

iSOAMM: An independent SOA Maturity Model

Christoph Rathfelder and Henning Groenda

FZI Research Center for Information Technology, Software Engineering
Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Germany
{rathfelder, groenda}@fzi.de

Abstract. The implementation of an enterprise-wide Service Oriented Architecture (SOA) is a complex task. In most cases, evolutionary approaches are used to handle this complexity. Maturity models are a possibility to plan and control such an evolution as they allow evaluating the current maturity and identifying current shortcomings. In order to support an SOA implementation, maturity models should also support in the selection of the most adequate maturity level and the deduction of a roadmap to this level. Existing SOA maturity models provide only weak assistance with the selection of an adequate maturity level. Most of them are developed by vendors of SOA products and often used to promote their products. In this paper, we introduce our independent SOA Maturity Model (iSOAMM), which is independent of the used technologies and products. In addition to the impacts on IT systems, it reflects the implications on organizational structures and governance. Furthermore, the iSOAMM lists the challenges, benefits and risks associated with each maturity level. This enables enterprises to select the most adequate maturity level for them, which is not necessarily the highest one.

1 Introduction

The use of an SOA promises organizations to adapt their software more rapidly to changing business needs. A successful implementation of an SOA is not limited to IT systems and requires changes throughout the whole enterprise [1, 2]. In order to handle this complexity, it is appropriate to implement an enterprise-wide SOA step by step using evolutionary approaches [3]. These approaches can also aid enterprises upgrading an already implemented SOA.

Maturity models are a possibility to support the planning and controlling of such evolutionary approaches. They can be used to assess the maturity and assist in improving the maturity as they provide the possibility to deduce a roadmap to a successful SOA implementation. The selection of the most adequate maturity level is an important part of developing a roadmap as the highest maturity level is not always the most suitable one for each enterprise. The benefits promised by a level have to be weighed against the costs to reach and maintain that level.

In order to ease the level selection, an SOA maturity model should exhibit levels, which are oriented at the capability of an SOA to support business processes. Additionally, it should also point out the benefits and costs that are associated with each maturity level. An SOA maturity model should furthermore be independent of the used technologies and products, as existing SOA implementations are based on a variety of technologies (e.g., CORBA, J2EE) [4, 5] and enterprises avoid to be dependent on a certain SOA vendor or product [4, 6]. Different case studies [6, 7] show that the success

of an enterprise-wide SOA implementation is often affected by organizational structures and IT governance. Therefore, an SOA maturity model should also consider the necessary alterations within these domains.

Existing SOA maturity models were in most cases developed by SOA vendors (e.g., IBM, BEA, HP, or Oracle) and cannot deny a dependency on the respective products. Additionally, the vendors take the desire to reach the highest maturity level for granted. Therefore, they often neglect supporting an enterprise in the selection of the most *appropriate* maturity level.

The contribution of this paper is the 1) product and technology independent SOA Maturity Model (iSOAMM), which 2) considers technical as well as organizational aspects. 3) It eases the selection of the most adequate maturity level by pointing out the challenges, benefits, and risks associated with each level. This distinguishes the iSOAMM from other SOA maturity models. The maturity levels are aligned with the SOA's capabilities to support business processes. This means that an SOA with higher maturity possesses more features, which are useful within business processes.

The development of the iSOAMM is based on a sound literature review as well as the experiences we have gained within different SOA projects. One example of a large SOA project we are currently involved in is the "Karlsruher Integriertes InformationsManagement" (KIM) [8]. Its aim is the implementation of a university-wide SOA which supports students as well as staff. In addition, we have taken several published SOA case studies (e.g., Deutsche Post [4–6], Credit Suisse [4, 5], ABB [7], and Sparkassen Informatik [9]) into account. Furthermore, we evaluated a variety of articles, which document best practices, success factors, and experiences related to SOA implementations, as well as publications, which present new research results and open research questions. In order to validate the iSOAMM we evaluated the case studies and rated the maturity of documented SOA implementation.

This paper is structured as follows: Section 2 gives an overview of related work. Section 3 describes the different viewpoints that are used to define the maturity levels. Section 4 presents the requirements, which have to be satisfied for each maturity level. Section 5 points out the challenges, benefits, and risks associated with each maturity level. Section 6 presents the validation of the iSOAMM. Section 7 concludes the paper and provides an outlook to future work.

2 Related Work

Regarding the maturity of enterprise architectures the US Department of Commerce (DoC) has developed the Architecture Capability Maturity Model (ACMM) [10]. It differentiates nine evaluation categories. This granularity enables to identify areas that are more ahead or behind than others and eases the planning of the next steps to reach a higher overall maturity. The different categories of the ACMM reflect that it is insufficient to analyze the architecture of a software system only, as the maturity is also heavily influenced by the organizational structure of an enterprise. The ACMM is a general architecture maturity model. However, the ACMM does not consider the particularities of an SOA, such as dynamic adaptation through loose coupling of services, the consequences of higher business alignment, or reuse of services.

The most well-known maturity model for SOA is the New SOA Maturity Model (NSOAMM) [11] developed by Sonic Software, AmberPoint, BearingPoint, and Systi-

net. The NSOAMM is limited to WS-based SOA implementations. It does not consider areas like security, monitoring, and management of services. Especially in this model, a particularly high product dependency of the model is obvious. Furthermore, it neglects areas like organizational structure and governance.

The Oracle Maturity Model (OMM) [12] is divided into five maturity levels. Each level is split into eight categories, which cover the software architecture and infrastructure as well as governance, development, and enterprise structure. Additionally, a list of Key Indicators (KI) for each maturity level sorted by the categories is available in [13]. The OMM stipulates the use of WS-technology to implement an SOA. Furthermore, the support of user integration or automated business-to-business (B2B)-processes is not part of this model.

The Service Integration Maturity Model (SIMM) [14] was developed by IBM in 2005. It consists of seven maturity levels, whereas only the last four maturity levels consider services [15]. Comparable to the OMM, the SIMM is split into seven independent categories but the KI are not publicly available. IBM has externalized the SIMM to the Open Group at the beginning of 2007. The Open Group plans to enhance it and publish it as the Open Group Service Integration Maturity Model (OSIMM) [15]. Hence, this will lead to a publication of the KI in the near future.

Based on the SIMM and the NSOAMM, Meier defined the Combined SOA Maturity Model (CSOAMM) [16]. It is a scientific model, which is a common denominator of these two models enabling a comparison of the evaluation results. With this target in mind, he considers only the different maturity levels and neglects the more detailed subdivision into categories given by the SIMM. Additionally, all the non-technical indicators regarded in the SIMM are not taken into account.

3 Evaluation Viewpoints

As already mentioned, it is not sufficient to limit the evaluation of an SOA's maturity solely to technology-dependent criteria. The iSOAMM uses the following 5 viewpoints which regard technological as well organizational aspects. Overall, they cover the same domains as the SIMM, the OMM, and the ACMM.

1. **Service Architecture:** This viewpoint regards architectural layers of an SOA as well as services, their roles within business processes, and the interaction between them. The architecture can vary from providing an integration layer only to direct support of business processes with orchestrated services, user interaction, and B2B-cooperation.
2. **Infrastructure:** The loose coupling of services facilitated by an SOA supports a rapid adaptation to new business requirements. However, this high adaptability requires a stable infrastructure [4, 6]. It is therefore necessary to examine the infrastructure separate from the services, their composition, and their interaction. The SOA infrastructure mainly provides a common communication layer to all services [4, 5], which can be extended by additional components and layers (e.g., monitoring or security enforcement) [4, 17].
3. **Enterprise Structure:** SOA affects IT systems as well as business processes [2]. Changes which affect organizational structure and responsibilities of the different divisions are therefore required [5, 7, 18]. This viewpoint regards the different divi-

sions of the company, which are affected by the SOA, as well as their responsibilities and duties.

4. **Service Development:** The design and implementation of services is a crucial aspect in the implementation of an SOA. As Cox and Kreger emphasize in [19], the development process of services needs to be adjusted and it is therefore regarded as a separate viewpoint. In general, an increase in maturity leads to a higher rate of automation within the development process [6, 9].
5. **Governance:** The successful implementation and usage of an SOA has to come along with an adaptation of the whole enterprise [1]. This viewpoint considers changes, rules, and guidelines that are relevant for the whole enterprise and are not limited to *Enterprise Structure* and *Service Development*. The topic of SOA governance is so large, that we can only present the main KI of this viewpoint for each maturity level.

4 iSOAMM Maturity Levels

After the introduction of the evaluation viewpoints, this section describes the five different maturity levels (*Trial SOA*, *Integrative SOA*, *Administered SOA*, *Cooperative SOA*, and *On Demand SOA*) and their Key Indicators (KI). In defiance of the iSOAMM's independence, the examples within the description of the maturity levels use web service (WS) standards as illustration since many implemented SOAs are based upon WS [20].

Note that each maturity level constitutes an enhancement of the previous level and hence bases on changes and features already introduced at lower levels. However, KI of lower levels can also be overruled, for example if the structure of an enterprise changes and organizational units are dissolved and replaced by others. Figure 1 gives an overview of iSOAMM and its subdivision into maturity levels and evaluation viewpoints. The different maturity levels are described in detail in the following subsections.

Viewpoint Maturity Level	Service Architecture	Infrastructure	Enterprise Structure	Service Development	Governance
5 On Demand SOA	dynamic services	service marketplace	service as business	service on demand	automated
4 Cooperative SOA	processes	management, event-driven	service aligned	model-driven	fair competition control
3 Administered SOA	orchestrated services	monitoring, security	centrally managed	documented, tool support	rules
2 Integrative SOA	integrated applications	communication	IT-oriented	hands-on experiences	guidelines
1 Trial SOA	islands	inhomogeneous	separated	unstructured	none

Fig. 1. Maturity Levels

4.1 Level 1: Trial SOA

This level of maturity can be attested to small, mostly independent SOA projects. Within these projects, an enterprise gains first experiences with services. There is no common technology or fixed set of standards that is used within all projects.

Service Architecture. The point-to-point interfaces between legacy applications, which link a pair of applications, are substituted by services that can be used by more than one

application. Due to the lack of standardization, it is possible that different services use incompatible technologies and standards. So this level is a collection of miscellaneous service islands rather than a real service architecture.

Infrastructure. Due to the independency of the small SOA projects, it is likely that different communication systems and standards are chosen. This inhomogeneous infrastructure often leads to incompatible service islands.

Enterprise Structure. The enterprise structure is characterized by a strict separation into independent business departments. Each department has its own application landscape which is developed and maintained by a separate IT section. Cooperation across business unit borders is very rare.

Service Development. The development of services is unstructured and done independently for each SOA project. In most cases, the purpose is to gain experience and develop best practices and guidelines for the implementation of an enterprise-wide SOA.

Governance. Early SOA projects are mainly initiated by IT departments, which are responsible for the integration of diverse applications. SOA is therefore often regarded as a pure IT project, which only marginally affects other business units. This is usually accompanied by a lack of support of SOA projects by the top management of the enterprise.

4.2 Level 2: Integrative SOA

The experience gained in SOA projects at the previous level or drawn from best practice reports is used to select an adequate infrastructural basis for the enterprise. The target of SOAs at this maturity level is mainly the integration of systems in the IT landscape and the realization of Enterprise Application Integration (EAI) [21].

Service Architecture. The different standards and technologies used in the previous maturity level are substituted by a common Service Bus (SB) [4]. The implementation of standardized service interfaces leads to a common high level Application Programming Interface (API), which can be used by frontend applications to access different backend systems [9].

Infrastructure. The common infrastructure represents the backbone of an SOA. Hence, the requirements in terms of scalability, reliability, availability, and performance that are imposed on the SB are very high [4]. Depending on the needs of an enterprise, a SB is built upon quite different technologies and standards. For example, Credit Suisse uses CORBA within their SB [4], Sparkassen Informatik uses WS technology [9], and the SB of Deutsche Post is based on J2EE technology [5]. The SB additionally provides logical addressing of services. This allows changing the physical location of a service.

Enterprise Structure. The cooperation between different business units increases in comparison to the previous maturity level. The major alteration is the introduction of an SOA team. It is an independent group of IT experts, which is the contact point for all business units regarding SOA-related questions. The team members are responsible for the design of the SB and for consulting and training personnel concerning the implementation and integration of services. For example, all enterprises regarded in [5] have established such a central SOA team.

Service Development. Service developers are supported by a knowledge-base that includes lessons learned, best practices and guidelines. Thanks to the regulation of the used standards and technology, a better tool support (a service stub generator or a test

environment for example) is provided and not every development team has to find the most appropriate toolset on its own.

Governance. A consistent change management and versioning becomes necessary since provided services can be consumed by several applications or services. Therefore, enterprise-wide guidelines have to be defined, that standardize the handling of change requests and the rollout of altered services [6]. Especially in the initial stage of an SOA of maturity level two, the integration of services into the SB is more complex than using different proprietary interfaces, because of a lack of experience. Therefore, an enterprise-wide compensation payment system has to be instantiated that balances these extra costs [4].

4.3 Level 3: Administered SOA

The third maturity level is characterized by orchestrated services. The IT-system oriented integration services described in the previous levels are orchestrated to implement services with a higher alignment of service's functionality to business processes. Figure 2 sketches the *Service Architecture* and *Infrastructure* of an SOA on level 3.

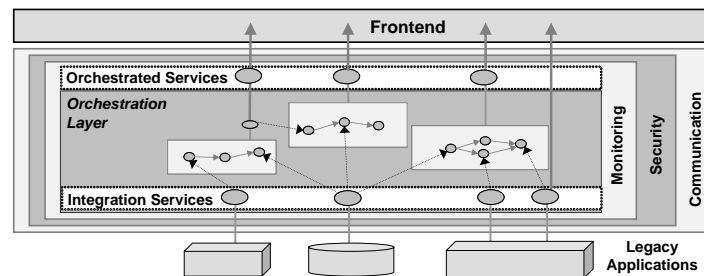


Fig. 2. iSOAMM: Maturity Level 3

Service Architecture. The existence of an orchestration layer distinguishes the *Service Architecture* between maturity level two and three, and allows a higher degree of business alignment. Orchestrations are generally implemented in a process-oriented way and represent the business logic that was formerly hardcoded into the frontend applications [22]. Orchestrated services represent the composition of IT-oriented functionality and realize business-oriented functionality. For example, the “Web Services Business Process Execution Language” (WSBPEL) [23] is one of the most well-known programming languages to implement WS orchestrations. Supplementary to the introduction of orchestrated services, a standardization of business relevant data types (*customer* or *product* for example) is necessary. This minimizes the need for data transformations within orchestrations and thereby eases the reuse of integration services that work on this data.

Infrastructure. The orchestration's implementations are process descriptions and not native executable code. These languages need a runtime environment, the orchestration engine, which is one of the mandatory extensions of the infrastructure. The SB has to be enriched with additional functionality [17]. The communication infrastructure has to be flanked with a monitoring and a security infrastructure. This extension can for example also be found in the “Web Services Architecture Stack” [24] proposed by the W3C. The monitoring infrastructure enables the observation of performance and availability of the services. It is sufficient to monitor the messages, respectively the service

invocations and the corresponding responses. A more detailed monitoring including the internal states of a service is not mandatory at this maturity level. Comparable to the monitoring, the security infrastructure of this maturity level operates only on message and service granularity. This enables the definition and enforcement of access rights for services, whereas the actual data of a service call is not part of the security policies. Furthermore, a secure message exchange between service provider and the service consumer is possible.

Enterprise Structure. Regarding the functionality and the data that is provided by services they can be split up into different service domains, e.g. accounting, customer master data, customer relationship etc. [6]. Each department is responsible for a certain domain and the included services. Besides the development of mainly orchestrated services each department is responsible for the operation of the services. Service operation is often assigned to an own IT unit or a common IT department. The SOA team is adapted to reflect the higher business alignment. On level two, it mainly consists of IT experts whereas on this level it is an interdisciplinary team that includes IT experts as well as business experts of different business units [7]. This team is also responsible for defining common data standards and the splitting of services into service domains.

Service Development. The knowledge and tool repository, which supports service developers, is enhanced and the share of automated development steps is increased [6]. The degree of automation can be risen by using Model-Driven Software Development (MDS) for the development of orchestrated services. The use of MDS is much easier for orchestrated services than integration services, as the first ones use only standardized services and feature a common implementation language.

Governance. An enterprise-wide policy has to be established that the service orientation paradigm has to be applied throughout the IT landscape. However, reasonable exceptions can still be allowed by the SOA team. The use of services by other parties induces costs for the operation at the service provider instead of the consumer. Hence, the compensation payment system has to be adapted so service providers are not punished for providing reusable services. The reuse factor of services can vary to a big extent, as the case study of Credit Suisse in [4] shows for example that in spite of an overall reuse factor of 1.6 some services are reused up to 12 times. Part of the governance of this level also is the establishment of enterprise-wide rules, guidelines, and policies which regulate security concerns.

4.4 Level 4: Cooperative SOA

This maturity level is characterized by Service Level Agreements (SLA), which have to be concluded between service consumer and provider. An SLA warrants a specified service quality if the consumer uses the service in conformance to a specified usage profile. An additional architecture layer closes the gap between services and business processes [25]. Corresponding to [11], it has to be distinguished between B2B-processes, which are mainly full automated, and internal processes, which involve human interaction. The *Service Architecture* and the *Infrastructure* of an SOA at this level are sketched in Figure 3.

Service Architecture. As mentioned above, the service architecture can have two different characteristics (B2B-processes and human interaction), even both at the same time. In order to support B2B-scenarios this layer supports the choreography of pro-

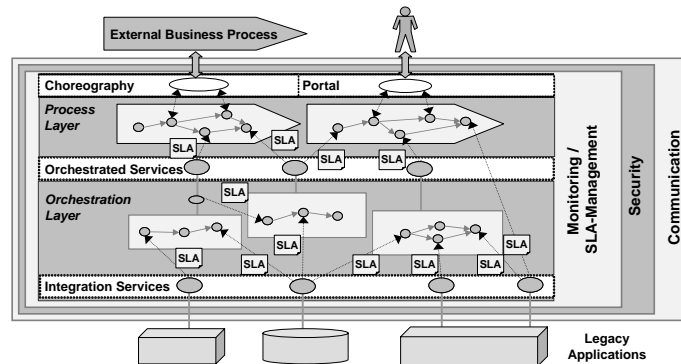


Fig. 3. iSOAMM: Maturity Level 4

processes. In contrast to an orchestration, choreography is a cooperation between processes [26]. The integration of human users is necessary to support most internal processes, thus they can not be implemented as orchestrations, which allow solely a composition of services. The common way is to use a portal that presents the tasks to the users, which are assigned within the process. One example for such a user integration is the "WSBPEL Extension for People" (BPEL4People) [27]. The availability of business rules is an additional property of the process layer. The business rules allow a reconfiguration of processes without a redeployment [28]. Secondary, as presented in [29], events play a decisive role within real-world business processes. Therefore, the communication between services is extended to support events in addition to direct service invocations by other services.

Infrastructure. First of all, the communication infrastructure has to be adapted in order to support the event-based communication whereas the possibility of active service invocations is still available. One example is the "Event Bus Infrastructure" [4] developed by Credit Suisse. The introduction of the process layer puts new requirements on the infrastructure. It has to provide a rule system. Additionally, components are necessary, which either allow the integration of user actions into a process - mainly by a portal - or the choreography of processes. Similar to orchestrations, processes require a runtime environment. These process engines are mainly orchestration engines that provide the additionally needed functionality of choreography and user integration. Generally, business experts rather than IT experts design processes. Therefore, descriptions of services that are located within the repository have to include a semantic description. An example of a UDDI-based repository that allows the integration of semantic descriptions is presented in [30]. At this maturity level, SLAs are concluded during the development of an orchestration or a process and changed rather seldom. An automation of this task is therefore not yet mandatory. Nevertheless, a more detailed monitoring of the services is essential, especially the internal states of orchestrations and processes have to be monitored [31]. An example of such a monitoring infrastructure for WSBPEL orchestrations is presented in [32]. The security infrastructure has to be extended. As shown in [33], it is insufficient to define access policies on services or even interface level. In fact, it is necessary to take the data into account, which is included in the service call.

Enterprise Structure. The subdivision of the enterprise into several departments, which correspond to the service domains, is refined into smaller units. Bieberstein et al [1]

propose a service alignment down to the granularity of teams that are responsible and specialized for only one service. Thus, the establishment of new business functionality is not solely a composition of services. It is rather a new interdisciplinary staffed combination of teams within the enterprise. Nevertheless, a separate IT department, which operates and maintains the infrastructure, is still necessary.

Service Development. The development process has to consider the potentials of business rules and events. The development is based on the MDSD and uses graphical models to design processes and orchestrations, which are transformed into interpretable code. The use of graphical models simplifies the development so that it is easier for business experts. Furthermore, the integration into the security infrastructure is an important extension of the development process. The SLAs include quality parameters, which are guaranteed by the provider. It is therefore essential to consider the quality of service (QoS) within the development process. This means the development process has to be extended with activities to predict or at least estimate the QoS in relation to the QoS guaranteed by the included services [31]. The SLAs include a charge, which has to be paid to use the service [34]. This allows the optimization of costs regarding the quality, which is guaranteed to the consumer [31].

Governance. At least from this maturity level on, all new IT systems have to be implemented in a service-oriented manner. Furthermore, it is mandatory that all legacy applications are extended with service interfaces and integrated into the SOA. The compensation payment system is replaced by SLAs because they define fees for the usage of the service. The teams can balance the additional expenses that are induced by the service development and the service operation on their own. Nevertheless, enterprise-wide rules and a regulating instance exercising fair competition control is necessary in order to prevent unfair enrichment through monopoly positions on crucial services. Key process indicators and metrics have to be defined, which enable monitoring of the enterprise-wide SOA adoption and the business processes. Especially the business process monitoring can use the already present monitoring infrastructure.

4.5 Level 5: On Demand SOA

At the previous level, services were published including their available quality levels. Such a request triggers the provider to check if he can offer the demanded quality level. The provider in turn often has to conclude new SLAs with his own service providers. In addition to the long term contracts on level four, it is possible to negotiate SLAs on a short-term basis. This can even be as short as a single service invocation. Because of these short-term SLAs, an automation of the SLA negotiation is necessary [35].

Service Architecture. Since services are selected automatically, a service and data ontology is needed [31]. The static binding of the services is replaced with a dynamic binding using the semantic description of the required functionality. This description in combination with the ontology enables the selection of suitable services. Furthermore, policies have to be defined that control the choice of the most adequate service. By using semantic service matching, it is possible that there is a mismatch between the data formats used in the process and the ones that are used by integrated services. A data ontology enables an automated transformation of different data formats.

Infrastructure. A trading platform, which is called marketplace [36] in the following, is required to support the automated search and provisioning of services. It can provide

a variety of different purchasing models. Two examples are public sale of services [36] and the selection out of several offers [34]. The introduction of automated service selection leads to further changes in the infrastructure because components for automatic SLA negotiation and service selection become necessary. One framework that allows such negotiations and additionally the monitoring of SLAs is described in [35]. In order to enable an optimized service selection at runtime, a detailed monitoring of the orchestrations and processes is needed [31].

Enterprise Structure. There are only minor changes to the enterprise structure in comparison to level four. The provisioning of services is now the primary goal of the enterprise and the management has to ensure the agility to adapt to changes in customer requests.

Service Development. In contrary to maturity level four, services are not chosen at design-time. In fact, they are selected during runtime and therefore service selection regarding the costs and the quality is more flexible. This shift of the optimization from design-time to runtime has some serious implications. Rules and policies, which control the service selection during runtime, have to be defined within the design phase. The service development evolves from a combination of services into a composition of functionality. Orchestrations and processes contain the semantic specification of services that can be used to automatically query service repositories or the marketplace for compatible services.

Governance. The business process monitoring evolves into Business Process Management, which allows the control and optimization of business processes. Due to the dynamic conclusion of SLAs, the monitoring of the compliance with the rules, which have been introduced with maturity level four, has to be automated

5 Challenges, Benefits, and Risks

This section presents the benefits that are promised by tackling the challenges of each maturity level. Additionally, the risks that are associated with the ascent from one level to the next are listed. This eases a comparison and evaluation of the most adequate maturity level because the benefits can be weighted up against the challenges and risks.

5.1 Level 1: Trial SOA

The challenge is comprised of the introduction of the service paradigm involving the way of thinking and often the use of new technologies. Particularly in this first maturity level, the development costs associated with the development of a service are slightly higher than the costs of point-to-point interfaces.

A major benefit for the enterprise is to gain experience in adopting SOA in their environment. Especially the lessons learned in the use of the technology and standards are valuable when striving for the next maturity level.

The risks involved in adopting the SOA approach at this maturity level are relatively low as only small and delimited projects are affected.

5.2 Level 2: Integrative SOA

The selection or rather implementation of the SB is difficult because many aspects like availability, performance, and especially scalability have to be considered [4]. Sec-

ondary, the developers have to be trained in the technologies and standards provided by the SB.

The availability of a high-level API that offers standardized access to the services is the main gain of an SOA on this maturity level [9]. Such an API eases the service reuse within new application and leads to an reduction of development costs [4]. Furthermore, the knowledge-base and extensive tool support can speed up the development time.

The SB is a central component within the enterprises IT landscape. As shown in [4], the scalability of the SB represents a large risk to the success of the SOA and with it to the success of the whole enterprise. The regulation of new common standards and technologies is an additional risk as such introductions often do not find wide acceptance amongst the developers [37].

5.3 Level 3: Administered SOA

The extensions of the infrastructure that are enjoined for this maturity level produce obviously high one-time effort. The introduction of an appropriate monitoring and security infrastructure is challenging as future requirements have to be anticipated. Secondary, a development method for orchestrated services has to be implemented.

The service orchestrations allow a stronger alignment of IT and business. They can be changed quickly according to new or changed business needs and the time-to-market is reduced [4,25]. The implementation of new frontend applications is easier due to the higher business alignment and therefore faster in general. Redundant effort is decreased through the exchange of experiences and the increased reuse potential of services enabled by the standardized data types. Additionally, establishment of the orchestration layer is comparable to the separation of business logic from the data layer and the user interface, which is generally considered an advantage.

The step from level two to level three requires the instantiation of the interdisciplinary SOA team and the reorganization of business departments. The acceptance of these changes presents a non-negligible risk for the SOA implementation. Services can be reused within several orchestrations and applications, therefore a failure of this single service can affect a lot of different services and applications.

5.4 Level 4: Cooperative SOA

An SLA management has to be introduced. Employees have to be trained in handling SLAs on the technical and on the business side. Furthermore, the integration of user interaction lead to substantial enhancements of the infrastructure. The application of business rules in orchestrations requires an anticipation of changes in designing of business processes. Another challenge is the reorganization of the departments into teams as this needs a lot of support by many stakeholders throughout the whole enterprise.

The process layer promises better business support by integrating human interactions. The choreography enables the realization of complex B2B-processes. The business rules allow a faster reaction to changes in the process specification and a seamless transition at runtime. The use of SLAs enables the inclusion of extra-functional quality properties and hence the consideration of costs in relation to quality properties like performance [31]. Additionally, the fine-grained security model allows a rollout of services even in vital areas of the enterprise. Last but not least, the service selection is eased by the fact that semantic searches in the service repository are possible. The decisive role that events play in business processes can now directly be mapped to the IT processes.

The risk for IT-related changes like introducing the process layer as well as a business rules engine is comparable low. The structural reorganization of the enterprise depends to a big extent on the support of all employees and also requires the sensitivity if all are willing to take the risk. In total, SOA becomes an essential factor for success or failure in achieving the business objectives of an enterprise.

5.5 Level 5: On Demand SOA

This maturity level induces large challenges. First, a change in the way of thinking is necessary. The on demand negotiation substitutes the offering of services with pre-assigned QoS. This means the customer can make demands, which the provider endeavors to realize. Secondly, mainly all activities related to the operation of the SOA have to be automated in order to enable the dynamic service selection and binding. This is not only a technical challenge, because the staff member carrying out these tasks manually before are affected.

This maturity level promises an even faster adaptability and higher flexibility than the previous levels [38]. A higher customer orientation is possible due to the on demand provisioning of services. Especially in combination with an underlying virtualized computing infrastructure, the proposed resource efficiency can be optimized [38].

If the new on demand paradigm is not put into practice by the employees, the promised benefits can not be reached. Furthermore, the full automation can be misinterpreted as a loss of control by the managers leading to a disaffirmation of SOA.

6 Validation

In order to validate the iSOAMM, we rated several SOAs based on the information provided in the case studies. For the sake of brevity, we only present the results and omit evaluation details. The SOA of Sparkassen Informatik [9] is one example that is ranked on level two. The KIM project was also ranked on level two when it started in 2004 and reached level three last year. The SOA of Deutsche Post [4, 6] is also ranked on iSOAMM level three. Currently, there are no level four SOAs known to the authors. That is not unexpected as SLA management and QoS assurance are still two large research challenges that are not fully mastered [39]. Level five is more or less a vision only, although a lot of research is done on topics linked to this level. For example, the European Union funds research projects which address these aspects of an SOA within the ICT domain of the 7th Framework Programme.

7 Conclusion and Outlook

This paper introduces the iSOAMM and its five maturity levels. The maturity levels were explored from five different viewpoints to highlight changes in IT systems, procedure, and the organizational structure of an enterprise. In addition, the challenges, benefits, and risks associated with each level were pointed out which enables enterprises to select the most adequate maturity level for them. In order to validate the iSOAMM, the implemented SOAs referenced in this article were ranked according to the maturity levels, as far as the corresponding documentation allowed it.

The iSOAMM enables enterprises to identify rewarding areas for further SOA adoption and to develop a roadmap for an SOA introduction. Due to the independency of the iSOAMM, they are free to choose the most adequate technologies, standards, and products. The iSOAMM consolidates current knowledge about the introduction and implementation of an SOA. It also merges the experience gained in SOA projects in industry with latest research results.

As a next step, we plan a further validation of the iSOAMM. For example, we aim to evaluate more enterprises using our model. We also plan to compare the KI of the iSOAMM to the ones of the OSIMM, as soon as they are publicly available. In the long run, the refinement of KI of the levels four and, in particular, five is planned. As these levels cover current research fields, new scientific knowledge and practical experience may lead to adaptations in the KIs.

References

1. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. *IBM Systems Journal* **44**(4) (2005) 691–708
2. Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., Rackham, G.: Impact of service orientation at the business level. *IBM Systems Journal* **44**(4) (2005) 653–668
3. Heffner, R., Fulton, L., Stone, J.: Key SOA Success Factors: A Starter Kit For SOA. Forrester Research (2006)
4. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA. reprint. edn. Prentice Hall PTR (2006)
5. Legner, C., Heutschi, R.: SOA Adoption in Practice - Findings from Early SOA Implementations. In: Proc. of European Conference on Information Systems (ECIS 2007). (2007)
6. Helbig, J.: SOA Serie: Teil 1-5. *CIO* (2007) <http://www.cio.de/schwerpunkt/d/Deutsche-Post-Brief.html>.
7. Gizanis, D., Heutschi, R., Solberg, T.: Global Order Management Services Support Businesses at ABB (2005) <http://www.alexandria.unisg.ch/Publikationen/23667>.
8. Freudenstein, P., Liu, L., Majer, F., Maurer, A., Momm, C., Ried, D., Juling, W.: Architektur für ein universitätsweit integriertes Informations- und Dienstmanagement. In: *INFORMATIK 2006 - Informatik für Menschen* (Band 1). (2006) 50–54
9. Zimmermann, O., Milinski, S., Craes, M., Oellermann, F.: Second generation web services-oriented architecture in production in the finance industry. In: *OOPSLA '04*. (2004)
10. Department of Commerce: (Introduction - IT Architecture Capability Maturity Model) http://ocio.os.doc.gov/groups/public/@doc/@os/@ocio/@oitpp/documents/content/prod01_002340.pdf.
11. Bachman, J., Ng, D., Kline, S., Horst, E.: A New Service-Oriented Architecture (SOA) Maturity Model. whitepaper (2006) <http://www.sonicsoftware.com/soamm>.
12. Trops, B.: SOA Maturity Modell, oder der Weg zu einer Service Orientierten Architektur. Java Forum Stuttgart (2006) http://www.jfs2006.de/jfs/2006/fohlen/B7_Trops.ORACLE.pdf.
13. ORACLE: (SOA Maturity Cheat Sheet) <http://www.oracle.com/technologies/soa/docs/oracle-soa-maturity-model-cheat-sheet.pdf>.
14. Arsanjani, A., Holley, K.: Increase flexibility with the Service Integration Maturity Model (SIMM) (2005) <http://www.ibm.com/developerworks/webservices/library/ws-soa-simm/>.
15. The Open Group, OSIMM Working group: Launch Presentation and WG Updates 1.0 (2007) <http://www.opengroup.org/projects/osimm/>.
16. Meier, F.: Service Oriented Architecture Maturity Models - A guide to SOA adoption. Master's thesis, University of Skövde, School of Humanities and Informatics (2006)

17. M. P. Papazoglou: Extending the Service-Oriented Architecture. *business integration journal* (2005) 18–21
18. Baer, T.: SOA: BUILDING THE ROADMAP. *zaphink white paper* (2007)
19. Cox, D.E., Kreger, H.: Management of the service-oriented-architecture life cycle. *IBM Systems Journal* **44**(4) (2005) 709–726
20. Austvold, E., Carter, K.: Service-Oriented Architectures: Survey Findings on Deployment and Plans for the Future. AMR Research, Inc., Research Report (2005)
21. Linthicum, D.S.: Next generation application integration. 3. printing edn. (Addison-Wesley)
22. Wong-Bushby, I., Egan, R., Isaacson, C.: A Case Study in SOA and Re-architecture at Company ABC. In: Proc. of the HICSS'06. (2006)
23. OASIS: Web Services Business Process Execution Language (WSBPEL) (2007)
24. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture. W3C Working Group Note (2004)
25. Zimmermann, O., Doubrovski, V., Grundler, J., Hogg, K.: Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned. In: Proc. of OOPSLA '05. (2005)
26. Peltz, C.: Web Service Orchestration and Choreography: A look at WSCI and BPEL4WS. *WebServices Journal* (2003)
27. Agrawal, A., Amend, M., Das, M., Ford, M., Keller, C., Kloppmann, M., König, D., Leymann, F., Müller, R., Pfau, G., Plösser, K., Rangaswamy, R., Rickayzen, A., Rowley, M., Schmidt, P., Trickovic, I., Yiu, A., Zeller, M.: WS-BPEL Extension for People (BPEL4People) (2007) <https://www.sdn.sap.com/irj/sdn/bpel4people>.
28. Charfi, A., Mezini, M.: Hybrid web service composition: business processes meet business rules. (In: Proceedings of ICSSOC 2004)
29. Schulte, R.W.: The Growing Role of Events in Enterprise Applications. Gartner Research (2003) http://www.gartner.com/DisplayDocument?doc_cd=116129.
30. Paoli, H., Schmidt, A., Lockemann, P.C.: User-Driven Semantic Wiki-based Business Service Description. In: Proceedings of I-Semantics 07. (2007)
31. Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for Web services composition. *IEEE Trans. on Softw. Eng.* **30**(5) (2004)
32. Momm, C., Mayerl, C., Rathfelder, C., Abeck, S.: A Manageability Infrastructure for the Monitoring of Web Service. In: 14th Annual Workshop of HP SUA. (2007)
33. Emig, C., Brandt, F., Kreuzer, S., Abeck, S.: Identity as a Service - Towards a Service-Oriented Identity Management Architecture. In: Dependable and Adaptable Networks and Services. Volume 4606/2007. (2007) 1–8
34. Tasic, V., Patel, K., Pagurek, B.: WSOL - Web Service Offerings Language. In: Proc. of the Workshop on Web Services, e-Business, and the Semantic Web (at CAiSE02). (2002) 57–67
35. Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web services on demand: WSLA-driven automated management. *IBM Systems Journal* **43**(1) (2004) 136–158
36. Lamparter, S., Schnizler, B.: Trading services in ontology-driven markets. In: Proc. of the 21st Annual ACM Symposium on Applied Computing (SAC'06). (2006)
37. Haft, M., Humm, B., Siedersleben, J.: The Architect's Dilemma - Will Reference Architectures Help? In: Quality of Software Architectures and Software Quality. (2005) 106–122
38. Crawford, C.H., Bate, G.P., Cherbakov, L., Holley, K., Tsocanos, C.: Toward an on demand service-oriented architecture. *IBM Systems Journal* **44**(1) (2005) 81–107
39. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal* **16**(3) (2007) 389–415

All web references were verified on January 7th, 2008.