Service Oriented Architecture Maturity Models: 
A guide to SOA Adoption?

Fabian Meier
meier.f@gmail.com
Service Oriented Architecture Maturity Models:
A guide to SOA Adoption?

Submitted by Fabian Meier to the University of Skövde, Box 408 S-54128 Skövde, SWEDEN, as a dissertation for the degree of Master of Science (M.Sc.) by examination and dissertation at the School of Humanities and Informatics.

30th August 2006

I certify that all material in this dissertation which is not my own work has been clearly identified and that no material is included for which a degree has previously been conferred on me.

Signed: ___________________________________________________________

Supervisor: Eva Söderström
Examiner: Benkt Wangler
Abstract

To support the difficult process of SOA adoption, SOA maturity models are published. In this thesis a literature analysis introduces the area of SOA maturity models and highlights benefits and criticism. As a result, the combined SOA maturity model (CSOAMM), a model to facilitate the interpretation and comparison of SOA maturity models, is proposed. CSOAMM is a model with 10 levels that describes SOA maturity by combining level characteristics of two recently published maturity models: SIMM and SOAMM. CSOAMM was created by a comparative analysis in combination with a relational content analysis. The model can be used for collaboration between companies and points out that a common understanding of SOA adoption exists.

Keywords: Service Oriented Architecture, SOA, Maturity Model, SIMM, SOAMM, CMMI, Adoption, Introduction.
Acknowledgements

I would like to thank:

- My supervisor Eva Söderström for her valuable guidance, encouragement and support.
- The authors of the SOAMM, Jonathan Bachman, Sheldon Borkin and Sean Kline for their comments.
- Anders Roxell for all discussions about my thesis.
- Björn Streicher, Daniel Teuber and Markus Lotter for commenting my work.
- My opponent Per Backlund for all comments.
- My girlfriend Jana and my parents for their encouragement and support.
- All other master students for accompanying me.
## Contents

1 Introduction .................................................. 1
   1.1 Problem area ........................................... 1
   1.2 Aim and objectives ..................................... 1
   1.3 Research approach ...................................... 2
   1.4 Thesis structure ......................................... 4
   1.5 Target group ............................................. 5

2 Background .................................................. 6
   2.1 SOA basics ............................................... 6
   2.2 Maturity model basics .................................. 10

3 SOA maturity models ........................................ 12
   3.1 Benefits and criticism .................................. 12
      3.1.1 Benefits ............................................. 12
      3.1.2 Criticism .......................................... 13
   3.2 SOA Maturity Model ...................................... 14
      3.2.1 Background .......................................... 14
      3.2.2 Structure ........................................... 15
   3.3 Service Integration Maturity Model .................. 17
      3.3.1 Background .......................................... 17
      3.3.2 Structure ........................................... 17

4 Combined SOA Maturity Model .............................. 20
   4.1 Introduction ............................................. 20
   4.2 Structure ................................................ 20
   4.3 Details ................................................... 21
   4.4 Relation to CMMI ......................................... 23
   4.5 Scenario .................................................. 24
   4.6 Analysis of CSOAMM ...................................... 25

5 Research summary ............................................ 27
   5.1 Research aim revisited .................................. 27
   5.2 Contribution ............................................. 27
   5.3 Validation ............................................... 28

6 Discussion ................................................... 29
   6.1 Research discussion ..................................... 29
   6.2 Related work ............................................. 29
   6.3 Future work .............................................. 30

References ..................................................... 31

Appendix A SOAMM - Characteristics i
Appendix B SIMM - Characteristics iii
Appendix C CSOAMM - Characteristics v
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Work process</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Thesis structure</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Enterprise architecture based on SOA</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>WS triangle</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>SOA Maturity Model - SOAMM</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Service Integration Maturity Model - SIMM</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>Combined SOA Maturity Model - CSOAMM</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>CSOAMM versus CMMI</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>Collaborative vision of company A and B</td>
<td>25</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Planning benefits of SOA maturity models</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Adoption benefits of SOA maturity models</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Criticism of SOA maturity models</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>CSOAMM level 5</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>SOAMM level characteristics 1 and 2</td>
<td>i</td>
</tr>
<tr>
<td>6</td>
<td>SOAMM level characteristics 3 to 5</td>
<td>ii</td>
</tr>
<tr>
<td>7</td>
<td>SIMM level characteristics 1 to 5</td>
<td>iii</td>
</tr>
<tr>
<td>8</td>
<td>SIMM level characteristics 6 and 7</td>
<td>iv</td>
</tr>
<tr>
<td>9</td>
<td>CSOAMM level characteristics -2 to 2</td>
<td>v</td>
</tr>
<tr>
<td>10</td>
<td>CSOAMM level characteristics 3 to 7</td>
<td>vi</td>
</tr>
</tbody>
</table>
List of Abbreviations

BAM Business Activity Monitoring
BPM Business Process Management
CMM Capability Maturity Model
CMMI Capability Maturity Model Integration
CSOAMM Combined SOA Maturity Model
COM Component Object Model
CORBA Common Object Request Broker Architecture
EAI Enterprise Application Integration
EJB Enterprise Java Bean
ESB Enterprise Service Bus
IT Information Technology
QoS Quality of Services
LDAP Lightweight Directory Access Protocol
OWL Ontology Web Language
ROI Return on Investment
SIMM Service Integration Maturity Model
SOA Service Oriented Architecture
SOAMM SOA Maturity Model
SOAP Simple Object Access Protocol
SOI Service Oriented Integration
SAML Security Assertion Markup Language
UDDI Universal Description, Discovery, and Integration
WS Web Service
WS-BPEL Web Service - Business Process Execution Language
WSDL Web Service Description Language
XML Extensible Markup Language
XSLT Extensible Stylesheet Language Transformations
1 Introduction

This section introduces the problem area and states the aim for this thesis. Objectives and corresponding methods are presented. The structure of the thesis is explained and potential target groups named.

1.1 Problem area

Service Oriented Architectures (SOA) is a recent hype topic. Gartner Research predicts that in 2008 SOA will be the basis for 80 percent of development projects (Cearley, Fenn & Plummer 2005). There exist journals like the "International Journal of Web Service Research (IJWS)\(^1\), conferences like the "International Conference on Service-Oriented Computing (ICSOC)\(^2\) and many workshops about SOA. Most articles concern technically related topics like Web Services, but the focus is changing and more and more architecture and business related topics are included. As Web Services are a key enabler for SOA, it is not surprising that academic research so far is focused on Web Services, while SOA itself is covered by fewer research projects.

For many, SOA is a jungle of acronyms where even different definitions of SOA exists. Bernard (2006) writes that it is disappointing that in 2006 still the lack of knowledge and awareness is the reason that adopting SOA is slow, since only 44% of those interested in SOA in 2005 have started pilot projects. Roadmaps are being published to clarify the process of SOA adoption. But as they use different vocabularies, one can assume that they will not help to clarify the SOA confusion. Recently published SOA maturity models with the aim to clarify the SOA confusion offer a parental framework for SOA roadmaps (Bloomberg 2005\(^a\), Arsanjani & Holley 2005\(^c\)). Through the fact that different maturity models are published, a couple of questions arise:

- Are they describing the same levels?
- How do they relate?
- Do SOA maturity models offer a similar way of SOA adoption?
- Does such a common way of SOA adoption exist?

1.2 Aim and objectives

The questions of section 1.1 can be summarised into the following research question: *How do different SOA maturity models guide SOA adoption?* This consequently leads to the aim:

*To develop a model for facilitating interpretation and comparison of SOA maturity models*

A better understanding of SOA maturity models helps to grasp and guide SOA adoption. The aim can be detailed by two groups of objectives:

1. Introduction to SOA maturity models:

   (a) To highlight benefits and criticism of SOA maturity models (adressed in chapter 3.1)

\(^1\)[http://www.idea-group.com/journals/details.asp?id=4138]
\(^2\)[http://www.icsoc.org/]
1.3 Research approach

(b) To introduce two current SOA maturity models (addressed in chapter 3.2 and 3.3)

2. Relation between maturity models:

(a) To show the relation between these two models by creating the Combined SOA Maturity Model (CSOAMM) (addressed in chapter 4)
(b) To show the relation to CMMI (addressed in chapter 4.4)
(c) To create a CSOAMM usage scenario (addressed in chapter 4.5)

1.3 Research approach

Figure 1 shows important research steps and underlying methods. Each step uses results of the steps before, which is illustrated by the arrows in the figure.

![Figure 1: Work process](image)

The thesis uses 3 different approaches:

- **A literature analysis** in chapter 2 to provide the background of SOA and maturity models and in chapter 3 for introducing SOA maturity models, especially SOAMM and SIMM.
- **An argumentative approach** in chapter 4 to create a combined model (CSOAMM) based on SOAMM and SIMM, to map this new model to CMMI and to create a scenario.
- **A validation** of the results is created in chapter 5 by using contacts with the authors of SIMM and SOAMM.

In the following paragraphs, arguments for the suitability of each approach lead to the details of the research process.

**Literature analysis** According to Berndtsson, Hansson, Olsson & Lundell (2002) a literature analysis is a systematic, purposive examination by analysing published sources to a given problem. Webster & Watson (2002) write that a literature review represents the foundation for research in IS. They write that a literature review is an appropriate method for starting a project with a topic of an emerging issue. As SOA maturity models are such an emerging issue, a literature review is an appropriate method and is used to explain SOA maturity models and to point out the benefits and criticism of them.

As introduction to SOA, two books (Erl 2005, Bieberstein, Bose, Fiammante, Jones & Shah 2005) were used to establish a first overview. For maturity models, the paper describing CMMI (Software Engineering Institute 2002) was used.

To establish a more detailed background, a literature analysis was conducted using academic research databases like Elsevier, ACM and IEEE.

---

3 CMMI stands for Capability Maturity Model Integrated and is the most common maturity model in the computer science area. CMMI is explained more in detail in chapter 2.2.
An internet research using www.google.com and www.yahoo.com was conducted to find company whitepapers, e-magazines and blogs. Here, the business view of SOA is more common, as companies have to sell SOA to their customers. A careful interpretation of company papers is necessary as Berndtsson et al. (2002) write that they are often not reviewed and not presenting neutral facts. This applies also to web resources like e-magazines and blogs, which were used, as they provide up-to-date informations, to include positive and critical voices. It must be kept in mind that the reliability of these resources can only be satisfied by the names of the authors and their reliability. For the background of SOA searchwords like "SOA", "Web Services", "SOA Adoption" and "Grid Architecture" were used. For the maturity model background searchwords like "Maturity", "Maturity Model" and "CMM" were used. To find information about SOA maturity models searchwords like "SOA Maturity", "SOA Roadmap", "Service Maturity", "SIMM" and "SOA maturity model" were used. New information supplied by newsfeeds and google alerts was added continuously.

The Service Integration Maturity Model (SIMM) (Arsanjani & Holley 2005) published by IBM, and the SOA Maturity Model (SOAMM) (Sonic Software Corp., AmberPoint Inc., BearingPoint Inc. & Systinet Corp. 2005) were described by using literature published by the authors of the models. These models were chosen since they are subject for standardisation by certain standardisation groups. The Open Group is working on SIMM, and OASIS is working on SOAMM. Summarising lists of level characteristics for each of the two models were created (see Appendix A,B).

**Argumentative Approach**  The argumentative approach can be divided into a comparative analysis and a relational content analysis. According to Abrahamsson, Warsta, Siponen & Ronkainen (2003), their comparative analysis of agile software development methods should provide a better understanding of various properties of methods and help to make judgements in a more informed way. In the research presented in this thesis, the comparative analysis should provide a better understanding of the original maturity levels. Consequently, a comparative analysis is an appropriate method for the aim of facilitating the interpretation and comparison of SOA maturity models.

While the comparative analysis highlights similarities of the two models, the relational content analysis focuses here on the relation between characteristics inside a model (Busch, Maret, Flynn, Kellum, Le, Meyers, Saunders, White, & Palmquist 2005). The combination of both methods leads to the combined model CSOAMM. The comparative analysis was used to connect the characteristic lists of both models (SOAMM, SIMM). The relation content analysis was used to extend the new list of CSOAMM characteristics with characteristics without a clear mapping between the models. The combination of the two models adds more details to the levels and hence provides a better understanding of the maturity levels. Characteristics in one of the levels mapping to two levels in the other model lead to two levels. CSOAMM can be easily extended into a framework describing SOA maturity.

May (2005) uses a similar approach for comparing different software architecture viewpoint models. He uses the IEEE Standard 1471 to create an arbitrary list for each type of element and adds additional elements as identified in the viewpoint models. The results are that viewpoints from different models can cover the software architecture

---

4 http://www.opengroup.org
5 http://www.oasis-open.org
6 see Appendix A and B
7 see Appendix C (included also the link information between SOAMM and SIMM)
domain better than one individual viewpoint model and that different vocabularies in
the models can be compared by a common reference vocabulary.

Since CMMI is a well-known maturity model in the software development area,
it can provide an easy entry to SOA maturity models by using company experiences
with CMMI. CMMI levels were analysed and similar characteristics were mapped to
CSOAMM.

A fictitious scenario was created to illustrate the value of CSOAMM. It serves the
purpose of showing the usefulness of CSOAMM for company collaborations. To show
this, three phases were used: a phase before collaboration, a phase when establishing
the collaboration, and a phase during collaboration. It was necessary to use different
company sizes, a different maturity model and different maturity of SOA adoption.

**Validation** In section 5 the results were summarised and validated. For validation,
the contact with the authors of SIMM and SOAMM were used. The authors were con-
tacted by email in the initial phase of the thesis and asked for their support. Both main
authors of the two models (Ali Arsanjani - SIMM, Jon Bachman - SOAMM) answered
and stated that they were interested in this work and willing to support it. After the
completion of section 3 "SOA maturity models", a copy of this section was send to
both authors to be validated and commented. Unfortunately the authors did not reply.
After finishing section 4 "CSOAMM", all authors including subauthors were contacted
for a validation of interpretation of their model and a comment on CSOAMM. After a
reminder, three out of five authors of SOAMM (Jonathan Bachman, Sheldon Borkin,
Sean Kline) validated the representation of SOAMM and added some information. The
SIMM model could not be validated.

1.4 Thesis structure

By presenting all sections and subsections, figure 2 provides an overview over this
thesis.

![Figure 2: Thesis structure](image)
1.5 Target group

As this thesis is one of the first academic works concerning SOA maturity models, it can be seen as introduction to SOA maturity models. This work can be of interest for all those interested in SOA adoption, such as companies, vendors, consultants and researchers. The main contribution is the combined model, CSOAMM, which can help to understand SOAMM and SIMM. It may help to clarify SOA and especially SOA maturity models. Companies, can for example, discover the benefits of SOA maturity models mentioned in section 3.1. CSOAMM can be especially useful for companies. Companies can use the mapping as communication tool when collaborating with other companies or use it internally to include external knowledge, based on different maturity model. It can help to establish maturity models as a ranking tool, as the different levels can be translated.
2 Background

This section provides the background for this thesis. Here, Service Oriented Architectures (SOA) and Maturity Models are explained.

2.1 SOA basics

To meet the business challenges of the 21st century, SOA initiatives have recently been adopted in many companies (Bieberstein, Bose, Fiammante, Jones & Shah 2005, Bieberstein, Bose, Walker & Lynch 2005). Starting with the definition of SOA, the subparts of the acronym SOA, ”architecture” and ”service-orientation” will be explained. The following subsection shows why SOA is important and then SOA adoption will be explained.

Definition of SOA  Bieberstein, Bose, Fiammante, Jones & Shah (2005, p.4) define SOA in the following way:

*A service-oriented architecture is a framework for integrating business processes and supporting IT infrastructure as secure, standardised components (services) that can be reused and combined to address changing business priorities.*

According to this definition, used throughout this thesis, SOA is a framework and not a technology. By highlighting business processes as business-related and the IT-infrastructure as more technology-related, the scope of SOA is shown. Standardised components as main characteristic of SOA provide the possibility of reusability and interoperability. With SOA it is possible to address changing business priorities as one challenge in the 21st century (Nadler & Tushman 1999). The term ”service-oriented architecture” can be split up in two parts ”architecture” and ”service-orientation”. Both concepts will be explained in the next paragraphs.

Architecture  Pereira & Sousa (2004) propose a method to create an enterprise architecture based on the common Zachman framework (Zachman 1987). They define an enterprise architecture as a framework describing how companies achieve business objectives. It consists of a business architecture, an information architecture, an application architecture and a technical architecture. As the focus in this master thesis is not technical, the more technical architectures such as the information architecture and the technical architecture are excluded. Figure 3 shows the tailored enterprise architecture and will be used for further explanations.

The tailored enterprise architecture includes a more business-oriented business architecture, the upper oval, and a more IT-oriented application architecture, the lower oval. Pereira & Sousa (2004) describe the business architecture as the result of defining business strategies, processes and functional requirements. The application architecture is an architecture focused on developing and implementing applications to fulfill business requirements. It is important to mention that a company often has more than one application architecture (Pereira & Sousa 2004). Erl (2005, p.87) writes that SOA can be applied to the enterprise architecture and the application architecture. In this thesis, the focus is on the enterprise architecture. Figure 3 shows that SOA pro-

---

*This view is consistent with the overall aim of the self-organised enterprise presented in both maturity models discussed in this thesis.*
Service orientation  In an interview on SearchWebServices.com (2006), Erl stated that companies must concentrate on the "service orientation" part in SOA. In his book Erl (2005) writes that service-orientation applies to the enterprise logic. By using the principle of service-orientation, the manner in which the enterprise logic is represented, viewed, modelled and shared will change. The World Wide Web Consortium (W3C) (Booth, Haas, McCabe, Newcomer, Champion, Ferris & Orchard 2004) define a service as:

... an abstract resource that represents a capability of performing tasks that represents a coherent functionality from the point of view of provider entities and requester entities. To be used, a service must be realised by a concrete provider agent.

Thus, a service can be seen as a resource, or in the context of IT as a small piece of software, with the aim to perform a task, which is realised by the provider and used by the requester.

Service-orientation could be understood as a form of realisation of the principle of separation of concerns (Erl 2005). Erl (2005) introduces a service interface layer between business process layer (business architecture) and application layer (application architecture). Services are connecting these two layers or architectures as shown in figure 3. Erl (2005) further decomposes this "SOA layer" into three sublayers: a layer establishing the connection to the business process layer (business service layer), a layer establishing the connection to the application layer (application service layer), and an orchestration layer to manage the services inside the service layer.

To be able to realise the concept of service-orientation, Erl (2005) introduces a set of principles to which services must be associated. The service-orientation principlesootnote{As this is just a short introduction, I recommend to read section 8 in Erl (2005, p.280-p.326) for a deeper explanation of these principles.}
2.1 SOA basics

BACKGROUND

2005, p.291) are that services share a formal contract, have an abstract underlying logic and are reusable, loosely coupled, composable, autonomous, stateless and discoverable. Services must be designed according to these principles and appropriate technology must be chosen.

Web Services  Erl (2005, p.324) states that half of these service principles are naturally supported by Web Services. Web Services naturally support contracts, loose coupling, abstraction and composability. Accordingly, Web Services are a good choice to realise the concept of service-orientation. On the other hand, Curbera, Duftler, Khalaf, Nagy, Mukhi & Weerawarana (2005) write that Web Services do not necessarily mean that SOA principles are being applied. In this thesis SOA is seen as a principle and WS as a technology to help to realise this principle. Web Services are according to W3C (Booth et al. 2004):

... a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL (Chinnici, Moreau, Ryman & Weerawarana 2006)). Other systems interact with the Web Service in a manner prescribed by its description using SOAP (Gudgin, Hadley, Mendelsohn, Moreau & Frystyk Nielsen 2003) messages, typically conveyed using HTTP with an XML serialisation in conjunction with other Web-related standards.

This definition introduces the basic standards in the Web Service area. The most important part is that interfaces in WSDL are machine-processable. Therefore it is possible, in combination with SOAP messages, to create an autonomous network of services.

To realise SOA by Web Services one distinguishes between service producer, who creates a service, and service consumer who uses this service. A service registry acts as mediator to discover services (Huhns & Singh 2005, Massuthe, Reisig & Schmidt 2005, Tibco Software Inc. 2005, Gao, Ma, Yen, Bastani & Wei-Tek 2005, Arsanjani, Curbera & Mukhi 2004). Figure 4 explains the relations between service producer, services registry and service consumer. Web Services will be created and then stored in a repository (Tibco Software Inc. 2005, Massuthe et al. 2005). Additionally, they get published in a registry (Bussler 2004, Huhns & Singh 2005) using e.g. UDDI (Clement, Hately, von Riegen & Rogers 2004). To find and discover a service, the service consumer sends a message to the registry and then has to select a service. After a service is chosen, the registry provides information, in form of a WSDL document, how the service can be accessed. It is important to mention that in a SOA, Quality of Service (QoS) requirements are included in the contract (Curbera et al. 2005). In addition, semantic approaches like OWL (Maximilien & Singh 2004) or directory services like LDAP (Willmott, Ronsdorf & Krempels 2005) exist to discover services.

The Enterprise Service Bus (ESB) is responsible for routing, controlling and transforming the communication between provider and consumer (Hermann & Kessler 2005, Tibco Software Inc. 2005). It can be understood as a layer, which is adding value by providing various functions, so far provided by the requestors and providers (Bieberstein, Bose, Walker & Lynch 2005).

The so-called "WS-*” Standards can help to establish a SOA and extend the basic SOA described before. These standards include messaging standards like WS-Security, WS-ReliableMessaging, WS-Acknowledgement and WS-Addressing.

\[10\] Further explanations can be found for example in Erl (2005) part 2

**Why SOA**  In the foreword to Bieberstein, Bose, Fiammante, Jones & Shah (2005), Daniel Sabbah and Jason Weisser write that the SOA paradigm represents an identifiable, market-analyst-certified solution to the enterprise’s business and IT challenges. A survey by the Aberdeen Group (2005) researched this more in detail and found out that most companies are implementing SOA to develop new capabilities or products. It followed the reasons of reuse, management of the complex IT and alignment with business. In another study (Austvold & Carter 2005), the most expected benefits from SOA were a faster and more flexible reconfiguration of business processes (48%) and the decrease of operational costs of information technology (28%).

The list of possible benefits from SOA can be easily extended. In a comment to this thesis, Jonathan Bachman (Sonic Software) highlights that it is important to remember that SOA has many levels of benefits and it would be too easy to just mention, for example, agility. He writes that one purpose of SOAMM (a SOA maturity model discussed in this thesis) is to show what benefits could be delivered in each maturity level.

**SOA adoption**  The adoption of service-oriented principles is a complex process with the aim of creating a service oriented architecture. It consists of many projects and has a scope of many years. When adopting SOA, different areas inside a company are affected (Pereira & Sousa 2004): organisational structure, people, workflow processes and technologies. Bieberstein, Bose, Walker & Lynch (2005) write consequently that the necessary organisational redesign - including culture and individual behaviour - must be well-planned.

Business process management (BPM) is closely related to SOA as it is common to optimise business processes while adopting SOA (Woodley & Gagnon 2005, Herr, Bath & Koschel 2004). Adopting SOA is not easy and a lot of challenges arise, e.g. immature standards and insufficient knowledge (Tilley, Gerdes, Hamilton, Huang, Müller, Smith & Wong 2004, Söderström 2005).

Cherbakov, Galambos, Harishankar, Kalyana & Rackham (2005) state that it is a big question how to incrementally transform the IT system in concordance with the
business transformation. Different roadmaps (e.g. Traventec Ltd. 2004, Bieberstein, Bose, Fiammante, Jones & Shah 2005, Bennett 2005, Schmelzer & Bloomberg 2005) try to describe the whole process of SOA adaptation. It is important to mention that all these proposed roadmaps cannot replace a company specific roadmap. However, they can help to plan and to create an individual roadmap. SOA maturity models, as being the topic of this thesis, should provide a parental framework for these roadmaps and an overview of the whole adoption process.

2.2 Maturity model basics

According to the Software Engineering Institute (2002), the Capability Maturity Model (CMM), as the most known maturity model, contains essential elements of effective processes. CMM should provide a guideline to stable, capable and mature processes. It should guide how to improve the management of development, acquisition and maintenance of products and services. However, they state that normally, a one-to-one mapping to organisational processes is not possible (Software Engineering Institute 2002).

The roots of CMM date to 1990 when the Software Engineering Institute (SEI) at the Carnegie Mellon University proposed a number of software process improvement areas to the US Department of Defence (Saiedian & Kuzara 1995). Later on, this framework was used by the Department of Defence as a quality criterion. Pyster (2005) describes it as a relatively complete and widely used framework for process improvement in software development. In the year 2002, a new version of the CMM, now called Capability Maturity Model Integration (CMMI), was proposed (Software Engineering Institute 2002). The focus changed from pure software practices towards the integration of systems and software practices.

Maturity levels are defined as evolutionary plateaus of process improvement, which should help to predict the future performance of an organisation by describing the range of expected results (Software Engineering Institute 2002). The CMMI defines six capability levels in the continuous form (Software Engineering Institute 2002):

0. Incomplete
1. Performed
2. Managed
3. Defined
4. Quantitatively Managed
5. Optimizing

The roots of maturity approaches lie in quality management. Crosby (1979) introduced the first maturity model named “Quality Management Maturity Grid”. Since this first maturity model, many models were introduced in different areas. There exist maturity models dealing for example with project management (Kerzner 2005), software maintenance (April, Hayes, Abran & Dumke 2005), business processes (Rosemann & de Bruin 2005) or outsourcing (Adelakun 2004). Maturity is not only connected to processes; there are maturity models that do not describe processes. Maturity models are also able to explain the maturity of an architecture (van der Raadt, Hoorn & van Vliet 2005). The maturity of an architecture describes the ability to manage development, implementation and maintenance of an enterprise architecture (van der Raadt et al. 2005). Different architecture maturity models exists e.g., the "Advancing Enterprise Architecture Maturity” (Ramchandan, Ghosh, Miller, Przysucha & Tiemann 2005) created by the American Council for Technology/Industry Advisory Council (ACT/IAC) or
the "IT Architecture Capability Maturity Model" (ACMM) (Department of Commerce (DoC) 2003).

Nowadays, Service Oriented Architectures is a new trend in IT architectures and consequently special maturity models (Sonic Software Corp. et al. 2005, Arsanjani & Holley 2005c) are published to describe the maturity of SOA. This thesis will describe SOAMM (Sonic Software Corp. et al. 2005) and SIMM (Arsanjani & Holley 2005c). Further, a combined SOA maturity model based on these two models is created.
3 SOA Maturity models

This section is an introduction to SOA maturity models, where "SOA maturity model" stands for the area of maturity models dealing with SOA. SOA maturity models provide an abstract overview of SOA adoption by characterising evolutionary levels.

Following an introduction with overall benefits and criticism, two SOA maturity models will be introduced and explained.

3.1 Benefits and criticism

It is important to keep in mind that SOA maturity models were published recently and this section is based only on company whitepapers, webmagazines and blogs. Consequently the benefits and criticism may not be neutral and may reflect the opinion of the author or company. This section still provides a good overview as the literature used cover most of the available literature published at the time the thesis is written.

3.1.1 Benefits

Through analysing the literature, it is possible to distinguish between two main groups of SOA Maturity Model benefits: benefits concerning the planning phase before starting SOA adoption and benefits during the adoption. The references of this section are summarised in table 1 and 2.

Planning phase benefits First of all SOA maturity models can be used to create and communicate a common definition of SOA inside a company. Whybrow(ed.) (2005) writes that Maturity models can help to clarify vague definitions of SOA and Book(ed.) (2005) speaks of a tool to educate business managers. Book(ed.) (2005) cites Jonathan Bachmann (Sonic Software), who says that SOA maturity models are models for understanding SOA.

<table>
<thead>
<tr>
<th>Benefits</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td>Sonic Software Corp. et al. (2005); Wainwright (2005); Bachman (2006); Groves (2006); Book(ed.) (2005)</td>
</tr>
<tr>
<td>Identify needs</td>
<td>Sonic Software Corp. et al. (2005); Bloomberg (2005b); Stamper(ed.) (2006)</td>
</tr>
<tr>
<td>Business impact</td>
<td>Sonic Software Corp. et al. (2005); Wainwright (2005); Frye (2006); Meehan (2005); Stamper(ed.) (2005); Bachman (2005); Book(ed.) (2005); Bloomberg (2005b); Sprott (2005a); Veryard (2005)</td>
</tr>
</tbody>
</table>

Table 1: Planning benefits of SOA maturity models

A long-term SOA vision with ultimate objectives and management goals can be established by the use of SOA maturity models. The specific technical and organisational needs can be evaluated, business benefits can be discovered and investments can be planned. By controlling all these factors, the models can help companies with their risk management. SOA maturity models can assist by providing a framework to support discussions about SOA plans.

Adoption benefits The most often mentioned benefit of SOA maturity models is that they can help to guide SOA adoption. This argument especially supports the research question of this thesis: "how different SOA maturity models guide SOA adoption".
Meehan (2005) cite Jonathan Bachman (Sonic Software), saying that people are looking for leadership on how to deal with SOA problems and need to be taught that SOA does not have “to be swallowed like a horse pill”. Wainewright (2005) speaks of SOA maturity models as a solution for companies, which have no clear idea on how to progress with SOA adoption. A maturity model can provide guidance by offering common capabilities which are applicable to every enterprise. Veryard (2005) highlights the parental character of maturity models in comparison to supportive roadmaps. Bloomberg (2005b) speaks of SOA maturity models as an approach to adopt SOA faster and with lower risk. Mittal (2005) describes SOA maturity models as an attempt to help organisations to define their own architectural guidelines. Sprott (2005a) talks about maturity models as a framework to develop company specific maturity models and roadmaps. SOA maturity models help to coordinate the different paths to SOA inside a company or across company borders (Sonic Software Corp. et al. 2005).

### Benefits

<table>
<thead>
<tr>
<th>Benefits</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guidance</td>
<td>Sonic Software Corp. et al. (2005); Freivald &amp; Kuhbock (2005); Frye (2006)</td>
</tr>
<tr>
<td></td>
<td>Meehan (2005); Bloomberg (2005b); Stamper(ed.) (2006)</td>
</tr>
<tr>
<td></td>
<td>Mittal (2005); Linthicum (2004); Sprott (2005a)</td>
</tr>
<tr>
<td></td>
<td>Wainewright (2005); Bachman (2005); Groves (2006)</td>
</tr>
<tr>
<td></td>
<td>Whybrow(ed.) (2005); Book(ed.) (2005); Veryard (2005)</td>
</tr>
<tr>
<td>Correlate different paths</td>
<td>Sonic Software Corp. et al. (2005); Arsanjani &amp; Holley (2005c)</td>
</tr>
<tr>
<td>Key Practices</td>
<td>Sonic Software Corp. et al. (2005); Book(ed.) (2005)</td>
</tr>
<tr>
<td>Well defined architecture</td>
<td>Arsanjani &amp; Holley (2005c); Frye (2006)</td>
</tr>
<tr>
<td>Measure Progress</td>
<td>Sonic Software Corp. et al. (2005); Frye (2006); Bloomberg (2005b)</td>
</tr>
<tr>
<td></td>
<td>Griffin (2005); Mittal (2005); Linthicum (2004)</td>
</tr>
<tr>
<td></td>
<td>Sprott (2005a); Groves (2006); Whybrow(ed.) (2005)</td>
</tr>
<tr>
<td></td>
<td>Book(ed.) (2005)</td>
</tr>
</tbody>
</table>

Table 2: Adoption benefits of SOA maturity models

A SOA maturity model can be understood as collection of key practises to achieve optimal flexibility and a well defined architecture (Book(ed.) 2005). SOA maturity models can be used to control and to measure the progress of SOA adoption (Frye 2006). Overall, SOA maturity models provide a basis for communication. They can be seen as tool for vendors to demonstrate product placements.

As shown, SOA maturity models are a management tool to get an overview over SOA and to help during the whole process of SOA adoption as a very abstract but easily understandable model.

### Criticism

As there is a lot of support for SOA maturity models, there are also doubtful voices. The references of this section are summarised in table.

The most mentioned criticism is, that the models were published too early. Freivald & Kuhbock (2005) argues that SOA is not even universally defined and so it is too early to publish a maturity model. Without an agreement on the scope of SOA, it is impossible to measure maturity according to Evdemon (2006). Matt Quinn (Tibco), cited by Aslett (2006), explains that they have a lot of customers just "using all standards under the sun because they don’t know better“. He argues that around three years of incubation and innovation time is needed to create best practices and formal methodologies. David Clarke (Cape Clear Software Inc.) argues with the same reason that there is no final process of SOA adoption yet and so its too early for such models (Aslett 2006). Wainewright (2005) supports this argument by arguing that everybody is viewing the higher maturity levels still in distance and not clear.
3.2 SOA Maturity Model

Another important criticism is that the models have a too technical perspective. Maltby (Deloitte) says, that it is a danger that companies focus on technological maturity because they will measure themselves by enabled Web-Services, which is the wrong approach (Aslett 2006). Sprott (2005a) also argues, that these models only present one part of a broader enterprise maturity model. Schmelzer (ZapThink) highlights that these models should only be called service maturity model as they are only describing the maturity of services and not the maturity of the architecture (Frye 2006).

Sprott (2005a) identifies as a danger that the models encourage a delayed introduction of an enterprise architecture. Sen (2006) criticises missing SOA governance guidelines in the models. Also, it is mentioned that the models, especially as they are published by vendors, are just product selling instruments and that vendors are the wrong group to publish such models (Veryard 2005).

As demonstrated by this analysis of websites and (e-)magazines the value of SOA maturity models is not without doubts. But it is important to highlight that the most criticism is that the models are published too early. This is not a criticism on the value of SOA maturity models itself. The second main criticism, that the models could have a too technical perspective, must be kept in mind when using SOA maturity models.

### 3.2 SOA Maturity Model

In this thesis, "SOA maturity model" stands for the area of maturity models explaining SOA. On the other hand, the capitalised expression “SOA Maturity Model (SOAMM)” instead stands for the model published by Sonic Software Corp. et al. (2005) and will be explained in this section. By using SOAMM level 3a - Business Services - the model will be exemplified. If not otherwise stated, the information in this section is based on the company whitepaper (Sonic Software Corp. et al. 2005).

#### 3.2.1 Background

The model with the simple name "SOA Maturity Model” was published in October 2005 by four vendors of SOA solutions: Sonic Software Inc., Bearingpoint, Systinet and AmberPoint. To show perspectives and interests of the authors of SOAMM, it is important to provide their background. The authors are:

- Jonathan Bachman, senior director of product marketing at Sonic Software Inc, a vendor for a Enterprise Service Bus and for Orchestration Servers. He stated that he worked closely with Dr. Borkin from Progress Software the parent company of Sonic Software.
- Daniel Ng, Senior Manager Integration Services at BearingPoint - a consulting company specialised in management consulting and systems integration.
• Sean Kline, Director of Product Marketing at Systinet, a company covering the topics SOA governance and business service lifecycle management by their products e.g., Systinet 2 which includes the Systinet Registry.
• Ed Horst, Vice President of Product Strategy at Amberpoint, a company specialised in SOA visibility, management and security.

The authors created the model using the feedback of around 2000 architects and developers in 2004/2005 at Sonic’s worldwide Architect Forum events. They also used industry analyst reports which are showing successful paths to SOA. The authors see the model as a mind-set and a methodology to prepare a company for successful SOA adoption. It should provide guidance to set a SOA vision and to measure the progress. The model was improved and tested during a 10-city roadshow, where it was presented to the audience of a series of ”Management Forums”. In a comment to this thesis, Sean Kline (Systinet) stated that they created the model more with the purpose to stimulate the dialogue than to be prescriptive. Hence, there are certainly many areas missing, like, for example, application delivery.

3.2.2 Structure

The SOA Maturity Model consists of 5 maturity levels, as shown in figure 5. For each level the authors define the most important benefits. Further, they provide an exemplary architecture to show the usage of technology.

![Figure 5: SOA Maturity Model - SOAMM (Sonic Software Corp. et al. 2005)](image)

The first maturity level "Initial Service" leads mainly to new functionality and can be seen as the initial learning phase. In the exemplary architecture, the authors introduce an Enterprise Service Bus (ESB) as middleware to connect services between different applications/services using service interfaces. Further, a Service Registry as a central point of references and a Service Level Management Service as service to monitor the performance of Web Services are introduced. Sean Kline (from Systinet) has given comments on this thesis and highlights the importance of a registry at the intersection between the run-time environment and other areas of an enterprise SOA.

With new functionality, mainly applied in multiple pilot projects, level two "Architected Services" can be reached. The aim is reusability of services and to define standards for the enterprise SOA. In the scope of multiple integrated applications, the
prime business benefits are IT cost reduction and control. The architecture is extended by a Service and Policy Repository to provide central storage of services and the possibility to handle SOA Governance using policies and service definitions. Further, a single Sign-On Service and an Exception Management Service are introduced. In this level, the mediating function of the ESB, to transform XML messages by using XSLT, is used. Sean Kline (Systinet) adds in a comment to this thesis that he sees change management, impact analysis and the formalisation of contracts at the business service level as important issues of this level. He highlights that the output of this level should be a model of SOA including interrelationships between service, their subcomponents, policies and consumers.

From level three it is possible to reach level four by two strategies: 3a and 3b. While 3a is focused on the improvement of internal business processes, 3b is focused on the improvement of processes with external partners. As an example for the other levels, SOAMM level 3a "Business Services" will be explained here more in detail. Level 3a is chosen because this level is probably the level most companies today are aiming for. The main focus in level 3a is the connection between business and technology. The authors list prime business benefits, scope, key goals, key practices, critical technology success factors, critical people, organisational success factors and the relevant standards for every level.

The authors mention as prime business benefit that business processes can be changed quickly and effectively. The scope is business processes across business units or enterprises. One of the key goals is to support complete business processes. In this level, business logic and application logic melt together to a service logic. Another key practice is to specify policies about how to use SOA for creating or modifying business processes. As WS-BPEL can be used to specify executable business processes and to orchestrate them, this standard is the most important standard in this maturity level. One technical success factor is, for example, event-driven processes. An organisational success factor is the IT partnership with the business. In the architecture example, an Orchestration Service is introduced to manage long-running processes which connects the architecture to business process management (BPM).

As the focus of level 3b is the collaboration with other companies, a collaborative service is introduced to connect internal services with external services. This service needs two interfaces, an internal and an external one and transforms the messages between the internal and external interface. Standards like RosettaNet\footnote{http://www.rosettanet.org} define external messages.

Level 4, "Measured Business Services", can be reached by strategy 3a, 3b or both, depending on the company vision of SOA. At this level, the performance of services can be measured in real-time and so the overall benefit is that the business can be transformed from a reactive into a real-time business. The exemplary architecture introduces a service to provide real-time event stream processing. In this service, rules and an event database are used to control the triggering of messages to services. A service responsible for business activity monitoring (BAM) is introduced to provide monitoring functions for management.

Level 5, "Optimized Business Services", provides automation in business processes. The business can be optimised by reacting and responding automatically using event-driven automation. Hence, this level provides the possibility of a self-organised enterprise. The architecture is an "enterprise nervous system", which is organised on its own, reacting to actions according to rules to optimise business goals, for example pricing...
Jon Bachman explains in a comment to this thesis, that if a company wants to optimise their business, they need also a level 5 capable team and infrastructure. This shows that through SOA, the connection between business and IT is an essential success factor.

### 3.3 Service Integration Maturity Model

This section describes the Service Integration Model (SIMM) and is based on several references (Arsanjani & Holley 2005c, Arsanjani 2005, Meehan 2006, Arsanjani 2002, Falkl 2005, Dueermeyer 2005). The structure of this section is identical with the SOAMM section. Hence, an introduction of the authors will lead to the description of the model.

#### 3.3.1 Background

The Service Integration Model (SIMM) was published by Ali Arsanjani and Kerrie Holley at 30th September 2005 on IBM Developerworks where IBM experts publish tutorials, sample code, standards, and other resources to assist software developers. Similar articles were published later in different online journals (e.g. Arsanjani & Holley 2005a, Arsanjani & Holley 2005b, Arsanjani & Holley 2005d). The authors have a broad knowledge in the area of SOA as illustrated in the following list, which describes their background:

- Ali Arsanjani (Ph.D.) is Senior Technical Staff Member\(^\text{12}\) of IBM and Chief Architect of the SOA and Web Services Center of Excellence in IBM Global Services. Further, he is adjunct Assistant Professor for Computer Science at the Maharishi University of Management.
- Kerrie Holley (J.D.) is Distinguished Engineer\(^\text{13}\) at IBM Global Services. Holley is the CTO for IBM’s SOA and Web Services Center of Excellence and Chief Architect in Application Innovation Services (AIS) and in e-business Integrations Solutions.

Together they organised for example the workshop "Service-oriented Architecture and Web Services Best Practices and Patterns” at the ACM Sigplan Conference for Object-Oriented Programming, Systems, Languages and Applications in 2005 (OOPSLA). The authors introduce the model by comparing SOA introduction with an aeroplane landing, where the pilot gets landing patterns like directions, approaches and coordinates. The Service Integration Maturity Model has a similar aim - to support the gradual process of SOA adoption by showing typical approaches. It is important to mention that the model describes service integration maturity and not SOA maturity itself.

#### 3.3.2 Structure

The SIMM consists of seven maturity levels presented in the left part of figure\(^\text{6}\) while the corresponding integration state is shown on the right side.

---

\(^{12}\)STSM is a title given to senior engineers, programmers and scientists in a non-certified profession. STSMs have demonstrated sustained technical achievement and are recognised inside IBM as experts in their field. An STSM contributes intellectual capital such as patents and publications, provides technical leadership on complex projects and often represents his or her business unit in corporate and high-level task forces. (www.ibm.com)

\(^{13}\)A DE is a key technical consultant in a deep area of expertise or across multiple areas. He or she is a strategist, shaping business decisions and processes. DEs also provide leadership for technical disciplines, competencies and professions, and develop architectures, standards, and tools. (www.ibm.com)

---

17
In level one, "Silo", the form of integration is data integration. Hard coded silos, with embedded and often inaccessible and concentrated functionality are common in this level. High performance systems with few changes are dominating and the question should be asked if the different systems or silos have similarities. This is the start of proprietary and ad-hoc integration.

In level two, "Integrated", the form of integration is application integration. This level can be seen as a form of Enterprise Application Integration (EAI). The information flow is getting harmonised and coordinated through message-oriented EAI approaches. The company has moved from structured methods to object-oriented methods and from a monolithic architecture to a layered architecture, where existing functionality can be exposed and so easily accessed. Still, however, proprietary connections and integration points are used and point-to-point exposure is dominating.

In level three, "Componentized", the form of integration is functional integration. Methods are changing to component-based development and the major parts of applications consist of components and no longer of modules as in the levels before. Hence, we can speak of a component architecture. Components and services across product lines are identified and legacy transformation and renovation methods are used to create systems with clear component boundaries and scope. A product management perspective is used to identify how products interoperate. With this information, it is possible to identify important services. The functionality is exposed in a modular way and the communication between components is realised by interfaces and contracts. This is the hard-wired version of SOA based on CORBA and COM, for example.

Services are connected to business needs, and competition considerations influence the decision of which services are opened for third party or business partner invocation. This level reduces the gap between business and IT and starts the process to change the corporate culture to support SOA principles.

In level four, "Simple Services", the form of integration is process integration. This is an early phase of SOA, where first web-services will be published internally or externally and the business view of IT changes from function-oriented to service-oriented. Consequently, service modeling methods are used. IBM calls this approach,
where business functions are integrated by services, service oriented integration (SOI). Michael Liebow, vice president of Web Services and SOA for IBM Business Consulting Services explains in an interview (Meehan 2006) that most of the companies IBM is working with try to achieve this level and that the majority of organisations is situated in the lower levels. In level 4, proprietary protocols and glue code are replaced by standard-based protocols. The interaction is moving from point-to-point connections towards interaction based on service descriptions published by each system. An Enterprise Service Bus (ESB) helps to integrate services across multiple applications inside and outside the enterprise.

In level five, "Composite Services", the form of integration is supply-chain integration. The focus lies on value chains and how they are supported by services. Hence, the connection between business processes and services is the most important issue. This level is the first real SOA level where services can form a contract and on-demand interaction is realised. An enterprise-wide IT transformation leads to composite applications consisting of services. Integration across business functions through the whole enterprise is guaranteed by an architected implementation. According to Liebow, real leaders in the industries have reached level 5 with some areas of their organisation.

In level six, "Virtualized services", the form of integration is virtual infrastructure. By decoupling services from applications, virtualized services are created. The architecture has changed towards a grid-enabled SOA, completely technology neutral. Management, monitoring and events are handled by separate services. In this level, new business models are deployed and old business models are transformed. The infrastructure can be finely tuned with the overall goal of on demand business transformations.

In level seven, "Dynamically reconfigurable services", the form of integration is eco-system integration. At this level, the overall aim of an on demand business is reached. An automatically reconfigurable architecture is introduced. Now the system can compose services or applications at run-time, based on policy descriptions. The modeling is grammar modeling and the infrastructure is based on a dynamic sense and response concept. According to Liebow (Meehan 2006), no organisation has yet reached levels six or seven.
4 Combined SOA Maturity Model

This section introduces the combined SOA maturity model (CSOAMM). A general introduction leads to the model structure and a more detailed description. In subsection 4.4, a mapping between CSOAMM and CMMI is presented and in subsection 4.5, a scenario to exemplify the usage of CSOAMM is shown. The final subsection presents a critical analysis of CSOAMM.

4.1 Introduction

Both SOAMM and SIMM were published in the autumn of 2005. Companies already use them or will start doing so soon. Especially companies, which are using consultancy or software of one of the authoring companies will be the first to use them. Since not only one model is on the market and proposed for standardisation, it can be assumed that different companies will use different models. This thesis addresses this issue by the aim: "to develop a model for facilitating interpretation and comparison of SOA maturity models". This section proposes a combined model (Combined SOA Maturity Model - CSOAMM) (see figure 7) as a tool to connect SIMM and SOAMM. It was not created to be a maturity model, nor as a check-list for a mature SOA. CSOAMM was created for a better understanding of SOAMM and SIMM and is intended to be used as a translation tool between these models.

<table>
<thead>
<tr>
<th>SIMM</th>
<th>CSOAMM</th>
<th>SOAMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Dynamically reconfigurable services</td>
<td>7. Dynamically reconfigurable services</td>
<td>5. Optimized Business Services</td>
</tr>
<tr>
<td>5. Composite Services</td>
<td>5. Internal and external Services</td>
<td>3a. Business Services</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3b. Collaborative Services</td>
</tr>
<tr>
<td>4. Architected Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Institutionalisation</td>
<td>2. Architected Services</td>
<td></td>
</tr>
<tr>
<td>2. First published WS</td>
<td></td>
<td>1. Initial</td>
</tr>
<tr>
<td>1. Componentized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0. Components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Integrated</td>
<td>-1. Integrated</td>
<td></td>
</tr>
<tr>
<td>1. Silo</td>
<td>-2. Silo</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Combined SOA Maturity Model - CSOAMM

CSOAMM was created by a comparison and mapping of maturity level characteristics. To create a suitable mapping, it was necessary to split some levels.

4.2 Structure

The CSOAMM consists of 10 levels numbered from -2 to 7. The size of the boxes in figure 7 does not indicate level importance. SIMM describes service maturity and covers a broader field than SOAMM, which describes only SOA maturity. This affects the interpretation and has the consequence that SOAMM starts on a higher level. A mapping of the lower levels is not possible and the mapping between the models must start on a higher level. Components, e.g. based on Enterprise Java Beans (EJB), are

---

14 see Appendix C for details
assumed to present a level of initial SOA\textsuperscript{15} To show this starting point of service oriented thinking, CSOAMM has a clear boarder on level 0. CSOAMM levels -2 and -1 represent the corresponding levels in SIMM. Due to the fact that SOAMM is more focused on Web Services, the first SOAMM level "Initial" maps to level 1 "First WS Tests" of CSOAMM. CSOAMM\textsuperscript{16} describes the whole process of SOA adoption starting with level 0 "Components" to level 7 "Dynamically reconfigurable services" as a combination of level characteristics of SIMM and SOAMM.

4.3 Details

As CSOAMM level -2 "Silo" and -1 "Integrated" are identical with SIMM level 1 and 2, and will hence not be further explained in this section. Level 5 exemplifies the combination process.

0. Components  The characteristic of this level is that components are identified and functionality is modular exposed. The communication is based on contracts and interfaces.

1. Technology Tests  The first services are identified and corresponding service definitions are created. Decisions, which services should be exposed are made and the corporate culture starts to adopt service oriented principles. It is possible to describe this level as initial learning phase, where developers learn to create services through pilot projects. A hard-wired version of SOA, based on CORBA or COM, is created using legacy transformation and renovation to establish legacy integration.

2. Published WS  The first Web Services are published and the business has adopted a service-oriented view of IT. SOA is integrated into development methodologies, and technology standards are specified. Proprietary protocols and glue code is replaced by services. This is the first level where SOA technology is applied to realise immediate organisational needs. New functionality is created. Processes get integrated using service description-based interactions. An Enterprise Service Bus (ESB) and a service registry is established to create the technical backbone of the architecture. In this level, initial ROI measurements are defined and applied to quantify costs, time and business benefits of the pilot projects.

3. Institutionalisation  SOA gets integrated into the organisation-wide development process. Through competency centres and organisation wide service-orientation training, SOA gets institutionalised. Business functions get integrated by services and governance based on policies. Service definitions are stored in a service and policy repository. Transformation services of the ESB help to support heterogeneity and distributed systems. Further, exception management is introduced as new service. This level has the aim to get control of SOA and reduce costs through integration.

\textsuperscript{15}An ongoing discussion in the research area is, if SOA must be based on Web Services. Both maturity models assume that Web Services are the dominating technology to proceed towards the highest maturity level.

\textsuperscript{16}A summarising table is presented in appendix C.
4. **Architected Services**  In this level, the control of SOA is increased and SOA is now a carefully architected implementation with multiple integrated applications and a single sign-on. Now, it is possible to prove the return of investment for the use of standard technology.

5. **Internal and external Services**  In the description of SOAMM (section 3.2) level 3a is used to exemplify the details of the model. The corresponding CSOAMM level 5 ”Internal Services” is used to explain the combination process. Therefore, the description will be more detailed here. This level is characterised by the fact, that whole business processes are mapped to SOA. The gap between business and IT is significantly reduced and all business functions are now connected to services. Consequently, an integration across business functions exists. There is no difference anymore between Business Process Management and Service Process Management. Services form contracts and the interaction is on-demand. This level is the final stage of an enterprise-wide IT transformation leading to composite services. Composite services are managed by an orchestration service, which is supporting long-running business processes. Policies for creation and modification of business processes are specified. Later in this level services get published externally. The connection between internal and external services creates the possibility to extend business processes to external organisations and hence whole supply-chains can be integrated. In this later stage of the level, standards like RosettaNet\(^{17}\) and ebXML\(^{18}\) get important as well as the implementation of cross-enterprise security. Returns on investment from reuse and responsiveness to changes can be proven in this level. A summary of level characteristics is shown in table 4, where ”I” in parentheses refers to characteristics of SIMM and ”S” to SOAMM characteristics.

The mapping of this level is based on the characteristics ”Support full business processes” from SOAMM which could be mapped to ”Integration across business functions” from SIMM. The characteristic to connect business processes and services is mentioned by both. SIMM is characterising their level by supply-chain integration, while SOAMM is speaking of external service enablement. Further, the description of business services (SOAMM) and composite services (SIMM) are similar.

<table>
<thead>
<tr>
<th>Number</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Support full business processes (S)</td>
</tr>
<tr>
<td>2.</td>
<td>Integration across business functions (I)</td>
</tr>
<tr>
<td>3.</td>
<td>Connection between business processes and services (I,S)</td>
</tr>
<tr>
<td>4.</td>
<td>Supply chain integration (I)</td>
</tr>
<tr>
<td>5.</td>
<td>External service enablement (S)</td>
</tr>
<tr>
<td>6.</td>
<td>Business Process Management (S)</td>
</tr>
<tr>
<td>7.</td>
<td>Specify policies for creation or modification of business processes (S)</td>
</tr>
<tr>
<td>8.</td>
<td>Services form a contract and on demand interactions (I)</td>
</tr>
<tr>
<td>9.</td>
<td>Enterprise wide IT transformation(composite applications) (I)</td>
</tr>
<tr>
<td>10.</td>
<td>Orchestration Service to manage long running processes (S)</td>
</tr>
<tr>
<td>11.</td>
<td>Prove return from reuse and responsiveness to change (S)</td>
</tr>
<tr>
<td>12.</td>
<td>Connect internal services with external services (S)</td>
</tr>
<tr>
<td>13.</td>
<td>Specify policies for SOA with business and trading partners (S)</td>
</tr>
<tr>
<td>14.</td>
<td>Extend business processes to external organisations (S)</td>
</tr>
<tr>
<td>15.</td>
<td>RosettaNet, ebXML (S)</td>
</tr>
<tr>
<td>16.</td>
<td>Implement cross enterprise security (S)</td>
</tr>
</tbody>
</table>

Table 4: CSOAMM level 5

A strong logical connection exists between characteristics 1 and 2, in 3 as well as

\(^{17}\)www.rosettanet.org
\(^{18}\)www.ebxml.org
between 4 and 5. The characteristics 6-16 must be connected on a weaker basis as they only relate to each other or to the main characteristics.

6. Measured Services  After extending business processes to external organisations, it is more important to transform the business from a reactive to a real-time business. To reach this it is necessary to decouple the services from the underlying applications, leading to virtualised services. The architecture is based on a grid-enabled and technology-neutral virtual infrastructure. It is necessary to fine-tune the infrastructure and to transform old business models or even create new models. Ongoing business process evaluation and reengineering is necessary. This is done by analysing business processes to define business-oriented performance metrics. Performance measurement can be done in real-time through a Business Activity Monitoring Service (BAM). In this level, management, monitoring and event handling of the architecture is based on separate services.

7. Dynamic Architecture  This is the final level with a dynamic architecture based on composite services created at runtime. An event-driven, automatically reconfigurable architecture is handling automated business processes. Based on policies and grammars, the architecture is acting according to rules to optimise business goals. In this level, it is possible to speak of SOA as an enterprise nervous system with a dynamic sense and response concept, which is leading towards a self-organised enterprise. The return on investment from continuous improvements and eco-system integration can be proven.

4.4 Relation to CMMI

While CSOAMM can be used to situate a company in a detailed maturity level, it is often important to compare different companies with less details. This is important when trying to explain SOA and especially SOA maturity to people without knowledge of specific SOA maturity models. Regarding the software development process, the Capability Maturity Model Integrated (CMMI) is well known and hence, it is helpful to explain SOA maturity by using the mapping to CMMI. Arsanjani & Holley (2005c) write that the value of a mapping to CMMI is doubtful when excluding the "bandwagon factor". But this "bandwagon factor" is still important to offer an easy entry and understanding of new maturity models (like SOA maturity models). To fulfil this purpose, this subsection offers a rough connection between CSOAMM and CMMI. The mapping is illustrated in figure 8.

As SOAMM is created on the basis of CMMI, Sonic Software Corp. et al. (2005) offer a mapping of their 5 levels to the 5 levels of CMMI (excluding level 0) in their whitepaper. The different mapping to CMMI of this section can be explained by the fact that mappings to CMMI are very flexible according to Arsanjani & Holley (2005c) due to the open and interpretable descriptions of CMMI. A consequence of this mapping is that SOAMM and SIMM have no clear mapping to CMMI and hence, this mapping provides only an overview.

The Software Engineering Institute (2002) describes the practice of CMMI level 1 to perform base practices to achieve specific goals. This can be seen as an initial understanding of SOA, which could be mapped to level 0 and 1 of the CSOAMM.

The goal of level 2 is to institutionalise a managed process by, for example, establishing policy, creating standards and training people. Similar activities can be found
in CSOAMM levels 2 and 3, where standards are specified, SOA is integrated into development methodologies and a competency centre is created in combination with organisation-wide training.

In CMMI level 3, the process should be institutionalised and tailored to the company to be able to apply it to the whole company. This can be mapped to CSOAMM levels 4 and 5 where the initial institutionalisation process of level 3 (CSOAMM) is continued with architected services, internal/external services in the whole enterprise and virtual services with the possibility of being published as external services.

In CMMI level 4, the process should be controlled by quantitative techniques. This corresponds to CSOAMM level 6, where performance measurement in realtime is established in the enterprise. The practice of CMMI to stabilise subprocess performance can be seen as "fine-tuning of the infrastructure" in CSOAMM.

CMMI level 5 is characterised by a process adopted to meet relevant current and projected business objectives through continuous process improvement. The dynamic architecture in level 7 is exactly providing this continuous improvement and adoption to business objectives.

### 4.5 Scenario

This section demonstrates the value of CSOAMM using a fictitious scenario. The scenario shows the possibilities of the model for interorganisational connections as a small part of CSOAMM usage possibilities. Section 4.6 discusses more possibilities.

The scenario uses a medium-size company (A) using SOAMM and a big company (B) using SIMM. While A has just started to adopt SOA, Level 1 "Initial" in SOAMM, B is a leader in SOA and can be positioned at the lower end of level 5 "Composite Services" in SIMM.

Through CSOAMM, A was able to include best practices based on SIMM presented in literature and conferences to their own SOAMM based roadmap. A used CSOAMM as a tool to understand SIMM. Later, these two companies became partners. It was of great value that company A already had knowledge about SIMM and was used to utilise CSOAMM to interpret the SIMM levels. Hence it was not necessary to use CMMI as basic communication instrument, as B was using with other partners.

To work closer together, company B had to get familiar with the CSOAMM and the SOAMM. With the help of CSOAMM it was now possible for both companies to understand the other maturity model and hence, their approach to adopt SOA.

They decided to rearrange their own roadmaps to guarantee a better connection.
between the companies. The first step was to clarify their own positions, so company A positioned itself on level 2 "First published WS" in the CSOAMM while company B positioned itself on level 4 "Architected Services". They decided that the different maturity levels are not problematic as company A is smaller and hence, able to pass the levels faster than the big company (B). The experience of company B should help company A to speed up their adoption, so that they reach level 6 "measured services" together.

![Figure 9: Collaborative vision of company A and B](image)

After level 6 both companies will use a similar roadmap to intensify the collaboration and to reach level 7 as their collaborative vision. Further, they will start to search for more partners, using CSOAMM to find partner with similar maturity. Overall the CSOAMM was a communication tool which helped to understand the other company.

### 4.6 Analysis of CSOAMM

CSOAMM shows that SOAMM and SIMM can be combined and compared. By combining the level characteristics of these two models into CSOAMM, a better overview of SOA adoption is possible, as the level descriptions include more characteristics than just one model. Through the improved level descriptions, it is easier for companies to situate themself and plan their SOA adoption better. As CSOAMM is primarily no maturity model on its own, the connection between the models could be seen as additional information to SOAMM and SIMM. CSOAMM is a tool to clarify and help to interpret these two models. However CSOAMM was not created to be a maturity model, it can be used as one. Through the fact that the area of SOA is already confusing and SOAMM and SIMM are in the standardisation process, it would be even more confusing to use an alternative maturity model, like CSOAMM, proposed in this thesis. Hence, we recommend to use a standardised maturity model and to adjust and extend it to the company needs with the help of CSOAMM, for example.

With the assistance of CSOAMM companies can use external knowledge, e.g. best practices published in literature, of both models. Consequently, they have much more detailed information, which they can include in their individual SOA adoption roadmaps and are not restricted to the maturity model they choose when starting SOA adoption. CSOAMM provides the basis for company collaborations, when using different maturity models. Overall, the model clarifies SOA adoption by showing that both models describe similar levels.

CSOAMM was constructed by combining characteristics of SOAMM and SIMM, using the published descriptions of the models. The characteristics often include buzz-
words and the descriptions are not very detailed. Accordingly, there is the possibility of misinterpretation. Through splitting levels, the borders are not strict and must be interpreted. This is even more important as the model is a combination of characteristics (as already implied by the name "Combined SOA Maturity Model") and not a one-to-one mapping. Knowledge about both maturity models is necessary to interpret CSOAMM correctly. Sheldon Borkin (Progress Software) highlights in a comment to this thesis, that they explicitly created level 3a and 3b in SOAMM, as it is depending on the individual business drivers if companies publish services first internally or externally. To their experience, it is common that companies create external services first. The orientational character of CSOAMM makes it necessary to interpret the models. Companies must define clear borders between the levels based on their own business drivers.


5 Research summary

This section summarises and validates the results of this thesis. Additionally, the contribution is shown.

5.1 Research aim revisited

The chapter 3 shows how the two maturity models SOAMM and SIMM guide SOA adoption. Hence, the research question ”How do different SOA maturity models guide SOA adoption?” is answered.

The aim ”to develop a model for facilitating interpretation and comparison of SOA maturity models” is fulfilled by CSOAMM, presented in chapter 4, as a model showing the relation between the two models proposed for standardisation, SOAMM and SIMM.

The first group of objectives, to introduce SOA maturity models, is addressed by the literature analysis in chapter 3. This is done by mentioning benefits and criticisms and introducing SOAMM and SIMM as recent maturity models. The most mentioned benefits are: ”help in defining a SOA vision” and ”guidance in SOA adaption”. The main criticism is the ”too technical focus”.

The second group of objectives, to show the relation between SOA maturity models, is addressed in chapter 4 by the creation of CSOAMM. CSOAMM was created using an argumentative approach, which could be divided into a comparative analysis and a relational content analysis and with the aim of facilitating interpretation and comparison of SOA maturity models. The comparative analysis resulted in a mapping between SOAMM and SIMM levels and a list with the mapping characteristics. This list was extended by adding related level characteristics of both models, which could not be mapped directly to the other model. For this a relation content analysis was used.

The benefits of CSOAMM were illustrated in a scenario in chapter 4.5. Further a mapping between CSOAMM and CMMI was created in chapter 4.4.

5.2 Contribution

This thesis introduces SIMM and SOAMM as two recent SOA maturity models to the scientific area. It is one of the first introductions to SOA maturity models, which could help to clarify SOA and especially SOA adoption. The main scientific contribution is CSOAMM as a tool to interpret and compare SIMM and SOAMM. It shows that a common understanding of SOA adoption exists and that SOA maturity models could be used complementarily. As CSOAMM explains the whole process of SOA adoption, it can be used to position and motivate research projects. Companies can use CSOAMM in practise for an initial understanding of SOA, planning SOA and then for the collaboration with other companies, as demonstrated in the scenario. CSOAMM can be used as a translation tool between maturity models. Companies are not restricted to the model they choose when starting SOA adoption. Although CSOAMM was not created to be used as a maturity model, it could be used as one. We don’t recommend this, as we think, that another maturity model would increase the confusion in the SOA adoption area. Especially, since probably after the standardisation of SOAMM and SIMM, already two standardised solutions will exist.

The mapping between CSOAMM and CMMI offers the possibility to relate existing maturity knowledge to the area of SOA. It enables an easy understanding of these new
models. The work could help to establish SOA maturity as quality criteria like CMMI to support the decision process in the choice of business partners.

The idea of a combined model as a tool for clarification and interpretation could also be applied in other research areas.

### 5.3 Validation

Anfara, Brown & Mangione (2002) write that validity of research is reached by showing internal validity (creditability), external validity (transferability), reliability (dependability) and objectivity (confirmability).

The benefits and criticism of SOA maturity models are *internally valid* by using an in-depth literature analysis. The description of SIMM and SOAMM are validated and extended through the authors of the models which were in contact by email throughout the work process. Through the triangulation of the descriptions by using literature and additional contacts to the authors this part is *internal valid*. CSOAMM is based on an in-depth content analysis of the maturity models and also commented by the authors of SOAMM. Personal comments proved that the resulting CSOAMM levels are reasonable and, to a certain degree, independent levels. To create *external validity* and replicability, the working process is described in detail with a reference to a similar working process. Consistency to existing literature is provided in all areas. An external empirical validation to establish more *objectivity* and *reliability* must be addressed in future works (see chapter [6.3](#)). In an email response, Sean Kline (Systinet) describes this thesis as a very good summary of work in this area. This comment of a practitioner in the SOA area, and especially author of one of the discussed maturity models, assures the contribution and value of this thesis.
6 Discussion

This section discusses the research process and shows related and future work.

6.1 Research discussion

The weakness of the literature analysis, especially concerning SOA maturity models, is that most literature is gathered from e-magazines and company papers, which is not as reliable as review academic literature. This weakness should be accepted since this thesis is one of the first academic research works concerning SOA maturity models. Hence, there are only a few academic sources to use. Therefore, we tried to present a rather complete list of references.

Because the descriptions of SOA maturity models were of a basic nature, a personal interpretation was necessary. In particular, detailed information of SIMM was distributed over several publications with different focus. The interpretation is influenced by the personal background with no practical experience in this area. One must admit that the higher levels of maturity models are predictive and not proven in practise yet, as no company has reached the highest levels. To overcome the weakness of interpretation, the authors of SOAMM and SIMM were asked to validate the model descriptions. Unfortunately, only SOAMM could be validated.

The interpretation of level characteristics was influenced the creation of the combined model (CSOAMM). The problem of wrong interpretation is even more important, as the mapping between levels is often based only on one similar description of a characteristic. Especially maturity levels, leading to two levels are problematic as the placement of related characteristics may be weak. This weakness influences the usage possibilities of the model and it should be only used with additional knowledge about the two models.

To conduct interviews with other people in addition to the authors would have been too early, as the models are very new. It would have been interesting to elaborate the need of such models by interviews, but through the recent character, answers would have been probably unsatisfying. The only interesting group for interviews would have been consulting companies, which can use their experience to answer. But as mentioned earlier, no company has reached the vision of service orientation yet, so only preliminary and speculative answers would have been given. The same situation occurs when using surveys. As the scope of the models is very large, case studies or field studies are too complex to cover the whole models.

6.2 Related work

The introduction to maturity models in the CBDI Forum (Sprott 2005a) presented SIMM and SOAMM and can be seen as closely related work to section 3.2. In this thesis, the model descriptions are done in a more detailed way than in the CBDI Forum article, while the CBDI Forum introduced more models. In the academic area, Haines (2004) created a theoretical framework for Web Service technology adoption with a more technical focus than the SOA maturity models. Kajko-Mattsson & Tepczynski (2006) included also organisational, role and process changes in their SERVIAM Framework for evolving and maintaining Web Services. However, their aim is not to describe the adoption process in levels.
6.3 Future work

As SOA maturity models describe SOA adoption, all kind of literature and best practices, published in books and articles, can be seen as related work (e.g. Marks & Bell 2006, Ciganek, Haines & Haseman 2006, Stal 2006, Pulier, Taylor & Gaffney 2005, Wong-Bushby, Egan & Isaacson 2006, McGovern, Sims, Jain & Little 2006, Woods & Mattern 2006). In this category, especially other maturity models and roadmaps must be mentioned as related work (e.g. Bennett 2005, Wilkes 2006, Traventec Ltd. 2004, Schmelzer & Bloomberg 2005, Sprott 2005). Consulting companies often have individual, non-public maturity models. Further, OASIS is working on Adoption Blueprints and a SOA Reference Model, which have a more technical orientation (OASIS 2006a, OASIS 2006b).

6.3 Future work

Future work for this thesis is to investigate the area of SOA maturity models in the future, for example how the standardisation will be done and how it effects the usage of the model. Further, an external empirical validation is necessary. For this, it is necessary to investigate if the maturity models describe reasonable levels, which is not yet proven, as no company has reached one of the highest levels. In this context, a survey could help to investigate the actual dominating level. A case-study could prove the reasonability of a level. This could be seen as proof of concept for this theoretical work. An interesting field is also to compare SOA maturity models and other architecture maturity models e.g., the model of the Department of Commerce (DoC) (2003). Moreover, CSOAMM should be extended into a framework. Further, detailed information should be included from other references to create a more detailed model with relations to all areas of SOA for example modelling approaches.
References


REFERENCES


## Appendix A  SOAMM - Characteristics

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| 1. Initial     | New functionality  
Initial learning phase  
Pilot projects  
ESB  
Service Registry  
Service Level Management Service  
Legacy integration  
Developers learn service development skills  
Apply soa technology to immediate organisational needs  
Define initial ROI measurements for SOA projects  
Apply ROI measurements to initial projects  
Create service definitions  
Integrate SOA into project development methodology  
Quantify costs, time and business benefits of pilot projects  
Standards: XML, XSLT, WSDL, SOAP, Java, .Net |
| 2. Architected Services | multiple integrated applications  
service and policy repository  
governance using policies and service definition  
Single Sign on  
Exception Management Service  
Transformation Service of ESB used  
It cost reduction and control  
support for heterogeneity and distributed systems  
reliable messaging, mediation, ease of deployment database integration  
versioning, internal security, performance management  
institutionalise use of SOA  
put in place architecture leadership for SOA  
prove return from use of standards technology  
specify technology standards for SOA  
Integrate SOA into organisation-wide development process  
provide organisation-wide SOA training and competency center  
use incremental integration  
Standards: UDDI, ReliableMessaging, WS-Policy, WS-Adressing  
Xquery, WS-Security, SAML |

Table 5: SOAMM level characteristics 1 and 2
<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| 3a. Business Services  | Connection between business and technology  
                        | business responsiveness  
                        | support full business processes  
                        | prove returns from reuse and responsiveness to change  
                        | specify policies for creation or modification of business processes  
                        | use event-oriented and mediation functionality of technology  
                        | Reuse, Ease of modification, availability, business process rules  
                        | event-driven processes, composite applications  
                        | Orchestration Service to manage long running processes  
                        | Business Process Management  
                        | WS-BPEL  |
| 3b. Collaborative Services | connect internal services with external service  
                        | extend business processes to external organisations  
                        | specify policies for collaboration with business and trading partners  
                        | implement cross enterprise security  
                        | external services enablement, translation of protocols  
                        | long running transactions  
                        | Standards: Rosettenet, ebXML, WS-Trust  
                        | Prove returns from use of services for collaboration  |
| 4. Measured Business Services | performance measurement in realtime  
                        | business can be transformed from a reactive to a real-time business  
                        | Business Activity Monitoring (BAM)  
                        | define and meet business oriented performance metrics  
                        | collect and analyse business process oriented real-time performance metrics  
                        | implement ongoing business process evaluation and reengineering  
                        | Event Stream Processing, Complex Event Processing  
                        | Event-driven dashboards and alerts  |
| 5. Optimizing          | provides automation in business processes  
                        | business can be optimised by reacting and responding automatically  
                        | event-driven automation  
                        | self organised enterprise  
                        | enterprise nervous system  
                        | reacting to actions according to rules to optimise business goals  
                        | provide enterprise-wide leadership for business and SOA governance  
                        | Prove return from SOA supported continuous improvement  
                        | implement self-correcting business processes  
                        | event-driven automation for optimization  |

Table 6: SOAMM level characteristics 3 to 5
Appendix B  SIMM - Characteristics

This table shows the summarised characteristics for each level of the Service Integration Maturity Model (SIMM).

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| 1.Silo        | Data Integration  
High performance systems with few changes.  
start of proprietary and ad-hoc integration |
| 2.Integrated  | application integration  
EAI  
harmonised and coordinated information flow  
layered architecture  
exposed functionality  
proprietary connections, point-to-point |
| 3.Componentized | functional integration  
components and services identified  
legacy transformation and renovation  
clear boarders and scope  
modular exposure of functionality  
communication through interfaces and contracts  
hard-wired version of SOA based on CORBA/COM  
Investigation of Product interoperation  
services connected to business needs  
decision which services should be exposed  
start of changing corporate culture |
| 4.Simple Services | Process integration  
first ws published  
business view of IT ⇒ service-oriented  
business functions are integrated by services - SOI  
most try to achieve this level  
proprietary protocols and glue code ⇒ standards  
point to point ⇒ interaction based on service description  
ESB |
| 5.Composite Services | supply chain integration  
connection between business processes and services!  
services form a contract and on demand interactions  
enterprise wide IT transformation  
composite applications consisting out of services  
integration across business functions  
architected implementation |

Table 7: SIMM level characteristics 1 to 5
<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Virtualized Services</td>
<td>virtual infrastructure&lt;br&gt;decoupling the services form the applications&lt;br&gt;grid enabled SOA&lt;br&gt;technology neutral&lt;br&gt;Management, Monitoring and Events are handled by extra services&lt;br&gt;new business models are deployed and old business models are transformed.&lt;br&gt;finely tuning of infrastructure</td>
</tr>
<tr>
<td>7. Dynamically reconfigurable Services</td>
<td>Eco System integration&lt;br&gt;automatically reconfigurable architecture&lt;br&gt;compose services or applications at runtime&lt;br&gt;based on policy descriptions services&lt;br&gt;Grammar Modelling&lt;br&gt;dynamic sense and response concept</td>
</tr>
</tbody>
</table>

Table 8: SIMM level characteristics 6 and 7
Appendix C  CSOAMMM - Characteristics

This table shows CSOAMM characteristics created out of the characteristics of SOAMM and SIMM. S indicates characteristics from SOAMM and I from SIMM. L indicates that this characteristic is used for mapping the models.

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2 Silo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data integration (I)</td>
</tr>
<tr>
<td></td>
<td>Hard coded silos (I)</td>
</tr>
<tr>
<td></td>
<td>Embedded and often inaccessible, concentrated functions (I)</td>
</tr>
<tr>
<td></td>
<td>High performance systems with few changes (I)</td>
</tr>
<tr>
<td></td>
<td>Start of proprietary and ad-hoc integration (I)</td>
</tr>
<tr>
<td>-1. Integrated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Application Integration (I)</td>
</tr>
<tr>
<td></td>
<td>EAI (I)</td>
</tr>
<tr>
<td></td>
<td>Harmonised and coordinated information flows (I)</td>
</tr>
<tr>
<td></td>
<td>Layered Architecture (I)</td>
</tr>
<tr>
<td></td>
<td>Exposed functionality (I)</td>
</tr>
<tr>
<td></td>
<td>Proprietary point to point connections (I)</td>
</tr>
<tr>
<td>0. Components</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functional Integration (I)</td>
</tr>
<tr>
<td></td>
<td>Components identified (I)</td>
</tr>
<tr>
<td></td>
<td>Modular exposure of functionality (I)</td>
</tr>
<tr>
<td></td>
<td>Communications through interfaces and contracts (I)</td>
</tr>
<tr>
<td>1. Technology Tests</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Some Services identified and defined (I,S,L)</td>
</tr>
<tr>
<td></td>
<td>Decision which services should be exposed (I)</td>
</tr>
<tr>
<td></td>
<td>Start of changing corporate culture (I,L)</td>
</tr>
<tr>
<td></td>
<td>Initial Learning and Pilot Phase (S,L)</td>
</tr>
<tr>
<td></td>
<td>Developers learn service development skills (S)</td>
</tr>
<tr>
<td></td>
<td>Components with clear boarder and scope (I)</td>
</tr>
<tr>
<td></td>
<td>Legacy transformation and renovation (I,L)</td>
</tr>
<tr>
<td></td>
<td>Legacy Integration (S,L)</td>
</tr>
<tr>
<td></td>
<td>CORBA and COM (I)</td>
</tr>
<tr>
<td></td>
<td>Hard-wired version of SOA (I)</td>
</tr>
<tr>
<td>2. Published WS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>First published Web Services (I)</td>
</tr>
<tr>
<td></td>
<td>Service oriented business view of IT (I)</td>
</tr>
<tr>
<td></td>
<td>Integrate SOA into project development methodology (S)</td>
</tr>
<tr>
<td></td>
<td>Specify technology standards for SOA (S,L)</td>
</tr>
<tr>
<td></td>
<td>Proprietary protocols and glue code is replaced by standards (I,L)</td>
</tr>
<tr>
<td></td>
<td>Services connected to business needs (I,L)</td>
</tr>
<tr>
<td></td>
<td>Apply SOA technology to immediate organisational needs (S,L)</td>
</tr>
<tr>
<td></td>
<td>New functionality (S)</td>
</tr>
<tr>
<td></td>
<td>Process Integration (I)</td>
</tr>
<tr>
<td></td>
<td>Service description based interaction (I)</td>
</tr>
<tr>
<td></td>
<td>Enterprise Service Bus (ESB) (I,S,L)</td>
</tr>
<tr>
<td></td>
<td>Service Registry (S)</td>
</tr>
<tr>
<td></td>
<td>Service Level Management Service (S)</td>
</tr>
<tr>
<td></td>
<td>Define and apply initial Roi measurements for SOA projects (S)</td>
</tr>
<tr>
<td></td>
<td>Quantify costs, time and business benefits of pilot projects (S)</td>
</tr>
</tbody>
</table>

Table 9: CSOAMM level characteristics -2 to 2
<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Characteristics</th>
</tr>
</thead>
</table>
| 3. Institutionalisation | **Institutionalise use of SOA** (S)  
Most try to achieve this level (I)  
Put in place architecture leadership for SOA (S)  
Integrate SOA into organisation-wide development process (S)  
Provide organisation-wide training and competency centre (S)  
Support of heterogeneity and distributed systems (S)  
Business functions are integrated by services (SOI) (I)  
Governance using policies and service definitions (S)  
Service and Policy repository (S)  
Exception Management Service (S)  
Transformation Service of ESB used (S)  
IT cost reduction and control (S) |
| 4. Architected Services | **Architected Implementation** (I,L)  
Multiple integrated applications with a single sign-on (S,L)  
Prove return from use of standards technology (S) |
| 5. Internal Services | **Support full business processes** (S,L)  
Integration across business functions (I,L)  
Connection between business processes and services (I,S,L)  
Supply chain integration (I)  
External service enablement (S)  
Business Process Management (S)  
Specify policies for creation or modification of business processes (S)  
Services form a contract and on demand interactions (I)  
Enterprise wide IT transformation(composite applications) (I)  
Orchestration Service to manage long running processes (S)  
Prove return from reuse and responsiveness to change (S)  
Connect internal services with external services (S)  
Specify policies for SOA with business and trading partners (S)  
Extend business processes to external organisations (S)  
RosettaNet, ebXML (S)  
Implement cross enterprise security (S) |
| 6. Measured Services | **Business transformation from a reactive to a real-time business** (S,L)  
Decoupling the services from applications (I,L)  
Grid enabled SOA (I)  
Technology neutral (I)  
Virtual Infrastructure (I)  
Finely tuning of infrastructure (I,L)  
Implement ongoing business process evaluation and reengineering (S,L)  
Deployment of new business models and transformation of old ones (I)  
Analyse business processes (metrics) (S)  
Define and meet business oriented performance metrics (S)  
Performance measurement in realtime (S)  
Management, Monitoring and Events are handled by extra services (I,L)  
Business Activity Monitoring (BAM) (S,L) |
| 7. Dynamic Architecture | **Compose services at runtime** (I)  
Automatically reconfigurable (I)  
Automated business processes (S)  
Self-correcting (S)  
Event-driven (S)  
Acting according to rules to optimise business goals (S,L)  
Based on policies and grammars (I,L)  
Dynamic sense and response concept (I,L)  
Enterprise nervous system (S,L)  
Eco System Integration (I)  
Self organised enterprise (S)  
Prove ROI from continuous improvements (S) |

Table 10: CSOAMM level characteristics 3 to 7