice services	Formalized services	Enterprise services	Network services
Operational infrastructure	No messaging framework	Experimental messaging framework	Project related messaging framework	Common messaging framework	Enterprise messaging framework	Network messaging framework

The six stages are mainly based on the stages in the CBDI model [25]. The stages of the CBDI model best reflect the cumulative adoption in an s-curve within an organization, and stages were distinguished based on this s-curve that has been observed in earlier research [19]. The early adopters within the SOA context are IT architects experimenting with SOA in a pilot project. SOA is then extended to be applied in business critical project and then eventually it is applied within the majority of departments of the organization. The first stage is added to the model to describe the situation with applications that have not been decomposed into application services. This stage reflects a situation where integration between application silo's is achieved with components but not with application services. Service-orientation has not started in this stage, therefore this stage has the number 0. The other stages are adopted from the CBDI SOA maturity model. Early learning (stage 1) is renamed to 'experimental'. Enterprise (stage 4) is renamed to 'institutionalized'. Ecosystem (stage 5) is renamed to 'networked'.

The six aspects were based on a review of the four current stage maturity models explained in Section 3.4, and reviewing these models with experts. Experts were asked to judge the importance of each aspect in an initial list, which was derived from the four models. Experts were also asked if aspects should be removed from the initial list, and if new aspects, which they considered were missing in the list, should be added. By interviewing experts from business and academia, multiple perspectives were taken into account. Social networking was used to sample the experts, so that experts were identified by people who know people who know experts in relation to SOA adoption [29]. This research was primarily conducted at a large IT consultancy organization in the Netherlands. In this way it was easy to access and interview experts directly or by phone. Experts were interviewed one-on-one, but also an expert session with different experts was conducted to enable the complementation of aspects of SOA adoption. Experts were also identified from publications in the business and scientific field, representing different perspectives towards SOA adoption. For example, a SOA evangelist was interviewed that had created a SOA maturity model, but also a scientific researcher in the field of ERP implementation was interviewed. The output of this process was a list of six aspects, which reflect literature and expert opinions about enterprise-wide SOA adoption.

The six aspects were further subdivided in a total of 31 key indicators. A similar process was used to subdivide the aspects into key indicators: an initial list of key indicators was identified from the four stage maturity models explained in Section 3.4, and the initial list was reviewed and complemented by various experts. The key indicators each have an equal weight with their aspect, due to time restrictions adding weighing factors to the key indicators was beyond the scope of this research.

4.2. Stages in SOA adoption

The process of adopting SOA has been split up into six stages. The stages are explained below, starting with the initial adoption stage and ending with the network stage.

4.2.1. Stage 0: Siloed

In this stage the organization may have considered SOA adoption, but has not started any project related to it. The application systems can be qualified as application silo's, which are domain-specific application systems in which no open standards are used for integration [26]. Funding is based on application-specific budgets. A start may have been made with decomposing the applications into component parts that can be used to build new composite applications, but these are not loosely coupled and cannot be used as flexible building blocks [39]. Adapters and other technologies may be in use to support communication between the different applications, but still strong dependencies exist between the application silo's. Information models are domain-specific. For example, the definition of a customer may differ within the different applications. As a consequence information about a customer must be translated to be used in another application, and conflicting information may be stored in different applications. Business processes may be automated based on componentized applications, but applying changes to the business processes is difficult because components are rigid. The business and IT strategy do not suport the use of business and application services.

4.2.2. Stage 1: Experimental

In the experimental stage, some IT architects are starting to experiment with the development of application services and orchestration of business processes. Some early SOA project(s) have started that can be characterized as pilot or proof of concept projects, and funding is given to these projects [19]. Because of the hype around SOA, some organizations may prefer not to label it a "SOA" project although principles of service-orientation are applied. The projects are used to showcase SOA and therefore the risk and impact of failure is brought to a minimum, often by developing application services within a greenfield situation with little complexity. Within the projects an experimental messaging framework is typically tested to support messaging between services. Information about how to develop and use services, and the experience that is gathered within projects are shared in an ad hoc way by informal contacts or presentations. A start is made with a domain-crossing information model, although with a narrow scope. The business and/or IT strategy address the potential of decomposing business and application logic into services.

4.2.3. Stage 2: Applied

After the experience in SOA pilot projects, in the Applied stage SOA is applied to business critical projects with more application legacy and complexity involved. The responsibility for development of application services lies at project-level. Integration with application services is achieved between the applications that have been decomposed into application services. One or more service messaging frameworks are used to facilitate communication between application services. Business processes are orchestrated based on the application services within a project scope. Knowledge and experience that is gathered is shared between the running projects as guidelines and best practices. Some standards arise for service description and message formats. Agreements are made about the development and use of service, such as agreements about the volume in which an application service can be used by different parties. The information model covers the scope of the different projects. The business strategy and/or IT strategy address the use of services and service composition by specific parts of the organization.

4.2.4. Stage 3: Integrated

In this stage, SOA is integrated within IT and the development and use of application services is formalized as policies. In most parts of the organization applications have been decomposed as application services and by using a common service messaging framework, services can be used throughout the organization. Service discovery is facilitated by a service registry that covers the application services throughout the organization. Business processes are orchestrated based on the application services shared in the organization. Application services are owned by central IT departments of business units. SOA training is also offered at business units at enterprise level. There is a funding system that supports the development of application services, and service consumers are billed for the use of services. There is a common information model specifying the business objects and open standards in the organization. The business and IT strategy address the development and use of services for most parts of the organization.

4.2.5. Stage 4: Institutionalized

In this stage, SOA has been institutionalized within the organization and is the standard for application development. Application silo's have been broken down into application services that are used as basis to orchestrate business processes. This is facilitated by an enterprise-wide messaging framework. Application services are owned by business units, and service consumers are billed for the use of application services. Open uniform standards are used for message formats, semantic interoperability and service description. SOA training is fully integrated into IT training. Business and application services are fully institutionalized within business and IT strategy.

4.2.6. Stage 5: Networked

In this stage SOA, is not only applied within the organization but also institutionalized for long-term and short-term business collaborations. Orchestration of business processes is not only restricted to partners, but possible for every organization that works with application services [19]. Application services, owned by the business units, are also offered to other organizations. In- and outsourcing are easily applicable based on the use of uniform standards on an inter-organizational level. There is an inter-organizational service registry to discover services and use them.

4.3. Maturity aspects of SOA adoption

Six aspects of an organization are considered in order to measure the maturity of enterprise-wide SOA adoption. These aspects represent significant views of the business and IT environment of the organization that evolve as a result of the SOA adoption. The first three aspects represent the organizational aspects, namely 'strategy & governance', 'organizational change' and 'business architecture'. The last three represent the technical aspects, namely 'information model', 'application architecture' and 'infrastructure'. Each aspect can be seen as a category, which is further broken down in key indicators. The aspects and their key indicators are described below.

4.3.1. Strategy & governance

This aspect describes how the top level and management is involved in the SOA adoption, how the business and IT strategy address business and application services respectively, how SOA is governed, and experience with outsourcing. The stages for this aspect are shown in Table 7.

Table 7: Stages for the 'strategy & governance' aspect

Silned	Experimental	Applied	Integrated	Institutionalized	Networked
Functional strategy	Ad hoc SOA strategy	Best practice SOA strategy	Formalized SOA strategy	Enterprise SOA strategy	Network SOA strategy

This aspect is subdivided into 5 key indicators: commitment from top level, business services addressed in business strategy, application services addressed in IT strategy, SOA governance & policies, and status of outsourcing.

Commitment from top level

Commitment from the top level of the organization is essential because SOA adoption is a long and complex process and has a major influence on business processes and applications [47]. The effects of SOA do not show on the short term, because SOA is a new way of working and people have to get accustomed to it. Without understanding of the impact of SOA and commitment from top level to

the SOA initiative, investment may stop because no benefits are shown on the short term [6, 17]. Although financial investments are important, top level can also support in other ways, for example, by supporting communication about SOA in the organization, and in this way creating awareness and understanding [57]. Awareness and understanding of employees is an important factor for the acceptance of information technology [53].

Business services addressed in business strategy

Interviews with experts showed that SOA and application services are mainly seen as an 'IT thing'. However, service-orientation can also be applied to the business, since the organization can transform to a service-oriented enterprise where business logic is decomposed into business services that can be reused within different business processes [4, 19]. If the business strategy addresses business services, a service-oriented business architecture is enabled. This also indicated that top level is thinking in terms of services and that the top level has more understanding of application services and business process orchestration.

Application services addressed in IT strategy

Within a SOA, application systems are decomposed into application services [4]. If the IT strategy addresses application services, this enables the transformation from application silo's to loosely coupled application services by the several IT functions within the organization. For those that are unaware of these application services, awareness and understanding of SOA can also be created.

SOA governance & policies

In the initial stages of SOA adoption, problems related to application services can be solved ad hoc, but as the adoption progresses, SOA governance becomes critical [49]. SOA governance can be described as the management control of application services in a SOA. As the scope of SOA within the organization grows, SOA governance must be formalized in terms of policies.

Status of outsourcing

The existence of outsourcing contracts for business processes is an indication that the organization is differentiating core and supportive processes [19]. This means that business processes in the organization are well described. The organization is then applying the virtual organization model, such that supportive processes are outsourced to partners [39]. SOA is an enabler of the virtual organization model by means of application services and business process orchestration.

4.3.2. Organizational change

This aspect describes the way in which the organization has evolved as result of the SOA adoption. It is essential that organizational systems and processes change in according to the management of application services [57]. The stages for this aspect are shown in Table 8.

Table 8: Stages for the 'organizational change' aspect

	Siloed	Experimental	Applied	Integrated	Institutionalized	Networked
Organizational change	Silo's owned by functional IT teams	Services owned by architects	Services owned by project teams	Central services owned by central IT department	Enterprise services owned by business units	Network services owned by business units

This aspect is subdivided into 9 key indicators: funding of application services, billing of application services, ownership of application services, scope of SOA program, SOA vision, belief in SOA effectiveness, SOA awareness, SOA training, and SOA skills.

Funding of application services

Funding of application services poses a challenge for organizations [14]. Whereas in the initial stage budgets are specific for application silo's, now the development and use application services must be funded. The volume of use of application services may differ, and as a consequence, the exploitation and maintenance costs may rise. A funding model is needed to support application services development, deployment and maintenance [25].

Billing of application services

The use of application services creates a service market with service providers and service consumers. A billing model should be developed that registers the use of application services and bills the volume of use to service consumers, to cover the costs of development and exploitation [25]. An organization may bill service consumers by using taxes within the IT departments.

Ownership of application services

When merging from application silo's to application services, the question is who should be the owner of the services. Owners are responsible for gathering requirements, development, deployment and operations management of a service [14]. When the SOA scope grows, it is essential that the ownership and the related responsibilities are well documented within the organization to support a proper collaboration between the services [49]. This is emphasized by the fact that within a SOA, application services are autonomous within a certain logic domain, and other parties in the organization depend on the development and availability of the service.

Scope of SOA program

The scope of the SOA program or projects is an indication of the acceptance of SOA within the organization. SOA is first applied in pilot projects with a greenfield situation, but the scope is then extended to business critical projects. After gaining experience, SOA is formalized as a standard within business units and the enterprise, and eventually becomes the standard for business collaboration.

SOA vision

SOA adoption has a major influence on the organization, and changes have to be made [15]. By having a SOA vision on the future situation and showing the consequences of the current situation, commitment can be obtained throughout the organization [58]. Organizations struggle with the long-term path of adoption and must not only focus on the fulfillment of short-term requirements within projects. A shared SOA vision creates commitment for a long-term plan and awareness about the boundaries of projects.

Belief in SOA effectiveness

The degree in which people believe SOA is beneficial within their jobs, is an important factor in their acceptance and thus their use [53]. The perceived effectiveness is based on their experience and social influence. The belief in the effectiveness of SOA within the organization is an important indication for the stage of SOA adoption.

SOA awareness

SOA awareness within the organization is also important. Are only some people aware of SOA, or is SOA a well-known standard for application development within the organization? Communicating the strategy and vision can support the SOA awareness about in application and business services.

SOA training

The way in which knowledge and experience about SOA is transferred within the organization influences the effectiveness of integration with application services and business process orchestration. Employees should have the proper knowledge and skills to perform their job with SOA [59]. When experimenting in pilot projects knowledge and experiences is transferred by informal contact in the form of best practices. However, as SOA is formalized as a standard for application development, SOA training must be given to transfer the different knowledge and skills needed to work with a SOA.

SOA skills

Different skills are needed to develop and maintain a SOA. Examples are the skills to support development of application services, operational use of application services and orchestration of business processes. Identifying the expertise needed within an organization and developing this expertise is essential for successful SOA adoption.

4.3.3. Business architecture

This aspect describes how SOA is applied to the business architecture. This involves the decomposition of business logic into business services, and the description of business processes with business services, and orchestration with application services. The stages for this aspect are shown in Table 9.

Table 9: Stages for the 'business architecture' aspect

Siloed	Experimental	Applied	Integrated	Institutionalized	Networked
Process automated with components	Ad hoc process orchestration	Best practice process orchestration	Formalized process orchestration	Process orchestration with enterprise services	Process orchestration with network services

This aspect is subdivided into 4 key indicators: decomposition into business services, description of business processes in business services, business process orchestration, and business & IT cooperation.

Decomposition into business services

Applying service-orientation to the business architecture can be done by breaking up the business processes in elementary business activities, and supported by them by business services [4]. These business services have a well-described inputs and outputs, and can be reused in different business processes as flexible building blocks [19]. Distinguishing business services within the organization enables the implementation of application services that implement the functionality described by these business services.

Description of business processes in business services

Distinguishing business services within the organization offers the possibility to compose and describe business processes [19]. Business processes are then flexible chains composed of business services that function as building blocks. When needed, the chain can be easily reconfigured to adapt to changing needs. Different but similar business processes can more easily be consolidated because they are based on business services. This is most relevant when lots of different business processes exist and their volume of use is low. Similar business processes can be configured easier in a process template, and in this way creating more efficient use of resources. Business process descriptions based on business services enable business process orchestration based on application services.

Business process orchestration

Business processes can be made executable if application services are orchestrated. An orchestration defines the order in which services are to be invoked and how different services are composed into an executable business process [2]. Business processes can be implemented by orchestration internal services but also services from business partners.

Business & IT cooperation

Orchestrating business processes based on application services requires a strong cooperation between business & IT within projects [49]. Expertise often focuses on either business process management or application development based on SOA. These two expertise fields have to be integrated in order to support orchestration.

4.3.4. Information model

This aspect describes how information exchange by application services is standardized. This involves standards for message formats, semantic interoperability, service description, a service registry and service discovery. The stages for this aspect are shown in Table 10.

Table 10: Stages for the 'information model' aspect

Silned	Experimental	Applied	Integrated	Institutionalized	Networked
Domain specific information models	Ad hoc domain-crossing information model	Best practice domain crossing information models	Common information model	Enterprise information model	Network information model

This aspect is subdivided into 6 key indicators: standardization of message formats, standards for semantic interoperability, service description, storage of service descriptions, service discovery, and governance of standards.

Standardization of message formats

SOA is enabled by the use of open and accepted standards for communication between Web Services [4]. Reachability is an important requirement for interacting with application services [35]. This is the main difference between SOA and application integration based on components, where different techniques and message formats are used for communication between application systems [39]. Using open standards for message exchange between services, eases the integration of heterogeneous applications.

Standardization of semantic interoperability

Information has not only to be exchanged in a standard way, but must also be interpreted by other applications services on a standard way. A SOA requires clear semantics, especially when traditional application domain boundaries are crossed [35]. To support semantic interoperability between application services standards are required, like ontology's and taxonomies. In the network stage, the need for standards is critical, because the message exchange between application services then crosses organizational boundaries.

Service description

A service description represents the information needed in order to use an application service [35]. Because services act as black boxes, a clear definition must be given on what the inputs and outputs are and at what location services can be reached [3]. A service description works as a manual for using the service, without giving information about the internal logic. Standards must be used for service description to facilitate interaction.

Storage of service descriptions

Visibility of application services is a key element of a SOA [35]. To be able to find services, service descriptions have to be stored in a service registry. In the experimental stage this is often done by storing service description in a spreadsheet. As adoption progresses, storage of service descriptions must be formalized in service registries with well-defined ontology's and taxonomies.

Service discovery

Storage of service descriptions enables service discovery. The service registry functions as a phone book, by listing the available services and offering service descriptions that contain information on how to interact with services [2]. In the experimental stage, service discovery is mainly done by informal contact or searching in a spreadsheet. In the network stage, an inter-organizational service registry is needed to facilitate discovery and selection of application services of business partners.

Governance of standards

Governance of the different standards related to the implementation of application services is important to support information integration between application domains. Information policies must be enforced to make sure the same standards are used. As the scope of SOA grows, standards are institutionalized at enterprise level to support interaction [35].

4.3.5. Application architecture

This aspect concerns how service-orientation is applied to the application architecture. This involves development of application services, a reference architecture, service lifecycle management and architectural decisions. The stages for this aspect are shown in Table 11.

Table 11: Stages for the 'application architecture' aspect

Siloed	Experimental	Applied	Integrated	Institutionalized	Networked
Componentized application silo's	Experimental services	Best practice services	Formalized services	Enterprise services	Network services

This aspect is subdivided into 4 key indicators: scope of application services development, SOA reference architecture, service lifecycle management, and architectural decision making.

Scope of application services development

When applying SOA to the application architecture, application systems are decomposed into application services [4]. In the experimental stage, application services are developed and used in a pilot project with a narrow scope. When the scope is extended, best practices about service development are formalized, and application services are offered throughout the organization [25]. In the network stage, application services can also be used by other organizations in business collaborations [19].

SOA reference architecture

A SOA reference architecture is a generic design template to guide the development of project-related SOA architectures and application services [35]. It can also be used as a monitoring tool to see if proper services are developed following standards. In the experimental stage, guidance is provided by best practices. Based on the formalization of best practices, a SOA reference architecture emerges to provide guidance for service development throughout the organization.

Service lifecycle management

Managing the service lifecycle of application services is important to implement application services throughout the organization [49]. Policies and guidelines have to be formalized to support the different states of application services. For example, version management is an issue for organizations. When should a new version of a service be developed? Should service consumers be forced to change to a new version?

Architectural decision making

Architectural decisions have to be made to implement SOA architectures. Project architectures that apply SOA must be reviewed for their accordance with the SOA reference architecture. However, specific implementation choices have to be made. For example, what if an architect wants to deviate from standards described in the SOA reference architecture? The formalization into policies to support architectural decision are an indication for the progress of SOA adoption.

4.3.6. Operational infrastructure

This aspect concerns the technical infrastructure that supports the runtime service environment. This involves a messaging framework for application services and secure messaging. The infrastructure or integration platform is often referred to as the Enterprise Service Bus (ESB). The stages for this aspect are shown in the white cells in Table 12.

Table 12: Stages for the 'operational infrastructure' aspect

Silned	Experimental	Applied	Integrated	Institutionalized	Networked
No messaging framework	Experimental messaging framework	Project related messaging framework	Common messaging framework	Enterprise messaging framework	Network messaging framework

This aspect is subdivided into 3 key indicators: scope of the service messaging framework, security and, support for Service Level Agreements.

Scope of service messaging framework

It is important that the operational infrastructure provides a standardized messaging framework and offers open standards for interaction between services. When application services are developed, the framework can be used to facilitate the implementation of messaging between the services [2]. In the experimental stage, the service messaging framework has a narrow project-scope. When application services are offered throughout the organization, an organization-wide service messaging framework is needed to support this interaction.

Security

Messaging between application services poses a challenge to organizations with respect to security [2]. Application services are loosely coupled and can easily be reused within different business processes, and so security can be an issue. While standardized messages can be forwarded by intermediaries, end-to-end point security must be supported. Standards for service messaging security implement integrity and confidentiality. Security must be integrated in the standard service messaging framework to facilitate secure interaction of loosely coupled services.

Support for Service Level Agreements

The use of application services to integrate applications across application domains and organizational boundaries, has resulted in a new set of management challenges [2]. These are caused by limited visibility and control over application services that are outside the control of an organization. Non-functional requirements like performance, availability and security are important, especially when orchestrating business processes based on applications services. As business processes are implemented as chains, an application service that is not available becomes the weakest link. Service level agreements (SLAs) must be defined and monitored to support control over the quality of service of the operational infrastructure.

4.4. Maturity assessment tool

The maturity assessment tool of our model consists of a questionnaire, that was constructed based on the 6 stages and the 6 maturity aspects. The questionnaire can be used to assess the maturity of enterprise-wide SOA adoption of an organization. The questionnaire can be found in Appendix A. The questionnaire is structured as follows:

- Introduction to the questionnaire that explains SOA and decomposition to services;
- General questions about the organization and the respondent;
- 31 multiple choice questions corresponding to the key indicators in the stage maturity model. These questions are grouped according to the 6 aspects. For every aspect an introduction is given;
- General comments.

The 31 questions represent the 31 key indicators described in section 4.4. For each key indicator, 6 scenario's have been described that match the 6 stages. The respondent is asked to choose the scenario that best fits the current state of the organization. An extra option "no opinion" was added to each question, to cover the possibility that the respondent does not know an answer to the question or that the respondent does not want to anser.

Question 1 is given as an example:

1. **How would you describe the commitment to SOA at the top level of your organization?**
- The top level is unaware of SOA or does not see it currently as an opportunity for the organization;
- The top level may know about SOA pilot projects that are running with SOA and/or give the pilot projects support;
- The use of SOA in some business critical projects is supported by the top level and seen as business opportunity;
- The top level identifies SOA as critical for the organization and supports the use of SOA throughout the organization;
- The top level fully supports the use of SOA throughout the organization;
- The top level fully supports the use of SOA inside the organization and in business collaboration;
- No opinion.

The first six options represent the stages from siloed (stage 0) until networked (stage 5). The questionnaire in Appendix A gives the options for each question in accordance with the stages. In the questionnaire that was sent to the respondents within the case studies, the answers were put in a random order, to avoid that the respondent influenced the answers.

4.5. Conclusions

Based on a review of SOA literature, semi-structured interviews with architecture experts and an expert group session, the Stage Maturity Model for enterprise-wide SOA adoption (SMM-SOA) was constructed with six stages and six maturity aspects (see Table 6). The model was validated by interviewing experts. The maturity aspects were subdivided into a total of 31 key indicators. A questionnaire was constructed with 31 questions that correspond to the 31 key indicators. This questionnaire functions as a maturity assessment tool for the stage maturity model and has been used in the case studies reported in Chapter 5.

5. Three case studies

This chapter reports the analyses of the three case studies that were conducted. Section 5.1 discusses the research method that was used to conduct the multiple-case study. Section 5.2, 5.3 and 5.4 report the within-case analysis for respectively Case Alpha, Beta, and Gamma. Section 5.5 presents the conclusions of this chapter.

5.1. Research method

A multiple-case study was conducted to assess the maturity of three organizations involved in a SOA adoption program. The goal of the multiple-case study was to test the constructed stage maturity model for different types of organizations. Conducting a multiple-case study was an appropriate research method, because SOA is a broad and complex phenomenon, the existing body of knowledge was insufficient to permit posing causal questions, an in-depth investigation was needed, and enterprise-wide SOA adoption cannot be studied outside the context in which it occurs [29, 31]. A case study is defined as an empirical enquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident [60]. The latter is true for enterprise-wide SOA adoption by organizations. SOA adoption is an abstract concept applied to the organization, but it influences the way in which the organization interacts with the environment (context) and it may even change organizational boundaries. Because of the complexity of enterprise-wide SOA adoption and the contextual environment, each case study included a semi-structured interview in combination with the maturity questionnaire. By transcribing and coding the interviews, the qualitative data was related to the stages, aspects and key indicators of SMM-SOA. To code the transcriptions, the software program NVivo was used [30]. Each case study involved a preparation with documentation, which included general information about the organization, and information about how the organization had adopted SOA in the business and/or application architecture.

After developing the case study design, the next step was to make a sample of organizations and to select cases [60]. The sample was based on two sources: we used the social networking strategy to identify cases of interest from people who know people who know what cases are rich in information [29]. This master thesis project was primarily conducted at a large IT consultancy organization in the Netherlands. In this way it was easy to access and interview people, which were involved in different IT implementations at larger organizations. These people were mainly IT architects, but they included also more sales-oriented roles such as account managers, and thus they had a variety of roles in relation to IT and SOA adoption programs. A variety of people were contacted directly, by phone or by email, and asked about running SOA adoption programs at client organizations. This often led to references to other architects. The other source we used was business literature about SOA adoption. Organizations which were adopting SOA, were derived from magazines and conference websites.

The intensity sampling strategy was used to select cases that manifested the phenomenon intensely and were rich in information [29]. SOA adoption is a new phenomenon, and so it would be useless to select organizations that did not adopt SOA in any way. Three cases were selected for the multi-case study, referred to as Case Alpha, Case Beta and Case Gamma. These cases reflect different type of industries: the public sector, the financial sector and the defense sector respectively.

After selecting the organizations, informants were initially sampled based on references and names which were derived from business literature. Using these names, social networking was used to identify other informants within the organizations. A fundamental criterion for selecting informants was that they had sufficient knowledge about SOA and the SOA adoption program in their organization. In each case, one informant was selected. All informants could be qualified as SOA evangelists, and they all had a leading role in the SOA adoption program in their organization. Intake interviews were conducted by phone or directly, to test the suitability of the informants and the SOA adoption program.

The results of the individual case studies are described below. The cross-case analysis is reported in Chapter 6.

5.2. Case Alpha

The first case is a municipality in the Netherlands, which we refer to as Alpha. Alpha can be qualified as a middle-sized municipality. In 2005, a program was initiated by the local public office, with the goal to improve the public service to citizens by centralizing the municipal data, and creating a digital entrance for the citizens and parties of interest. This program contained different projects, of which one focused on integration is the SOA project. This project is aimed at centralizing the municipal data, and it is focused on decomposing the application architecture. Another project related to SOA is a project focused on process description, aimed at describing generic process templates to support process reuse. The manager of the SOA program was interested in the possibilities of SOA, and so in 2005 a research was conducted by a third party to assess the consequences of SOA adoption in the organization.

5.2.1. SOA adoption program

Alpha started to adopt SOA driven by three issues. One issue is related to the volume in which business processes are used. The organization has many business processes to handle a variety of citizen requests, for example a request to cut a tree, but also a request for a burial. The diversity of these business processes is high and their individual use is low: a total of 1.200 citizen requests per year is above average. Decomposition of the business architecture into business services is performed in the project 'process description'. Business processes are decomposed to business services, so that these processes can be standardized in terms of uniform business processes. These uniform business processes can then be reused to execute the diversity of business processes, leading to an increased use of the capacity.

Another issue in Alpha is that data is stored differently in each application. An example is the format in which street names are formatted and stored, this is done in a variety of ways. The goal is to decouple data like street names from the different applications by using application services. SOA is applied here as an information integration solution.

The last issue concerns the high number of applications. The organization has around 300 applications of which the capacity use is considered too low. The goal is to apply SOA to decompose the application systems into application services, and to enable reuse of these services. The applications are decomposed into functional components that offer a certain functionality. Although the future vision focuses on decoupling applications into loosely coupled application services, the current state is that these applications are application components. With regard to the application architecture, the organization chose to outsource the development of these functional components

to smaller suppliers. To increase bargaining power in the outsourcing of applications, Alpha decided to form a consortium with other municipalities. This was enabled by the open relation between municipalities, since they are not competing within a market. Although the components currently used are not loosely coupled, open standards are used for messaging between the components. Standardization of open standards for messaging is promoted and enforced by a national reference architecture that applies the SOA concepts.

5.2.2. Results of the maturity assessment

The respondent of the questionnaire was the manager of the SOA program. The semi-structured interview was conducted with this same person. The results to the questionnaire are shown in Figure 15.

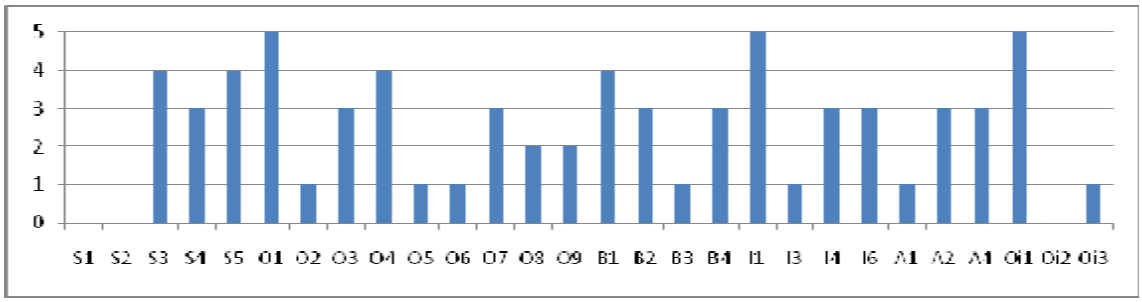

Figure 15: Results of the maturity assessment with Alpha

The key indicators are ordered according to the key indicators of SMM-SOA which are described in Chapter 4. For example, S1 denotes the first question on the 'strategy & governance' aspect, while O7 denotes the seventh question of the 'organizational change' aspect. The respondent did not answer three questions, these are not shown in Figure 9. The horizontal axis shows the questions, while the vertical axis represents the assessed maturity stage for each key indicator. As can be seen, the results on the key indicators range from stage 0 till stage 5. These extreme answers can be explained based on the semi-structured interview that was conducted. For example, key indicator O4 is in maturity stage 4 because the municipality is middle-sized and the number of IT employees is 8, so the scope of projects then more easily covers the whole organization. This is also forced by small budgets, so that a new project is initiated fully or not at all. The budget does not allow pilot project and experimentation, which is indicated by the top-down approach that was taken [4]. Decomposition of business processes into business services was done upfront for the whole organization. Based on this upfront work, uniform business process were distinguished.

The average maturity stage per aspect that was assessed is shown in Figure 16.

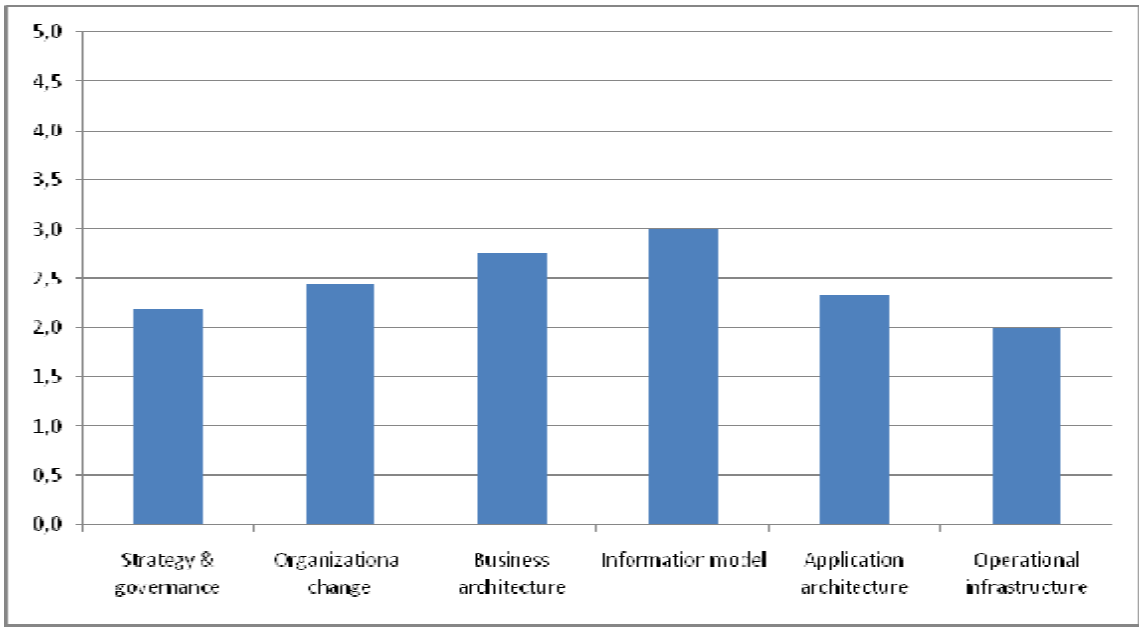

Figure 16: Average maturity stages per aspect for Alpha

The average maturity stage ranges from 2.0 at operational infrastructure, till 3.0 at the information model. The average maturity on aspects range from the stage Applied till the stage Integrated. The most relevant information from the interviews with respect to the questionnaire is discussed per aspect.

Strategy & governance. The top level of the organization, the municipality governors, are not aware of the SOA concept and its possibilities. This in contrast to the management of the public office that is involved in the SOA adoption program. The policy is to decompose business processes into business services, but this is addressed in the IT strategy and not in the business strategy. A strong focus is on outsourcing with smaller application suppliers and increasing bargaining power by training up with other municipalities.

Organizational change. There is a funding model for the development of application components, enabled by the experience in outsourcing. A top-down approach was used to adopt SOA in the organization and only a few people share a vision on the future SOA situation. The SOA program was initiated by a small group. Knowledge and experience about SOA is shared as best practices between projects.

Business architecture. As described earlier, business process are described based on business services. However, this is not performed by orchestrating application services, the organization is currently only experimenting with orchestration. Although this is still experimentation, collaboration between business and IT to support this resides on business unit level. This can be explained by the top-down approach used to adopt SOA in this organization.

Information model. Because of the national reference architecture, open standards for messaging are promoted intensively and used in this organization. Because application services are currently under development, service description is still experimental.

Application architecture. There is a strong need and vision towards decomposing the application systems into application services, but current decomposition is performed into coarse-grained application components. There are two enablers for adopting SOA: there is a national SOA reference architecture, and the architectural decisions are made on enterprise-level using the top-down approach.

Operational infrastructure. Standardization of service messaging is promoted from national level, and facilities are provided to support this. Currently the use of different communication styles between application systems remains an issue.

5.3. Case Beta

The second case is an agency in the defense sector, which we refer to as Beta. Beta is focused on effectiveness of its operations, which are measured by the effectiveness of the different missions. The organization aims at effective use of its resources. To support effectiveness of the missions, the specific needs for each mission are translated to choose capabilities. So when a mission is initiated, the needs are determined first. Based on these needs, the actors that can deliver these capabilities are determined. It does not matter who the actor is delivering the operational service, as long as it is effective within the specific mission.

5.3.1. SOA adoption program

In 2004 a start was made with an Overarching Architecture. The OA covers both the operational services, as well as application components that implement these services. From the OA perspective, the application systems of the member countries are seen as black boxes which are implemented by the member states. The purpose of the OA is to function as an interoperability framework, which is enabled by using open standards. The organization uses the OA to develop and maintain a top-down description of the desired configuration of the organizational system. Three types of architectures exist within Beta:

- Overarching architecture: enterprise architecture with a term of 8 to 10 years. The OA consists of services that can be qualified as business services;
- Reference architecture: architecture for specific domains, with a term of 3 to 5 years. The architecture consists of application components, which can be seen as application components or application services;
- Target architecture: project architecture that defines specific implementations.

The OA was developed by Beta, which is a leading architecture agency. Beta is responsible for the planning, scientific, development, and acquisition functions of communication and information systems. Beta is customer-funded. These customers are other internal agencies that request support in relation to designing and building architectures.

The OA was developed with a top-down approach primarily focused on operational services that can be qualified as business services. A strict distinction is made between services and components.

Services describe the functionality in terms of business logic, and components are their translation to the application logic. The architecture extends their architecture framework with service-views. Although service-orientation is promoted, it is not forced to be applied to the application architecture. The concrete implementations are mainly outsourced to suppliers and proven technology is preferred, so the choice of applying SOA is influenced by these suppliers. However, SOA as a concept is integrated within the OA and influences the reference and target architectures on the long term. Beta mainly focuses on the decomposition of business logic in business services, but decomposition of application logic into loosely application services is also desired.

5.3.2. Results of the maturity assessment

The respondent did not fill in the questionnaire, because the questionnaire was considered to be focused on a commercial organization, and not focused on a organization in the defense sector such as Beta. Although no maturity assessment was performed, we considered the information that was gathered from the semi-structured interview in combination with relevant public documents as useful. The semi-structured interview was conducted with a senior architect.

Strategy & governance: A transformation program was the main facilitator for the service-oriented OA. The transformation has commitment from the top level, and the OA is part of this. Top level support is thus implicit. The strategy is reflected by the transformation program, which addresses the use of business services (operational services). Although there is a future vision for application services, this is not addressed by the IT strategy. The SOA vision is promoted by some experts of Beta. Other agencies within Beta are involved in the development of target architectures in which choices are made about application services. Outsourcing of application development is ordinary within the defense industry.

Organizational change: the organization is mainly effected by the transformation program. No significant organizational changes have been made to support development of application services.

Business architecture: the transformation program mainly focuses on the business architecture. Business logic has been decomposed into business services in an operational service model, and business processes are composed based on the quality of the output of these services. Application services are not the standard for application development, and so business processes are not orchestrated.

Information model: although information is not exchanged between application services, standardization is an important issue within Beta. This is enforced by the fact that application systems in the architecture originate from different member countries. Developing standards for messaging and for semantic interoperability are of high importance within the organization. A standardization agency is responsible for developing and enforcing standards within Beta.

Application architecture: There are three levels of abstraction for architectures as described earlier. The concepts used in the application architecture are called components. Target architectures are developed in projects. Guidance about the target architecture can be given on request, but no enforcement takes place to use SOA for the application architecture.

Operational infrastructure: the interviewee could not respond to this aspect because operational infrastructures are project-related.

5.4. Case Gamma

Organization Gamma, further referred to as Gamma, can be qualified as a large financial service provider. The ICT function supports Gamma with respect to ICT, and it functions as the central IT department. The ICT function is divided into different units, of which one is responsible for the IT architecture of the Gamma.

5.4.1. SOA adoption program

A service architecture was initiated in 1999 for the application architecture to deal with the complexity around application systems and the large amount of application legacy: Gamma has more than 500 applications. The service architecture functions as a SOA reference architecture for project architectures. Within Gamma, SOA is an IT concept and thus the services can be seen as application services. A service in the service architecture is defined with three characteristics :

- A service crosses domain boundaries;
- A service has a meaning within the business;
- A service is on the application-to-application level.

Gamma has around 100 application services. An example is a money transfer service. Projects architectures are under the ownership of a business domain and are considered to conform to the service architecture. Each project is required to deliver a project start architecture that is reviewed by project-relevant IT architects. In this way the project start architecture is audited to see if it conforms to the reference architecture. Services are defined within the project and not by the ICT function. A new program was started to actively guide projects with developing services. Before this program, auditing architects had a more passive role. However, more guidance was needed for the development of application services. Decomposition of the business architecture into business services has been done for a long time but this is seen as independent of the service architecture. Orchestration of business processes based on application services is starting but is in an initial phase.

5.4.2. Results of the maturity assessment

The respondent of the questionnaire was an IT architect within the architecture unit and the promoter of the service architecture. The interview was conducted with the same person.

The results of the questionnaire are shown in Figure 17.

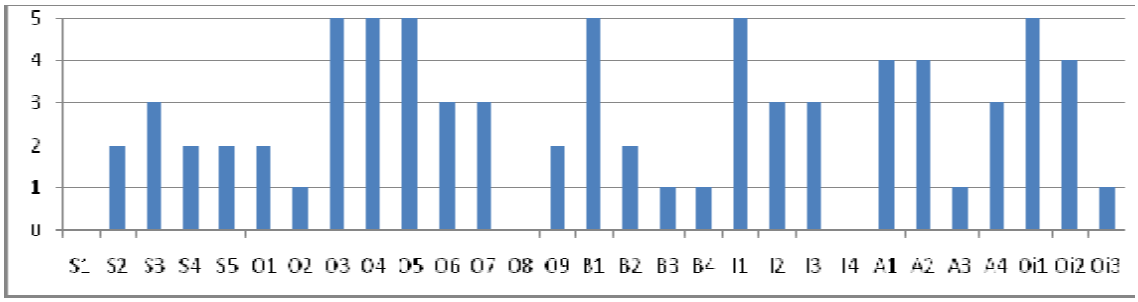

Figure 17: Results of the maturity assessment with Gamma

Two questions could not be answered by the respondent, these are not shown in Figure 11. The results on the key indicators range from stage 0 till stage 5. These extreme answers can be explained

based on the semi-structured interview that was conducted. Firstly, standardization between banks has always been an important issue and thus applications have focused on interoperability with (inter)national clearinghouses. Much of the answers that were assessed in stage 5 (network) can be explained by the involvement of Gamma in standardized financial messaging frameworks like SWIFT, which enforce standardized messaging and inter-organizational applications services. It is not clear if short-term business collaborations are also possible with in- and outsourcing of application services, and thus if the answers reflect the current situation at Gamma, because the network stage should represent a situation where short-term and long-term business collaborations with application services are easily applicable. The respondent used two colors two answer the questions, namely red and yellow. Red represented the top level of business and yellow the top level of IT. Question 1 about commitment of top level was answered with both colors, it was answered in stage 0 for the top level of business, but stage 3 for the top level of IT. This reflects the fact that SOA is seen as an IT concept, and the top level of business is not aware of SOA. Five questions were answered with the red color. These questions were related to the business strategy, business services and outsourcing. As explained earlier, decomposition of business processes in business services was seen as a business matter, and SOA and application services were seen as IT matter.

The average maturity stage per aspect that was assessed is shown in Figure 18.

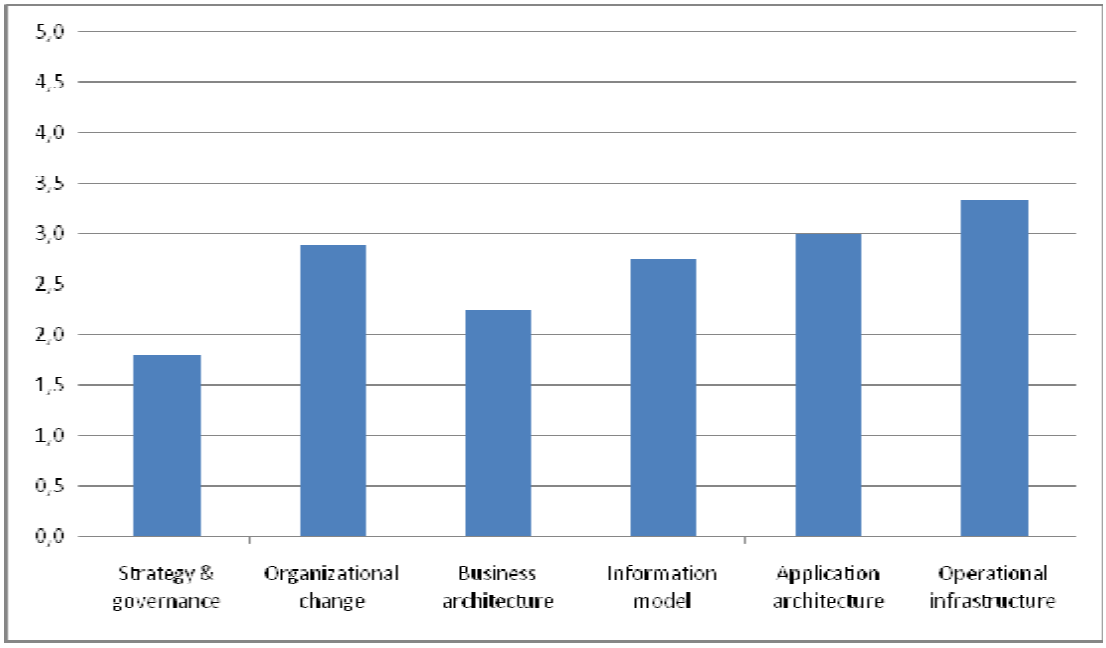

Figure 18: Average maturity stages per aspect for Gamma

The average maturity ranges from 1.8 for the 'strategy & governance' aspect, to 3.3 for the 'operational infrastructure' aspect. So the maturity on aspects range from the stages experimental/applied to the stages applied/institutionalized. The most relevant information from the interviews in addition to the questionnaire is discussed below per aspect.

Strategy & governance: a clear distinction was made between the top level of the organization and the management of the ICT function. The term SOA was only used within IT and business projects that use the service architecture. This was reflected by the fact that SOA was addressed within IT

strategy and the service architecture document. The ICT managementteam gave support for the service architecture. Decomposing business logic into business services was not a matter of ICT and was only included in the business and business strategy, and not in the IT strategy. Management commitment and funding were seen as essential for developing a SOA, because the initial costs are high. Without a budget, it is considered to be impossible to develop a service portfolio.

Organizational change: applications services are owned by the business domain, but the central IT department audits the development of application services in projects. A new program was started to actively support the development of application services. Some training is given, but this is mainly on-the-road training. Best practices are used between projects to transfer knowledge and experience about service development. Billing of services is possible, but this is currently not done. An issue related to trust within a SOA was related to the blackbox principle: programmers want to see the code behind application services developed by others, to see if it is well-coded. This contradicts the autonomy of services.

Business architecture: as explained earlier, the decomposition of business logic and description of business processes based on these business service is done on the business side. Orchestration based on application services is still experimental and is done in joint teams of business and IT people.

Information model: standardization is strongly forced by the financial messaging networks the bank is involved in. There are standards for message formats and semantics Gamma and all other financial service providers have to comply to. Gamma has A data model, which defines standards for internal use. Services are described in a standard way in most parts of the organization, but these descriptions are not stored in a service registry.

Application architecture: development of application services is considered to be applied in every project based on the SOA reference architecture. However, development of services in projects is performed ad hoc, and so a new program is started to actively guide the development of services. Project architectures are reviewed by IT architects from the architecture unit at the start of the project, based on policies. Application services are offered centrally throughout the organization. An example is the BKR-service to check if a customer is financially credible.

Operational infrastructure: The service messaging framework is inter-organizational. An example is SWIFT, which is a financial messaging network for banks and other financial institutions. Secure messaging is an important issue within the financial sector. Security is integrated within financial messaging networks.

5.5. Conclusions

Three case studies were conducted to assess the maturity of enterprise-wide SOA adoption with three organizations. The goal of these case studies was to test the constructed stage maturity model in practice with different type of organizations. The case study involved filling in the questionnaire, which functioned as a maturity assessment tool, and having a semi-structured interview with the respondents to get insight in the SOA adoption program within the organization.

6. Cross-case analysis

This chapter reports the cross-case analysis that was performed. Section 6.1 discusses the research method that was used to conduct the cross-case analysis. Section 6.2 discusses the general observations from the three cases. Section 6.3 discusses the cross-case analysis according to the 6 maturity stages. Section 6.4 presents the conclusions of this chapter.

6.1. Research method

The structure of the cross-case analysis is performed according to the six maturity aspects. The software program NVivo was used to relate quotations from the interviews, which were conducted in the three case studies, to the maturity aspects, key indicators and stages. Not only interview quotations were related, also quotations from expert interviews, documentation, and publications gathered within this research were related to the aspects, key indicators and stages. In this way, the cross-case analysis was supported by data, which was ordered according to the elements of the stage maturity model. The cross-case analysis does not include a comparison of the maturity assessment values of the different cases, because we consider each case as too complex and unique. Furthermore, the cross-case analysis does not validate the maturity stages of SMM-SOA, because we consider the number of cases for this purpose as not enough. The main goals of the cross-case analysis were to enrich the maturity aspects of SMM-SOA with experiences in practice, and to identify cross-case patterns in SOA adoption.

6.2. General observations

In the process of sampling cases, selecting cases and conducting the case studies we made some general observations.

Firstly, the SOA-related terminology was used and interpreted in a variety of ways by the organizations and people involved, including the experts. The key terms used in the case studies were *Service-Oriented Architecture, service* and *maturity*. The terms SOA and service were interpreted either in a more business-oriented or application-oriented way. When used in the context of application architecture, the term application service was sometimes used to refer to applications components that are tightly coupled. In-depth questioning was needed in the semi-structured interviews, to determine the precise scope of the interview. The term maturity often led people to think of process maturity models like CMMI. Lewis et al. also made the observation that the persons involved with SOA typically have a limited understanding of SOA, and that they have misconceptions about SOA that lead them to oversimplify the expected effort required to adopt an enterprise-wide SOA [5]. However, other literature states that organizations do not blindly adopt SOA, and that they choose a focused approach to handle the complexity [11, 48].

Secondly, interviewees had a clear and focused vision on the SOA adoption program within their organization. The focus of the SOA adoption program was either on the decomposition of business processes into business services, or decomposition of the applications into application services. The clear vision can be explained by the fact that the interviewees could be qualified as SOA evangelists. Each interviewee was found based on references in business literature about SOA adoption, in which they reported their experience with their SOA adoption program.

Thirdly, organizations that were confronted with problems in their SOA adoption program hesitated to participate in our research. This can be explained by the fact that the maturity of SOA adoption was assessed within this research, and the risk of getting a 'low' maturity as verdict. The observations that we made by having intake interviews with organizations, which were not willing to cooperate in a case study, were also used to perform the cross-case analysis.

6.3. Analysis based on maturity aspects

Before a cross-case analysis is performed based on the maturity aspects, the most relevant differences between the organizations are summarized in Table 13.

Table 13: General differences between the three cases

	Alpha	Beta	Gamma
Sector	Public sector	Defense sector	Financial sector
Number of services	5	-	Around 100

An important difference is the type of sector of the organizations. The only organization aimed at making profit is Gamma. The granularity of services is also a difference. While the services at Alpha are coarse-grained, Gamma has more fine-grained services. The size of the organizations is also a difference. Alpha can be qualified as a small organization in relation to Beta and Gamma.

6.3.1. Strategy & governance

There is consensus that commitment from management or from the top level of the organization is important, this is confirmed by literature [47]. An IT architect stated that SOA is not only a software package, but that the organizational side must be managed. Except for Gamma, the SOA adoption program or project was part of a larger program that was already started, and thus commitment was given upfront. Financial support is seen as the most important enabler for the SOA program. Without money, a service portfolio cannot be developed, which relates to the high initial costs of the SOA adoption program. Different issues play a role here. Within a financial service organization, due to a reorganization, the investment in their SOA adoption program was stopped, causing the program to stop. In a insurance organization, the problem related to the top level responsible that had done investments in IT technology in the past that failed, and even though the responsible was convinced by the benefits of SOA adoption, given the situation it was impossible to give financial support. In all three cases, a distinction could be made between top level and IT management. The formal approval for the SOA program was directly or indirectly given either by top level or by IT management. One IT architect stated that the question for support should be in the boardroom, but the how-question must be answered outside this room. Another IT architect pointed at a risk in this, because if the top level did not understand how SOA could support the organization, commitment would be lost with time. In contrast, another IT architect pointed at the responsibility of the architect who should concretize the goals of applying SOA.

Two approaches for SOA adoption could be distinguished from the cases: applying SOA either to the business architecture (Alpha/Beta) or to the application architecture (Gamma). Alpha and Beta had different reasons to decompose the business architecture into business services, respectively to increase effectiveness and to increase efficiency.

The three cases all used a top-down oriented approach, where analysis and decomposition of services is done upfront [4]. A bottom-up approach was found to be more problematic in relation to commitment, as each time 'buy-in' of the business has to be obtained as application services are developed. The IT architects in our group discussion agreed that SOA cannot be introduced as a big bang solution. The consensus in literature confirms this by stating that organizations should take a gradual approach to SOA adoption [11, 48]. This may contradict to the top-down approach that was used in the three cases, but the possibility of a top-down approach was very dependent on the situation. Often the SOA evangelists had a position within their organization in which they could promote and gain support for SOA to be applied throughout the organization.

6.3.2. Organizational change

There is consensus in literature and in our group discussion with IT architects, that the organizational aspects are important within a SOA adoption. One SOA project leader refers to this as 'setting up a SOA is 20% technique and 80% organization'. In contrast, within a public administration organization, there was no plan to change the organization in relation to SOA, because this was not required since it was only considered to be an IT change. One IT architect stated that a project organization is not that open for a SOA implementation, because organizing a SOA can be risky, and projects do not want to take that risk. The related proposition is that SOA adoption is probably easier in a line organization. SOA adoption can have consequences to the organizational structure. In a financial organization this led to reorganizations in different departments.

Funding and billing models for application services were not yet standardized within the organizations. In Case Alpha, there were five coarse-grained application components, so there was no need for a predetermined funding and billing model. The funding and billing of the use of these application components was managed ad hoc. The application services of Gamma were more fine-grained, but the business domain owned the services in projects, so funding was provided by the business domain. Billing of services was not yet done within the organization in each cases. One SOA governance expert stated that this can be arranged by three methods: corporate taxes, direct chargeback for services, or a mix of these two methods. When offering application services to other organizations or vice versa, agreements about the price were made upfront. An important aspect about funding and service use is proposed by Gamma, stating that the ideal situation in which users can pick a service from a catalogue and start using it is unrealistic. Before using a service, the service provider must investigate if it is possible to deliver the service in a higher volume with the current resources. If not, the resources that are required in the new situation have to be organized first. In Case Beta and Gamma, services were owned and developed within business projects at a lower level of architecture, and guidance was provided for development.

Within each case, the interviewee had a clear vision about how the organization could apply SOA. One IT architect stated that communicating this vision to the key players within the organization is important. This vision was communicated in different ways in each case, for example, through the SOA reference architecture, by giving presentations, and by putting information on the intranet. In none of the cases official SOA training was given. SOA training was arranged ad hoc when needed and training was given on-the-job. In one organization, a matrix was used to determine which skills and knowledge different stakeholders were considered to possess, to develop services. Knowledge and experiences were often transferred as best practices and guidelines in a reference architecture.

One IT architect stated that the SOA vision should also address the benefits of SOA for the organization. Three questions should be asked: Why is transformation needed? Where are the pain points in the organization? And where are potential opportunities? Although this should be done, one IT architect stated that this is often clear in the board room, but is not communicated to the organization. Another issue relates to the belief in the SOA effectiveness. Because services act as a black box and encapsulate the application logic, this creates trust problems. Programmers want to see the program code behind the other services, which is in contrast with the principles of SOA. A better procedures to test services is regarded as a solution to resolve this issue.

6.3.3. Business architecture

An important distinction that could be observed from the three cases is between modeling in terms of business services, and orchestration of business processes based on application services. The first one was often the responsibility of a 'business process'-oriented function in the organization, while an IT architecture function was often responsible for orchestration. It can be concluded from the cases, that business process orchestration is still in an experimental phase. One SOA governance expert proposed that application services should be institutionalized first before orchestration can take place. In contrast, there is more experience with the decomposition of business processes into business services, which have a specified input and output. Within Beta, the focus was on the output, since the main goal was to increase effectiveness. This is in contrast to Alpha, which models business services to increase efficiency, by forcing processes into uniform process models.

6.3.4. Information model

One IT architect pointed that application services cross boundaries, and not only standards for message formats but also for semantic interoperability are crucial. There was consensus about this in the three cases, but this seems to be hard to realize. Working groups exist at enterprise-level or at sector-level to develop, promote and/or enforce standards for message formats, ontology and taxonomy. Both for Alpha and Gamma, standards are enforced centrally in the sector. Alpha has to follow the national reference architecture NORA that also defined standards for message formats. Gamma has a long history with financial messaging, such as messaging of financial transactions within international financial messaging network like SWIFT. Within these networks, banks are enforced to comply to the standard for message formats. Standardization within organizations is found to be difficult. One SOA project leader stated that the more people involved, the more difficult it is to reach agreement. From the cases we conclude that services are described, but that they are often stored in an experimental service registry. Service discovery is done by social networking, in which also the terms of service use are agreed upon.

6.3.5. Application architecture

The development of application components differs in the cases. Case Gamma is the only case, in which application services are considered to be developed in projects. Within Case Beta, functionality is defined in the overarching architecture, but implementation choices can be made in target architectures. These choices are highly dependent on what suppliers have to offer, as outsourcing is the standard in the defense industry. In Case Alpha, five coarse-grained application components were defined, and the development of these components were outsourced to smaller suppliers. The overarching architecture at Beta and the service architecture, both function as service-oriented reference architectures. In Case Alpha, a SOA reference architecture was used to guide development of application components. The only case in which application services were developed, service

lifecycle management was still in an experimental state. This is because more guidance was considered to be needed for developing services.

6.3.6. Operational infrastructure

Based on the case studies, a distinction can be made between the internal service messaging framework, and participation within external service messaging frameworks (sector-central) organizations. External service messaging frameworks involve many parties, and so official standards are enforced. The internal service messaging framework is under development. In Case Gamma, 90% of the message queues were in the test phase, while only 10% was in production. The definition of service level agreements was still in an initial phase in each case.

6.4. Conclusions

A cross-case analysis was conducted, structured according to the six maturity aspects. The three general observations of the case studies were:

- The terminology that was used was interpreted differently by involved people;
- The organizations in the case studies had a clear vision on their SOA adoption program;
- Organizations facing issues with SOA adoption hesitated to participate in our research.

The key points of analysis that can be extracted from the cross-case analysis are:

- Commitment from the top level or management is important for successfully adopting an enterprise-wide SOA;
- The initial phase of SOA adoption is costly, investments have to be done at this point;
- SOA was either applied to the business architecture, or to the applications architecture;
- The bottom-up approach is risky in the sense that buy-in has to achieved from the business during the program;
- Organizational aspects are important to consider when adopting an enterprise-wide SOA;
- Business process orchestration is in an initial phase;
- Standardization is considered to be important, but its realization is difficult;
- A distinction must be made between internal and external service messaging frameworks.

7. Conclusions

This chapter presents the conclusions of this thesis by discussing the research results. Section 7.1 answers the three research questions. Section 7.2 discusses the relevance of the results. Section 7.3 discusses the results of this research and its limitations. Section 7.4 discusses further research.

7.1. Conclusions

The main research question for this thesis was formulated as:

> ***How is an enterprise-wide Service-Oriented Architecture (SOA) adopted by organizations and how is SOA integrated in the IT and business strategy of organizations?***

This research question was subdivided into three research questions, which are answered below.

RQ1: Which stages can be distinguished in the adoption process of an enterprise-wide SOA?

RQ2: How can maturity of SOA adoption be measured and made operational?

Research questions 1 and 2 are answered by the matrix of our stage maturity model, which is presented in Table 14. The stages are shown on the horizontal axis, the most important aspects on the vertical axis. Our model was further explained in Chapter 4.

Table 14: Stage maturity model for the adoption of an enterprise-wide SOA (SMM-SOA)

	Siloed	Experimental	Applied	Integrated	Institutionalized	Networked
Strategy & governance	Functional strategy	Ad hoc SOA strategy	Best practice SOA strategy	Formalized SOA strategy	Enterprise SOA strategy	Network SOA strategy
Organizational change	Silo's owned by functional IT teams	Services owned by architects	Services owned by project teams	Central services owned by central IT department	Enterprise services owned by business units	Network services owned by business units
Business architecture	Process automated with components	Ad hoc process orchestration	Best practice process orchestration	Formalized process orchestration	Process orchestration with enterprise services	Process orchestration with network services
Information model	Domain specific information models	Ad hoc domain-crossing information model	Best practice domain crossing information models	Common information model	Enterprise information model	Network information model
Application architecture	Componentized application silo's	Experimental services	Best practice services	Formalized services	Enterprise services	Network services
Operational Infrastructure	No messaging framework	Experimental messaging framework	Project related messaging framework	Common messaging framework	Enterprise messaging framework	Network messaging framework

RQ3: How is SOA integrated in the IT and business strategy of organizations?

Organizations follow a different approach in SOA adoption: they are either focused on decomposing the business logic into business services, or decomposing the application logic into application services and orchestrating business processes based on these services. The former is included in business strategy while the latter is included in IT strategy.

7.2. Relevance of results

The managerial and theoretical relevance of this research are discussed below.

7.2.1. Managerial relevance

At first, our stage maturity model can function as a roadmap for organizations that are considering or already adopting SOA. They can assess their current maturity on different aspects, and use the current maturity as basis to determine goal stages on the different aspects. IT consultancy organizations can use the model to assess the current maturity of client organizations, and to determine the desired goal situation. The stage maturity model is generic, and should be adjusted to the pain points and priorities of individual organizations. Our model can also help organizations to develop a SOA vision and to communicate this to their organization, by showing the technical and organizational aspects of SOA adoption.

7.2.2. Theoretical relevance

Our stage maturity model offers a scientific framework for SOA adoption within the IT field. The model distinguishes growth stages that represent the adoption process of organizations, technical and organizational aspects that we consider as important for adoption success, and an assessment tool to assess organizations. Using the elements of SMM-SOA, the model can be used to analyze the adoption behavior of organizations that are adopting an enterprise-wide SOA. Especially, the model can be used to explain the success and failure of SOA adoptions and to learn from current and future experiences.

7.3. Discussion

Below the results of the research are discussed and its limitations, structured according to the general considerations for quality of research [60].

7.3.1. Construct validity

Construct validity concerns the question: were appropriate operational measures used for the concepts being studied [60]. Our stage maturity model is the operational measure for the concept 'maturity of enterprise-wide SOA adoption'. Triangulation was used to construct the model, by interviewing experts related to IT architecture, SOA adoption, and/or ERP implementation. The experts that were interviewed can be categorized into three groups: IT architecture experts from a large IT consultancy organization, experts on SOA adoption in practice, and academic researchers that have published about SOA adoption. Our stage maturity model was also reviewed in interviews with experts. A limitation about the stage maturity model is that it has mainly been constructed and validated based on expert opinions of a desired future situation, and not based on empirical evidence. This is in contrast with the ERP stage maturity model that was presented in Section 3.4.1, this model was based on the current state of adopting programs. Because SOA is a young technology, this was not possible for SOA adoption. Case studies were used to validate the relevance of the different maturity aspects, but the case studies could not be used to validate the stages. Another

limitation is that the key indicators of the stage maturity model have no weighing factors within their maturity aspect. All key indicators were considered to have an equal impact on their maturity aspect, and so only the average maturity value could be calculated.

7.3.2. Internal validity

Internal validity concerns the question: can the results be explained by other factors [60]? In each case one informant was chosen as a respondent to the questionnaire and as interviewee. This person was selected based on sufficient knowledge about the SOA adoption program within their organization, because the aspects cover both organizational and technical aspects. It can be questioned why only one person was selected as informant per organization. This was partly due to time limitation. The other reason was that the person was considered to be the SOA expert within the organization, and others referred to this person when asked for participation. Semi-structured interviews were conducted with the respondent to interpret the answers given in the questionnaire within the specific context.

7.3.3. External validity

External validity concerns the question: can the results be generalized to similar situations [60]? To support generalization, three different organizations were selected to respond to the questionnaire. They differed in the size of their organization and the type of sector. A main concern is the different focus and scope of the SOA adoption program. Two organizations were mainly focused on decomposition into business services, while the other was focused on decomposition into application services. This research was aimed at exploring the practice situation and more cases are needed to generalize the stage maturity model based on characteristics like the type of sector and the size of the organization.

7.3.4. Reliability

Reliability concerns the question: can the operations of a case study be repeated with the same results [60]? Interviews were conducted semi-structured and thus the topics could be influenced by the interviewee and the researcher. Analysis was structured, by coding quotations of the notes and transcriptions to the elements of our stage maturity model. The answers to the questionnaire were randomly ordered to avoid respondents to influence the answers. This was done to handle a potential 'image' bias, in which respondents assess their organizations as more mature than they actually are, to present their organization in a more positive way. To collect the data, a case study database was developed with documents, publications, transcriptions of the interviews and interview notes. In this way, the raw data was divided from the analysis data.

7.4. Further research

Looking at the limitations discussed in section 7.3, further research can focus on the construction of SMM-SOA, and how the model can be generalized to different organizations. SMM-SOA was mainly constructed based on expert opinions. SOA is still a young technology, and it is still evolving to be a mature technology. Further research can revise the maturity stages and maturity aspects based on experiences in practice. The key indicators in SMM-SOA each have an equal weighting factor within their maturity aspect. Further research can add weighing factors to the different key indicators within a maturity aspect, to assess a weighted maturity value for the six aspects of the model. Extending to this, weighting factors can be added to the values of the six maturity aspects to calculate an overall maturity value for an organization.

Further research is also needed to generalize the model based on experiences in different organizations. Within this research, three organizations were assessed on their maturity of enterprise-wide SOA adoption. The three cases were considered to be too complex and unique to generalize upon. Further research should assess more organizations, to test the external validity of our model. Differentiation between organizations can take place based on characteristics such as the type of sector, and the size of the organization.

Another potential direction of future research is how adopting organizations and IT consultancy organizations use SMM-SOA. Organizations can be analyzed by looking at how they use a stage maturity model, and if and how these models can assist the adoption process effectively.

References

1. Themistocleous, M., Z. Irani, and R.M. O'Keefe, *ERP and application integration: exploratory survey*. Business Process Management, 2001. **7**(3): p. 195-204.
2. Alonso, G., et al., *Web Services: concepts, architectures and applications*. First ed. 2003: Springer. 354.
3. Papazoglou, M.P., *Web Services: principles and technology*. First ed. 2007: Prentice Hall.
4. Erl, T., *Service-Oriented Architecture: Concepts, technology, and design*. The Prentice Hall Service-Oriented Computing Series. 2005: Prentice Hall PTR.
5. Lewis, G.A., et al. *Common misconceptions about service-oriented architecture*. in *Commercial-off-the-Shelf (COTS)-Based Software Systems, 2007. ICCBSS '07. Sixth International IEEE Conference on*. 2007: IEEE Computer Society.
6. van den Berg, M., J. Hompes, and J. Truijens, *SOA in Nederland: een digitale aandoening?*, in *PrimaVera Working Paper Series*. 2007, Amsterdam Business School: Amsterdam, the Netherlands. p. 25.
7. Brahe, S., *BPM on top of SOA: experiences from the financial industry*, in *Business Process Management*. 2007. p. 96-111.
8. Bloomberg, J. (2005) *What to look for in a SOA maturity model?* Zapflash.
9. Mahajan, R., *SOA and the enterprise: lessons from the city*, in *IEEE International Conference on Web Services (ICWS'06)*. 2006, IEEE Computer Society: Chicago, USA. p. 939-944.
10. Acharya, M., et al., *SOA in the real world: experiences*, in *Service-Oriented Computing - ICSOC 2005*. 2005, Springer-Verlag: Amsterdam, the Netherlands. p. 437-449.
11. Legner, C. and R. Heutschi. *SOA adoption in practice: findings from early SOA implementations*. in *Proceedings of the Fifteenth European Conference on Information Systems*. 2007. St. Gallen, Switzerland: University of St. Gallen.
12. Austvold, E. and K. Carter, *Service-Oriented Architectures: survey findings on deployment and plans for the future*. 2005, AMR Research, Inc. p. 28.
13. Fontanella, J. and R. Saia, *The SOA in IT benchmark report: what CIOs should know about how SOA is changing IT*. 2005: Boston, MA, USA. p. 31.
14. Bieberstein, N., et al., *Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals*. IBM Systems Journal, 2005. **44**(5): p. 691-708.
15. Malinverno, P., et al., *Hype cycle for application infrastructure middleware, platforms and architecture*. 2007, Gartner. p. 51.
16. Rogers, E.M., *Diffusion of innovations*. Fifth ed. 2003: Free Press.
17. Babcock, C., *The SOA gamble: one in three companies are dissapointed, our survey finds*, in *InformationWeek*. 2007.
18. *CMMI for development, version 1.2*. 2006, CarnegieMellon Software Engineering Institute: Pittsburgh, PA, USA. p. 573.
19. Meijborg, R., *Service oriented enterprise: service oriented enterprise maturity framework*, in *Faculty of Information Science*. 2006, University of Utrecht: Utrecht, the Netherlands.
20. Nolan, R.L., *Managing the crises in data processing*. Harvard Business Review, 1979. **57**(2): p. 115-126.
21. Gibson, C.F. and R.L. Nolan, *Managing the four stages of EDP growth*. Harvard Business Review, 1974. **52**(1): p. 76-88.
22. Holland, C.P. and B. Light, *A stage maturity model for enterprise resource planning systems use*. SIGMIS Database, 2001. **32**(2): p. 34-45.
23. Holland, C.P. and B. Light, *A critical success factors model for ERP implementation*. IEEE Software, 1999. **16**(3): p. 30-36.
24. Hong, K.-K. and Y.-G. Kim, *The critical success factors for ERP implementation: an organizational fit perspective*. Information & Management, 2002. **40**(1): p. 25-40.
25. Sprott, D., *SOA maturity assessment survey*. CBDI Journal, 2007(3): p. 4-10.

26. Szakal, A., *Open Group Service Integration Maturity Model (OSIMM): launch presentation and notes from first working group 1.2.07*. 2007. p. 18.
27. Lam, W., *Investigating success factors in enterprise application integration: a case-driven analysis*. European Journal of Information Systems, 2005. **14**(2): p. 175-187.
28. Meier, F., *Service oriented architecture maturity models: a guide to SOA adoption?*, in *School of Humanities and Informatics*. 2006, University of Skövde: Skövde, Sweden. p. 53.
29. Paré, G., *Investigating information systems with positivist case study research*. Communications of the Association for Information Systems, 2004. **13**: p. 233-264.
30. *NVivo 8 research software for analysis and insight*. 2008, QSR International. Research software for qualitative analysis.
31. Eisenhardt, K.M., *Building theories from case study research*. Academy of Management Review, 1989. **14**(4): p. 532-550.
32. Britton, C. and P. Bye, *IT architectures and middleware: strategies for building large, integrated systems*. Second ed. 2004: Addison-Wesley.
33. Group, T.O., *The Open Group Architecture Framework (TOGAF) Version 8.1.1*. 2007.
34. Lee, J., K. Siau, and S. Hong, *Enterprise integration with ERP and EAI*. Communications of the ACM, 2003. **46**(2): p. 54-60.
35. OASIS, *Reference model for service oriented architecture 1.0*. 2006, OASIS. p. 31.
36. Harding, C. *Definition of SOA*. 2006; Available from: http://www.opengroup.org/projects/soa/doc.tpl?gdid=10632.
37. Schmelzer, R. (2006) *Solving the service granularity challenge*. Zapflash.
38. Steghuis, C., *Service granularity in SOA projects: a trade-off analysis*. 2006, University of Twente: Enschede, the Netherlands. p. 90.
39. Papazoglou, M.P. and M. Ribbers, *e-Business: organizational and technical foundations*. 2006: John Wiley & Sons.
40. W3C, *SOAP version 1.2 part 1: messaging framework (second edition)*. 2007.
41. W3C, *Web Services Description Language (WSDL) version 2.0 part 1: core language*. 2007.
42. OASIS, *UDDI Version 3.0.2*. 2004.
43. OASIS, *Web services business process execution language version 2.0* 2007, OASIS. p. 264.
44. Juric, M.B. (2008) *A hands-on introduction to BPEL*. Developer: J2EE & Web Services.
45. Kenniscentrum, *NORA 2.0, Nederlandse Overheid Referentie Architectuur: samenhang en samenwerking binnen de electronische overheid*. 2007, Kenniscentrum. p. 283.
46. Baskerville, R.L., *Investigating information systems with action research*. Communications of the Association for Information Systems, 1999. **2**(19): p. 1-32.
47. Holtkamp, M., *Public services integration & interoperability: the use of an enterprise service bus to facilitate service-oriented architecture in a public environment*. 2006, University of Twente: Enschede, the Netherlands. p. 132.
48. Hau, T., et al., *Where to start with SOA: criteria for selecting SOA projects*, in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. 2008, IEEE Computer Society. p. 314-314.
49. Schepers, T., et al., *Het inrichten van een SOA governance levenscyclus*. Via Nova Architectura, 2008: p. 1-6.
50. Oxford dictionaries. *Definition of maturity*. 2007 November 20, 2007].
51. Sprott, D., *The SOA maturity model*. CBDI Journal, 2005(12): p. 4-17.
52. *Diffusion of innovations*. 2008; Available from: http://en.wikipedia.org/wiki/Diffusion_of_innovations.
53. Venkatesh, V., Morris, MG, David, GB, Davis, FD, *User acceptance of information technology: towards a unified view*. MIS Quarterly, 2003. **27**(3): p. 425-478.
54. Scheper, W., *Business IT alignment: oplossing voor de productiviteitsparadox*, in *Information and organization*. 2002, University of Utrecht: Utrecht, the Netherlands.
55. Arsanjani, A. and K. Holley (2005) *Increase flexibility with the Service Integration Maturity Model (SIMM)*. IBM developerWorks.

56. Sprott, D., *The SOA maturity model: Part II*. CBDI Journal, 2006(3): p. 5-11.
57. Muntslag, D.R., *De kunst van het implementeren*, in *Technology & management and computer science*. 2001, University of Twente: Enschede, the Netherlands. p. 32.
58. Willcocks, L. and G. Smith, *IT-enabled business process reengineering: organizational and human resource dimensions*. The Journal of Strategic Information Systems, 1995. 4(3): p. 279-301.
59. Daft, R.L., *Management*. Fifth ed. 2000: Harcourt College Publishers.
60. Yin, R.K., *Case study research: design and methods*. Third ed. Applied social research methods series. 2003: Sage Publications.

Appendix A: Maturity assessment tool

This appendix contains the questionnaire based on the SOA stage maturity model. Within the second part, the general questions, the relevance of the questions are explained. This was not done in the questionnaire that was send to the respondent in the three case studies.

Introduction

This questionnaire is part of a master thesis project about the adoption of a Service-Oriented Architecture by organizations. This thesis is written as part of the master study Business & IT given at the University of Twente. A Service-Oriented Architecture (SOA) is a framework for integrating business processes and supporting IT infrastructure by secure, standardize components that can be reused and combined to address changing business priorities. Services can be divided into business and application services. A service is a logical representation of respectively business or application logic with a specified outcome. Services are reusable, share a formal contract, are loosely coupled, abstract underlying application logic, are composable, are autonomous, are stateless and are discoverable. The goal of this research is to propose a maturity model for SOA adoption. A maturity model distinguishes different stages in the adoption of a SOA by organizations. You have been selected to fill in this questionnaire because you are involved in a SOA project or SOA program in your organization. The adoption of SOA is split up in six aspects of maturity: Strategy & governance, organizational change, business architecture, information model, application architecture and operational infrastructure. Each of these six aspects is further broken down in multiple questions. The total number of questions is 31. Each of the questions belonging to the six aspects have six options that represent descriptions of different situations of SOA adoption. Please choose the description that best fits the current situation of your organization. The questionnaire starts with a section covering general questions about you and your organization. After the general questions six sections follow, each section covers an aspect of maturity of SOA adoption. The questionnaire ends with an open field for your comments. If you have any questions about this questionnaire, you can contact me by e-mail or phone.

General questions

Name of respondent
This is an obvious question, not relevant for analyzing the results.

Function title of respondent
This is relevant for analyzing the information that is provided by the respondent.

Organizational role in relation to SOA adoption and services
The function title does not indicate by explain how the respondent is related to the SOA adoption. The involvement in the SOA adoption program may be one of the roles of the respondent within the organization.

Age
The age of the respondent may be relevant by indicating the interest in new innovations.

Gender
This is a general question.

Name of organization
General question about the organization.

Year that the organization was founded
This may be relevant for the amount of legacy applications and the degree of innovation.

Industry sector
The type of industry influences the context in which SOA is adopted.

Number of employees
This is an indication for the size of the organization.

Number of IT employees
This is an indication of the importance of IT within the organization.

Percentage of internal IT employees
This is an indication for the degree in which expertise about IT and SOA is needed.

Average yearly turnover
This is an indication of the size of the organization.

Year started with service-orientation and SOA thinking in the organization
This is an indication of the degree of innovation.

Has the organization done take-overs, mergers or reorganizations in the last 5 years?
Take-overs, mergers and reorganizations often lead to integration issues within and between the organizations. This may influence the need for integration with services.

Strategy & governance

This aspect describes how SOA is included in the strategy and how SOA is governed. It describes how the top level and management of the organization are involved in the adoption of SOA. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

1. **How would you describe the commitment to SOA at the top level of your organization?**

- The top level is unaware of SOA or does not see it currently as an opportunity for the organization;
- The top level may know about pilot projects that are running with SOA and/or give the pilot projects support;
- The use of SOA in some business critical projects is supported by the top level and seen as business opportunity;
- The top level identifies SOA as critical for the organization and supports the use of SOA throughout the organization;
- The top level fully supports the use of SOA throughout the organization;
- The top level fully supports the use of SOA within the organization and in business collaboration;
- No opinion.

*Business activity can be decomposed into **business services**: Basic business activities that can be reused within multiple business processes.*

2. **Does the business strategy of your organization address business services?**
- The business strategy addresses the organization as functional units;
- The business strategy addresses the potential of decomposing the organization in business services;
- The business strategy addresses using business services for parts of the organization;
- The business strategy addresses using business services throughout the organization;
- Internal business services are fully institutionalized in business strategy;
- Internal and external business services are fully institutionalized in business strategy;
- No opinion.

3. **Does the IT strategy of your organization address application integration with application services?**
- The IT strategy addresses the use of individual software applications;
- The IT strategy addresses the potential of application integration with application services;
- The IT strategy addresses application integration with application services for different software applications;
- The IT strategy addresses application integration with application services throughout the organization;
- Application integration with internal application services is fully institutionalized in IT strategy;
- Application integration with internal and external application services is fully institutionalized in IT strategy;
- No opinion.

4. **How would you describe SOA governance and policies in your organization?**
- There are no policies to support SOA governance;
- Policies are developed ad hoc within SOA pilot projects;
- There are policies to support the governance of services within some business critical projects;
- SOA governance is centralized and applied in most parts of the organization;
- SOA governance is fully integrated within IT governance and applied throughout the organization;
- SOA governance for internal and external services is fully integrated within IT governance;
- No opinion.

5. **What is the status of outsourcing in your organization?**
- There are no contracts for outsourcing;
- Some departments experiment with outsourcing processes;
- Departments distinguish between in- and outsourcing;
- In- and outsourcing of processes is centralized and applied in most parts of the organization;
- In- and outsourcing of processes is institutionalized at enterprise level and applied throughout the organization;
- The organization insources all its core processes, all supportive processes are outsourced;
- No opinion.

Organizational change

This aspect describes the way in which the organization has evolved as a result of the SOA adoption. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

6. **How are application services funded in your organization?**
- Application services are not funded;
- SOA pilot projects are funded;
- Some business critical projects that use application services are funded;
- Most central application services at enterprise level are funded;
- There is a institutionalized funding system for shared application services at enterprise level;
- The funding system supports central application services that are used internally and externally of the organization;
- No opinion.

7. **How are application services billed to service users in your organization?**
- There are no application services;
- Application services are developed ad hoc and are not billed to service users;
- Application services are billed to service users within some business critical projects;
- Most central application services are now billed to service users throughout the organization;
- Billing of service users is institutionalized for all application services throughout the organization;
- All application services throughout the organization are billed to service users internally and externally of the organization.
- No opinion.

8. **Who owns application services in your organization?**
- There are no application services;
- Application services are owned by IT architects at pilot project level;
- Application services are owned by different IT architects within some business critical projects;
- Application services are owned by the central IT departments of business units;
- Application services are owned by business units and offered only internally;
- Application services are owned by business units and offered internally and externally;
- No opinion.

9. **What is the scope of the SOA project or program in your organization?**
- There is no SOA project or program;
- Pilot project or proof of concept project with a narrow scope;
- SOA is applied to some business critical projects;
- Service development and maintenance is centralized within business units or at enterprise level;
- Service development and maintenance is institutionalized at enterprise level for all application services throughout the organization;
- Service development and maintenance is institutionalized at inter-organizational level;
- No opinion.

10. How would you describe the vision of the future SOA situation in your organization?
- There is no vision of a future SOA situation;
- Some people have thought about the future SOA situation;
- There is a vision of the future SOA situation within some business critical projects;
- Most parts of your organization share a vision of the future SOA situation;
- There is one clear vision about the future SOA situation shared throughout the organization;
- There is one clear vision about the future SOA situation shared throughout the organization and on inter-organizational level;
- No opinion.

11. How would you describe the belief in the effectiveness of SOA in your organization?
- Nobody believes in the effectiveness of SOA;
- Some people belief in the effectiveness of SOA;
- In some departments there is a belief in the effectiveness of SOA;
- In most parts of the organization there is a belief in the effectiveness of SOA;
- There is a strong belief in the effectiveness of SOA for internal use throughout the organization;
- There is a strong belief in the effectiveness of SOA for internal and external use throughout the organization;
- No opinion.

12. How would you describe SOA awareness in your organization?
- Some or no people know about the concept of SOA;
- Some people are aware of application services and know about pilot project(s);
- People participating in some business critical projects that apply SOA have knowledge and experience with SOA;
- There is SOA awareness within most business and IT departments;
- There is SOA awareness throughout the organization;
- There is an organization-wide awareness of the use of application services for internal and external use;
- No opinion.

13. What SOA training does your organization offer?
- There is no SOA training;
- Knowledge and experience about SOA is shared ad hoc;
- Knowledge and experience about SOA is shared as best practices between projects;
- SOA training is centralized at enterprise level and offered in most parts of the organization;
- SOA training is institutionalized and fully integrated into IT training;
- SOA training is offered within and between organizations;
- No opinion.

14. How would you describe the SOA skills in your organization?
- Nobody has SOA skills;
- Some people have SOA skills;
- In some business critical project people have SOA skills;
- In most parts of the organization people have SOA skills;
- People throughout the organization have SOA skills for internal application services;
- People throughout the organization have SOA skills for internal and external application services;
- No opinion.

Business architecture

This aspect describes how service-orientation is applied to the business architecture and how business processes are orchestrated by service composition. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

15. **How is business logic decomposed into basic business activities (business services) in your organization?**
- Business logic has not been decomposed;
- Decomposition of business logic into basic business activities is done in pilot project(s);
- Decomposition of business logic into basic business activities is done within some business critical projects;
- Decomposition of business logic into basic business activities is being standardized throughout the organization;
- Decomposition of business logic into basic business activities is performed consistently in a standard way throughout the organization;
- Decomposition of business logic into basic business activities is performed consistently in a standard way throughout the organization and within business collaborations;
- No opinion.

16. **How are business processes described in basic business activities (business services) in your organization?**
- Business processes are not described in business services;
- Experimentation is done to describe business processes in business services;
- Business processes are described in business services within some business critical projects;
- Describing business processes in business services is standardized throughout the organization;
- Business processes are described in business services consistently in a standard way throughout the organization;
- Business processes are described in business services consistently in a standard way throughout the organization;
- No opinion.

17. **How are business processes orchestrated with application services in your organization?**
- Business processes are constructed from tightly-coupled components;
- Experimentation is done to orchestrate business processes with application services;
- Business process orchestration with application services is based on best practices from experience in other projects;
- Business process orchestration with application services is standardized throughout the organization;
- Business process orchestration with internal application services is performed consistently in a standard way throughout the organization;
- Business process orchestration with internal and external application services is performed consistently in a standard way throughout the organization;
- No opinion.

18. **How would you describe the cooperation between business and IT stakeholders to support business process orchestration with application services in your organization?**

- There is no cooperation between business and IT stakeholders to support business process orchestration with application services;
- Business and IT stakeholders jointly experiment in a pilot project to support business process orchestration with application services;
- There is business and IT cooperation in some business critical projects that orchestrate business processes with application services;
- Business and IT stakeholders are integrating at business unit level to support business process orchestration with application services throughout the organization;
- Business and IT stakeholders have fully integrated to support central application services throughout the organization;
- Business and IT stakeholders have fully integrated to support business process orchestration throughout the organization with internal and external application services;
- No opinion.

Information model

This aspect concerns the modeling of information exchange. This includes the standardization of message formats but also semantic interoperability. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

19. **How are the message formats between application services standardized in your organization?**

- Different message formats and techniques are used to exchange information between individual software applications;
- Standard message formats are tested to communicate between application services within the boundaries of a pilot project;
- Standard message formats are used to communicate between application services in some business critical projects but these standards differ between projects;
- Open standards for message formats are used to communicate between application services in most parts of the organization;
- Open standards for message formats are used to communicate between all application services throughout the organization;
- Open standards for message formats are used to communicate between all internal and external application services;
- No opinion.

20. How is semantic interoperability supported in your organization?
- There are no standards for interpretation of information exchanged between individual software applications;
- Experimentation is done in achieving semantic interoperability between application services in a pilot project;
- There are some standards in line to achieve semantic interoperability between application services of some business critical projects;
- Standards for semantic interoperability between application services are proposed on enterprise level;
- There is semantic operability of information exchanged between internal application services throughout the organization;
- There is semantic operability of information exchanged between internal and external application services;
- No opinion.

21. How are application services described in your organization?
- Application services are not described;
- Application services are described ad hoc;
- There are standards to describe application services but these differ between projects;
- Application services are described in a standard way in most parts of the organization;
- All internal application services are described consistently in a standard way throughout the organization;
- All internal and external application services are described consistently in a standard way;
- No opinion.

22. How are service descriptions stored in your organization?
- Service descriptions are not stored;
- Service descriptions are stored in an experimental service registry or a spreadsheet;
- Service descriptions are stored in different service registries;
- Service descriptions of most application services throughout the organization are stored in one service registry;
- There is one service registry storing service descriptions of all internal application services throughout the organization in a standard way;
- There is one inter-organizational service registry storing service descriptions of internal and external application services in a standard way;
- No opinion.

23. How are application services discovered in your organization?
- Application services cannot be discovered;
- Application services can be discovered in an experimental service registry or spreadsheet and they are promoted ad hoc by informal contact;
- Application services and their characteristics can be discovered in one or more small-scale experimental registries;
- Application services and their characteristics can be discovered in one service registry that covers application services in most parts of the organization;
- There is an extensive service registry to discover and select application services throughout the organization based on their characteristics in a standard way;
- There is an extensive inter-organizational service registry to discover and select internal and external application services based on their characteristics in a standard way;
- No opinion.

24. How would you describe the performance of information governance in your organization?
- The information models are specific for different individual software applications and services are not included in the information model;
- The information model for services is experimental and owned by an IT architect;
- The information models cover different SOA projects and are owned by the SOA-project managers;
- One information model for services is proposed and governed by IT at enterprise level;
- There is one information model governed at enterprise level by the business;
- There is one information model seamlessly governing information exchange between multiple organizations;
- No opinion.

Application architecture

This aspect concerns how service-orientation is applied to software applications and application logic. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

25. How would you describe the scope of application services development in your organization?
- Application services are not developed;
- There are some experimental application services developed ad hoc in pilot project(s);
- Application services are developed within some business critical projects;
- Application services are developed centrally in most parts of the organization;
- Central application services are common throughout the organization;
- Central applications services are offered internally and externally to other business partners and vice versa;
- No opinion.

26. How would you describe the SOA reference architecture used in your organization?
- There is no SOA reference architecture;
- There is informal contact about how to develop application services;
- Best practices are shared between SOA projects;
- A SOA reference architecture emerges at enterprise level that is used in most parts of the organization;
- Central application services throughout the organization follow one standard SOA reference architecture;
- Internal and external services follow an inter-organizational SOA reference architecture for their development;
- No opinion.

27. How would you describe service lifecycle management in your organization?
- There is no management of the service lifecycle;
- Application services are developed and managed ad hoc;
- An initial service lifecycle for application services has been defined;
- Service lifecycle management is performed in a standard way in most parts of the organization;
- Service lifecycle management is institutionalized for internal application services throughout the organization;
- Service lifecycle management is institutionalized for internal and external application services at an inter-organizational level;
- No opinion.

28. How are architectural decisions concerning application architecture with SOA taken and enforced within your organization?

- No architectural decisions are made in relation to services;
- Architectural decisions about service development are made ad hoc;
- Architectural decisions about service development are based on initial policies within projects;
- Architectural decisions about service development in most parts of the organization are made at enterprise level based on policies;
- Architectural decisions about service development throughout the organization are made at enterprise level;
- Architectural decisions about service development for internal and external service development are made at inter-organizational level;
- No opinion.

Operational infrastructure

This aspect concerns the technical infrastructure that supports the runtime service environment. Please choose the description that best fits with the current situation of your organization. If you cannot answer the question, please choose "no opinion".

29. What is the scope of the service messaging framework currently used in your organization?

- There is no framework to support service messaging;
- There is an experimental framework for service messaging;
- There are one or more service messaging frameworks supporting different projects;
- One service messaging framework is centralized at enterprise level and used in most parts of the organization;
- There is one service messaging framework used throughout the organization;
- There is one inter-organizational service messaging framework that support internal and external services;
- No opinion.

30. How is security of messaging integrated within service messaging in your organization?

- There is no framework for security of service messaging;
- Experimentation is done with securing service messaging;
- Secure service messaging is applied in different SOA projects;
- Security of service messaging is integrated into a centralized service messaging framework;
- There is an organizational framework for secure service messaging;
- There is an inter-organizational framework for secure service messaging;
- No opinion.

31. **How would you describe the support for Service Level Agreements (SLAs) for Quality of Service of the operational service infrastructure in your organization?**

- No SLAs are defined for quality of service of the operational service infrastructure;
- Some initial SLAs are defined for quality of service of the operational service infrastructure;
- There are best practices for defining SLAs for quality of service of the operational service infrastructure;
- Defining and monitoring SLAs about quality of service of the operational service infrastructure is standardized at enterprise level;
- Defining and monitoring SLAs about quality of services of the operational service infrastructure is institutionalized at enterprise level;
- Defining and monitoring SLA's about quality of service is institutionalized for the operational inter-organizational service infrastructure;
- No opinion.

Comments

If you have any comments regarding this questionnaire, please put them in the box below.

This is the end of questionnaire. Thank you for filling in this questionnaire.

Appendix B: Acronyms

API	Application Programming Interface
AR	Action Research
CS	Case Study
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol over Secure socket layer
IS	Information System
IT	Information Technology
MEP	Message Exchange Pattern
OSIMM	The Open group Service Integration Maturity Model
PoC	Proof of Concept
QoS	Quality of Service
RPC	Remote Procedure Call
RQ	Research Question
SAAS	Software As A Service
SLA	Service Level Agreement
SOA	Service-Oriented Architecture
SOAP	Originally for 'Simple Object Access Protocol' but this was dropped when updating to SOAP version 1.2
SOE	Service-Oriented Enterprise
TOGAF	The Open Group Architecture Framework
UDDI	Universal Description Discovery and Integration
WS	Web Service
WS-BPEL	Web Services Business Process Execution Language
WSDL	Web Service Description Language
XML	eXtensible Markup Language