2007

SOA Adoption in Practice - Findings from Early SOA Implementations

Christine Legner
University of St. Gallen, christine.legner@unisg.ch

R. Heutschi
roger.heutschi@unisg.ch

SOA ADOPTION IN PRACTICE - FINDINGS FROM EARLY SOA IMPLEMENTATIONS

Legner, Christine, University of St. Gallen, Müller-Friedberg-Str. 8, CH-9000 St. Gallen, Switzerland, christine.legner@unisg.ch

Heutschi, Roger, University of St. Gallen, Müller-Friedberg-Str. 8, CH-9000 St. Gallen, Switzerland, roger.heutschi@unisg.ch

Abstract

Despite the wide range of advantages which many authors associate with the introduction of a service-oriented architecture (SOA), comprehensive SOA implementations continue to be scarce in practice. Consequently, questions arise as to how the concept is adapted in practice. This paper compares the scientific view of SOA concepts with initial practical experience from first SOA implementations. Based on an SOA model which is derived from recent SOA publications it examines SOA realization in four case studies. From the cross-case analysis, the authors derive three focus areas of SOA adoption, a prioritization of SOA design principles as well as typical steps towards SOA implementation in practice.

Keywords: Service-oriented architecture (SOA), SOA adoption, service design, enterprise architecture

1 INTRODUCTION

Despite the wide range of advantages which many authors associate with the introduction of a service-oriented architecture (SOA), comprehensive SOA implementations continue to be scarce in practice (Mougayar 2005; Wilhelmi & Klesse & Wortmann 2005). In view of the versatility of the concept, the questions arise as to how the concept is adapted in practice and what problems are posed by its use.

The object of this paper is to compare the scientific view of SOA concepts with initial practical experience with SOA implementation. The research questions we want to answer are:

(1) To what extent do they adopt the SOA concept as postulated by the scientific literature?

(2) Which approach do companies take in implementing SOA?

For this purpose the paper first assesses existing research and publications in order to derive the key characteristics of SOA as architectural style (section 2). It then goes on to look at four SOA implementations from practice and analyzes how they adopt SOA concepts. Given the fact that none of the implementations addresses all aspects of the SOA concept, we derive three focus areas of SOA adoption. Each of them is characterized in terms of drivers, goals, benefit potentials and architecture measures for implementing an SOA (section 3). Finally, we delineate the conclusions from our analysis as well as the implications for future research.

In our research, we use a qualitative case study research design as described by (Yin 2002), which has been recommended by several authors as essential for understanding the complex interactions between technology and organizations. The case studies were conducted in 2005 on the basis of literature and document analyses as well as personal interviews and are documented separately

2 SERVICE-ORIENTED ARCHITECTURE

An architecture describes and defines the structure of a system by outlining the basic structures of its system components and their interrelationships as well as giving guidelines for their design and further development (Shaw & Garlan 1996). The understanding of the SOA concept in the context of this paper is based on the definitions of (W3C 2004; van Zyl 2002; Oasis 2005; Gioldasis & Moumoutzis & Kazasis & Pappas & Christodoulakis 2003):

Services represent abstract software elements and/or interfaces which provide other applications with stable, reusable software functionality at an application-oriented, business-related level of granularity using widely applied standards.

An *SOA* is a multiple-layer, distributed information system (IS) architecture which encapsulates parts of the application architecture as services. It can be considered an architectural style which according to (Fielding 2000, 13) is „a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style". The literature is largely agreed on the basic architectural elements of an SOA. However, opinions differ as to the essential SOA design principles.

2.1 Architectural Elements of an SOA

As a multiple-layer integration architecture, an SOA differentiates architectural elements at the following layers (Vogler 2004; Schelp & Schwinn 2005; Erl 2005; Alonso & Casati & Kuno & Machiraju 2003):

- The *application system layer* encompasses software applications which implement the required functionality using their own data pools.

- *Services* structure and encapsulate the data and functions on the application layer according to the requirements of cross-application processes and form a standardized, organization-wide interface and communication layer (Erl 2005). Application domains assume the role of service providers. They group together associated business functions and data on an architecture-wide basis and as far as possible without redundancy (Schelp & Schwinn 2005; Richter & Haller & Schrey 2005). Services communicate by means of messages and are described from the technical and business points of view by means of service specifications. Service specifications are published centrally in a service directory through which potential service consumers can identify suitable services.
- On the *workflow integration layer*, the flow logic of a cross-functional and cross-application business process is defined in the form of an executable process model – a workflow. A workflow represents an automated (sub)process at the moment of execution, which transfers documents, information or tasks from one processing resource (service or human) to the next on the basis of specific rules (WFMC 1999).
- The *desktop integration layer* brings together the business applications required to fulfill tasks in one workplace. It places the emphasis on the viewpoint of an employee role or user role. Portals or composite applications (CA) represent the now typical form of desktop integration. These integrate services for automating individual tasks.

Following (Schelp & Schwinn 2005), it is possible to distinguish between an application-related and an application-neutral view of the architecture components of an SOA (see Figure 1). The application-related view concentrates on components which implement business logic (e.g. applications, workflows). The application-neutral view describes integration mechanisms and infrastructure components without a direct business relationship which provide services and protocols for implementing the system integration.

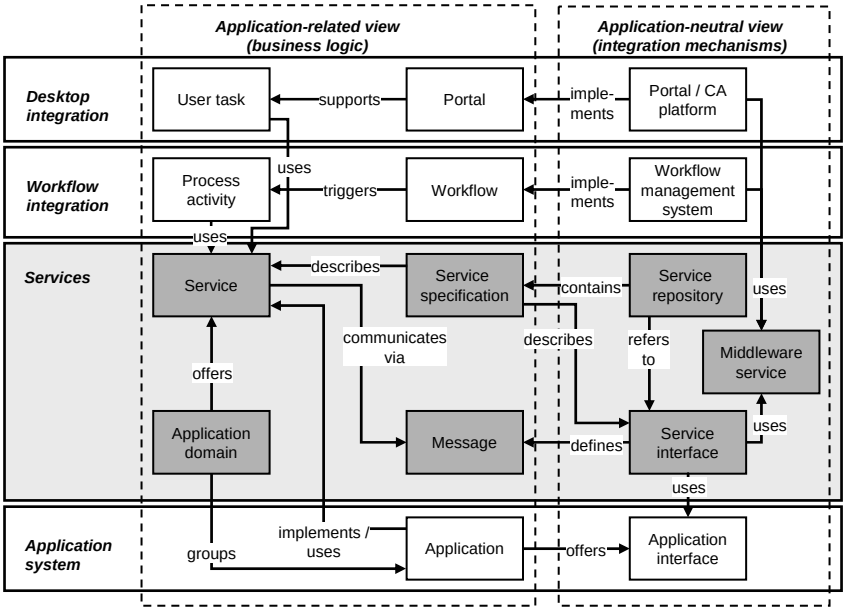

Figure 1 SOA Layers and Architecture Components

2.2 SOA Design Principles

Whereas there is general consent with regard to the architectural components of an SOA, the essential SOA design principles are still under discussion. Table 1 gives an overview of the design principles most frequently stated in SOA publications and groups them into four categories: interface orientation, interoperability, autonomy and modularity, and business suitability.

2.2.1 Interface orientation

Most sources postulate that in services have to abstract from implementation details (W3C 2004) and provide well-defined interfaces described in an implementation-independent manner. Service consumers should not require any information above and beyond the service specification in order to be able to invoke them (Baskerville & Cavallari & Hjort-Madsen & Pries-Heje & Sorrentino & Virili 2005). Thus, a comprehensive service specification not only contains a technical interface description but also describes semantic and dynamic attributes and quality characteristics of a service (Papazoglou & Georgakopoulos 2003). Service interfaces in an SOA represent stable, binding contracts between service providers and users. They are managed in a central repository and are only adapted in clearly defined modification cycles (Klesse & Wortmann & Schelp 2005).

Design Principle		(Baskerville et al. 2005)	(Brown et al. 2002)	(Erl 2005)	(Fritz 2004)	(Klesse et al. 2005)	(McGovern et al. 2003)	(Newcomer & Lomow 2004)	(Papazoglou 2003)	(W3C 2004)
Interface Orientation	Abstraction from service implementation	■	■	■	■		■	■	■	■
	Comprehensive, uniform service specification	■	■	■		■	■	■		■
	Stable, managed service contracts			■	■	■		■		
Inter-operability	Technical standardization	■	■		■		■	■	■	■
	Business standardization		■					■		
	Use of open, widely applied industry standards	■			■			■	■	
Autonomy / Modularity	High service cohesion and weak logical coupling	■		■	■	■	■	■		
	Loosely coupled communication		■	■			■	■	■	
Business Suitability	Service granularity oriented toward business concepts		■				■	■		■
	Generalization of services			■			■	■		
Cells shaded in grey indicate sources which state the respective design principle										

Table 1 Statement of Design Principles by Existing SOA Research

2.2.2 Interoperability

In order to guarantee seamless integration of applications in a heterogeneous environment, an SOA relies on interoperable, standard-based interfaces. Services possess uniform interface descriptions and communicate by means of uniform protocols and data formats (Papazoglou 2003). Although an SOA is not tied to a specific technology, Web services constitute a widely used and highly promising approach to platform- and vendor-independent standards on the transport and communication layer. Some authors postulate that technical standardization has to be complemented by common semantics for business tasks and data (Newcomer & Lomow 2004). For this technical and business standardization, SOAs should, if possible, use open and widely applied industry standards (Fritz 2004).

2.2.3 Autonomy and modularity

An SOA decomposes the existing application architecture and structures it into a manageable number of partially autonomous subsystems, i.e. domains and services. In accordance with well-known principles of module or component design, functions or resources with high interdependency (cohesion) are grouped together in such a way that their logical dependency on other subsystems (loose coupling)

is as low as possible (Vinoski 2005; Papazoglou & Yang 2002). Besides the logical de-coupling, loosely coupled communication reduces runtime dependencies. It can be achieved by means of dynamic service addressing via a logical name (e.g. a uniform resource identifier, URI), asynchronous, message-based communication between service users and providers and stateless service interaction (Kossmann & Leyman 2004; Brown et al. 2002; Erl 2005).

2.2.4 Business suitability

Although service granularity, i.e. the scope of functionality a service exposes, is considered a key design decision within an SOA, there is an ongoing debate as to what extent services should reflect business concepts. Fine-grained services address small units of functionality or exchange small amounts of data. In order to realize complex business scenarios in a distributed environment, coarse-grained services which exchange a larger quantity of data in one operation and support largely complete process activities are said to be more appropriate (McGovern et al. 2003). Services should also be sufficiently generic to allow their reuse in several processes and/or by several users (Newcomer & Lomow 2004).

3 SOA IMPLEMENTATIONS IN PRACTICE

The following sections investigate the status of early SOA implementations on the basis of four examples from practice. Based on the architectural elements and design principles from the previous section we compare how SOA concepts are adopted in practice.

3.1 Selection Criteria

Since comprehensive SOA implementations are still rare, the following criteria were applied for the selection of the four cases (Table 2): Firstly, companies have been involved in SOA projects for at least one year and have documented their target architecture. Secondly, the SOA focus should not be limited to a narrow pilot scenario but encompass major business areas.

Company	Sector	SOA Reach	Status at Time of Investigation
Deutsche Post Brief (DPB)	Logistics	Company-wide, all core business processes	Productive
Credit Suisse (CS)	Finance	Company-wide, all core business processes	Productive
T-Com (TC)	Telecommunications	Company-wide, fulfillment (distribution and production processes)	Under development
Zuger Kantonalbank (ZGKB)	Finance	Company-wide, customer service for account products	Productive

Table 2 Overview of the Case Studies

3.2 Overview on SOA Implementations

Deutsche Post Brief can look back on some six years of experience with implementation of the SOA concept. The poor availability of key information (e.g. customer information), inadequately integrated business processes, high maintenance and operating costs of the IS architecture plus the growing size and risks of IT projects constituted drivers for an SOA. The SOA encompasses the main business processes and applications of the business division Mail. Based on a comprehensive process analysis, the logical restructuring of the application architecture into domains, and the identification of services, the development of the so-called Service Backbone, a centralized technical integration infrastructure, formed an integral part of the SOA implementation. The Service Backbone is based on the J2EE framework and comprises a number of best-of-breed products of different vendors.

	DPB	CS	TC	ZGKB
Desktop Integration	Amongst others, use of services in employee, customer, or distribution partner portals			Support for customer advice processes (opening an account, contact management, etc.) based on SAP enterprise portal and composite application framework
Workflow	No workflow-based service use at the time of the case studies			Support for front/back office communication based on SAP business workflow
Service Layer	• 20-30 services (approx. 100 planned) in 13 domains • Central, standardized integration infrastructure based on J2EE and best-of-breed products • Uniform service specification in central service directory	• Approx. 300 services (approx. 900 service operations) in 20 application domains • Central, standardized integration infrastructure based on CORBA and IBM WebSphere platform • Uniform service specification in central service directory	• Approx. 50-100 services planned, 5 application domains (area of Fulfillment only) • Central, standardized integration based on IBM WebSphere platform • Uniform service specification in central service directory	• Approx. 20 services, majority of them implemented directly on SAP banking application server • No central integration infrastructure, different interface technologies (ABAP objects / SAP RFC, Java RMI) • Uniform service specification, documents stored in server directory

Table 3 SOA Realization in the Case Studies

Like Deutsche Post Brief, the case of *Credit Suisse* represents a mature implementation of a company-wide SOA. In the case of Credit Suisse, an application landscape which had grown over the years and lacked coordination led to an increasing level of complexity. As a consequence, the costs involved in setting up Internet-based customer channels, for example, or replacing a mainframe accounting system by an object-oriented solution would have been unacceptable. Beginning 1998, the company decided to implement a SOA for its entire Swiss banking business and realized it in several phases. CS started off with the definition of 20 core domains, the formulation of architectural guiding principles and the development of a synchronous, CORBA-based service bus. This infrastructure was complemented by a synchronous messaging bus based on the IBM WebSphere platform in 2002 and then enhanced by a bulk transfer infrastructure for large amounts of data.

At the time this analysis was conducted, *T-Com* was still at an earlier stage of implementation. The company was initially concentrating on implementing an SOA for the fulfillment process, with an emphasis on order creation and processing. Repeated internal reorganizations, the expansion of indirect distribution channels and an extensive enlargement of the product and service portfolio provided the starting point for SOA considerations at T-Com. These manifested themselves above all in the areas of order processing and production in the form of redundancies, technical heterogeneity and close dependencies among the supporting information systems. The company started off SOA realization 2005 by defining guidelines for service design, development and use, implementing a central service repository and defining the IBM WebSphere platform as its technical integration base.

Zuger Kantonalbank applied SOA principles when developing an integrated workplace for customer advisors, i.e. within the context of a concrete application development and integration project. The project at ZGKB was triggered by inadequate support for customer advice processes from recently introduced standard software. The user interfaces as well as the functionality, e.g. for opening of accounts or the management of cash cards, were unsuitable for the target group of customer advisors. From 2003 to 2004, the company encapsulated data and functions of backend accounts / payments

systems (mainly ERP applications) as services and composed them into user-centric automated task flows (guided procedures). Technically, the solution is based on SAP's NetWeaver platform.

3.3 Classification of Case Examples in the SOA Model

The examples show that the implementation of SOAs is still in its infancy. None of the companies tackles all architectural elements outlined in chapter 2.1 (see Figure 2). Deutsche Post Brief, Credit Suisse and T-Com placed the main emphasis on developing a domain architecture and implementing a service layer. The latter abstracted from the current applications and was based on a central, standardized integration infrastructure. This means that three out of the four companies are primarily focusing on the application system and service layers. From the outset, the SOA projects of Credit Suisse and Deutsche Post Brief encompassed the company-wide application architecture and/or all core business processes, while the actual implementation of services was prioritized according to current business requirements and performed on a project-by-project basis. T-Com did not want to extend the SOA to further processes and application domains until the next stage.

At Zuger Kantonalbank, the externalization of process logic from existing applications and the composition of services in task flows was the main focus. Although existing application functions were also packaged as services and described in a central directory, the organization initially decided against implementing a platform-independent service layer with standardized interface technologies based on a central middleware infrastructure (an enterprise service bus). Since a large part of the services are implemented by one application (SAP Banking), the definition of application domains also played a minor role. Within this analysis, ZGKB was the only company starting SOA implementation within a concrete application development and integration project.

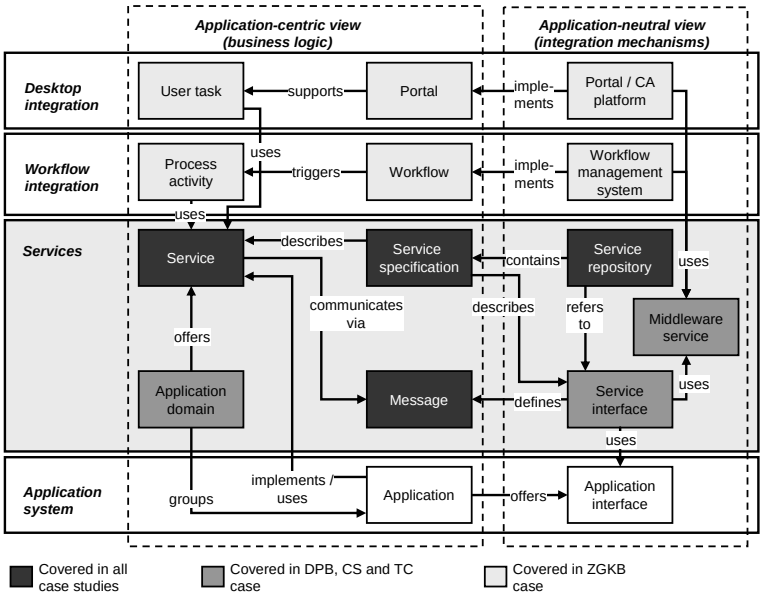

Figure 2 Architectural Elements Covered by the Case Studies

3.4 Focus of SOA Adoption

By adopting SOA, the four companies were striving to raise the efficiency and effectiveness of IT use and increase the capability of existing applications to adapt in line with new business requirements. Three objectives for the adoption of SOAs emerge from the case studies considered (see Table 4):

- As a *standardized integration infrastructure*, an SOA increases the technical connectivity of heterogeneous applications, reduces the diversity of interface technologies and therefore cuts

integration and maintenance costs. In view of the growing number of heterogeneous application platforms, operating systems, integration infrastructures and development tools, Deutsche Post Brief, Credit Suisse and T-Com pursued the goal of standardizing the integration of existing application systems and establishing a cross-platform integration infrastructure for this purpose. This is comparable with well-known objectives of Enterprise Application Integration (EAI).

- The use of SOAs for *decoupling application domains* is aimed at reducing dependencies and redundancies between existing applications. Credit Suisse, Deutsche Post Brief and T-Com decrease project risks as well as costs for the new or further development of applications by dividing their application landscapes into domains which are structured from a business point of view and eliminating redundancies. Domains offer reusable services, and are autonomously governed and developed. The decoupling of the domain architecture is not such a dominant feature in the case of ZGKB, which is targeted at improving end-user support in a very specific business process.
- Companies pursue SOA projects for *flexible user and/or business process integration* with the aim of simpler and faster adaptation or new development of cross-application processes. All the companies investigated integrate services in portals in order to provide better support for user processes. At the time of conducting the case studies, however, externalizing flow and control logic in the form of workflows and/or task flows was only a key objective of the SOA in the case of Zuger Kantonalbank. As a rule, the other companies implemented service composition logic hard-coded within the respective applications. Although workflow systems were also either in use or planned at Credit Suisse, Deutsche Post Brief and T-Com at the time, service orchestration on the basis of workflows was a measure which these companies would not be emphasizing until a later date.

Primary Focus of SOA Adoption	**Related Benefits**	**DPB**	**CS**	**TC**	**ZGKB**
SOA as standardized integration infrastructure					
Uniformly documented and managed interfaces	• Better transparency in respect of dependencies and/or interfaces in the application architecture • Easier understanding of interfaces	■	■	▭	
Cross-platform integration and technology reduction	• Decrease in costs for system integration through standardization of interfaces from a technical and business point of view • Reduced operating costs through harmonization of platforms and technologies as well as concentration of technical skills	▭	■	▭	
SOA for decoupling application domains					
Local restriction of changes in the application architecture	• Better manageability and separability of IT projects through domain decoupling • Clearly regulated responsibilities for business functions and data	▭	▭	▭	
Reuse and reduction in redundancy	• Decrease in development and operating costs through reuse and reduction in redundancy • Shortening of project durations and better time-to-market through stronger reuse	▭	■	▭	■
SOA for flexible user / business process integration					
Decoupling of process and application systems alterations	• Simplified communication between IT and business areas due to common terminology and easier mapping of process models to the application architecture • Faster process adaptation by separating stable business logic from dynamic process logic	▭	■	▭	■
Improvement of process support	• Faster development of user interfaces specific to a role or access channel • Faster realization of workflows coordinating existing application functionality	▭	▭	▭	■
■ realized ▭ realized in part or envisaged					

Table 4 Primary Focus of SOA Adoption and Related Benefits

As a result of their advanced implementation of the SOAs, Deutsche Post Brief, Credit Suisse and Zuger Kantonalbank have already been able to observe the benefit potentials in day-to-day IT operations and in individual development projects, and to draw quantitative or qualitative conclusions. T-Com, on the other hand, is still at an early stage of implementation, and the envisaged benefit potentials are not yet verifiable.

3.5 Relevance of SOA Design Principles

Within their SOA projects, the investigated companies formulated architectural design principles which we compared to the set of design principles outlined in section 2.2. Table 5 summarizes the results of this comparison. The companies which implemented the SOA for a larger area of the application architecture applied these principles to a greater extent. They also considered a business-semantic standardization of the service interfaces to be important, despite the rare mention of this design principle in literature as shown in Table 1. In contrast with the other case examples, Zuger Kantonalbank implemented the SOA for a small and homogeneous area of the application architecture. Both the workplace for customer advisors as service user and the banking system which implemented the majority of the services used were based on the SAP product range and the number of services developed was low. As a consequence, the abstraction from the technical and business service implementation, management of the service contracts, and the technical and semantic interface standardization were of lesser importance. However, this is set to grow in the future with the further expansion of the SOA.

SOA Design Principles	**DPB**	**CS**	**TC**	**ZGKB**
Abstraction from service implementation	■	■	■	□
Comprehensive service specification	■	■	■	■
Stable, managed service contracts	■	■	■	□
Technical standardization	■	■	■	□
Business standardization	■	■	■	
Use of open and widely used industry standards	□	□	□	
High service cohesion and weak logical coupling	■	□	■	■
Loosely coupled communication	□	□	□	□
Service granularity oriented toward business concepts	■	■	■	■
Generalization of services	■	□	■	■
■ applied □ partly applied				

Table 5 Application of the SOA Design Principles

Comprehensive use of industry standards was not found in any of the case examples. The reason for this is that standards which encompass different manufacturers and technology platforms are either currently non-existent or have not yet reached an acceptable level of maturity. Where technical standards are concerned, Deutsche Post Brief goes the furthest. The company makes sure that only J2EE standards are used in the integration infrastructure and avoids manufacturer-specific extensions in order to avoid reliance on one particular software supplier as far as possible. The other companies pursue a strategy of obtaining the integration platform from one manufacturer and also using manufacturer-specific extensions, at least in some areas of the integration infrastructure. Loosely coupled communication between services has also not been consistently implemented for all services in any of the cases. Despite the fact that all the companies strive for dynamic service addressing and stateless

service interaction, service and service user nonetheless frequently communicate by means of synchronous mechanisms for reasons of performance.

3.6 Step-wise Approach towards SOA Adoption

The examined companies took a stepwise approach to SOA implementation comprising three groups of activities: SOA projects usually started with changes in IT organization and governance which were complemented by the formulation of architectural guidelines. These two sets of measures were prerequisites to the development of services which is usually subject of application development and integration projects.

3.6.1 Organization and governance

In order to establish SOA architectural principles in the organization, Deutsche Post Brief, Credit Suisse and T-Com defined new *architectural roles and competencies* or extended the tasks of existing roles respectively. To this end, the companies appointed central architecture boards and specific roles which define and communicate SOA principles and supervise their enforcement in IS projects. These central boards in a next step defined the *objectives* of and the *areas of application for SOA.* The development of the SOA is governed using metrics to *measure the outcome of the architectural programs and principles*. Credit Suisse in particular defined a comprehensive SOA-specific architecture scorecard to measure its progress periodically. Deutsche Post Brief also uses metrics like the rate of changes in a service interface or service reuse in order to measure architecture quality.

3.6.2 Formulation of architectural principles

The specific objectives which the companies pursued with their SOA implementation formed the basis for the formulation of *architectural principles*. These comprise guidelines as to when and how to develop services, standardized development and review processes, or principles for the service design amongst others.
By designing a *domain architecture*, the companies structured their application architectures from a business point of view and thereby supported the decisions where and by who services are to be developed: Interfaces between applications of different domains are to be implemented as services, whereas alternative coupling mechanisms are allowed to integrate applications within a domain. The domain architecture serves as a long-term plan for the future development of the application architecture.
The central boards and architects also decided on the *architecture of the technical SOA-infrastructure* (which central integration capabilities to support, which platforms and standards to use) and defined corresponding technical architectural principles.

3.6.3 SOA realization in application development and integration infrastructure projects

Apart from Zuger Kantonalbank all companies implemented a central integration infrastructure for the service layer. This infrastructure standardizes service interfaces, offers central integration mechanisms (repository, message bus etc.) and forms the basis for a simple and platform neutral usage of services. Whereas early SOA projects often developed their infrastructures in a best-of-breed approach combining products from several vendors, companies increasingly use comprehensive SOA platforms from a single vendor (e.g. IBM WebSphere or SAP NetWeaver).
All of the examined companies gradually developed and reused services within the scope of business-driven application development and integration projects. Differences can be observed in their approach of identifying potential services, however: While Credit Suisse decides by means of formalized project reviews and architectural principles within each development project whether certain functionality is to be implemented as a service, Deutsche Post Brief or T-Com conducted architecture-wide analyses to plan each domain’s service portfolio in advance.

4 SUMMARY AND OUTLOOK

In a fully-fledged service oriented architecture, application functionality is available as a service and can easily be combined and rearranged in workflows and task flows in order to flexibly support cross-application business processes and the needs of end-users. The analysis of early SOA implementations shows that in practice even companies which have been gaining experience with SOA implementation for several years do not address all aspects of the SOA concept in parallel. Instead, they tend to select a focused approach to SOA adoption. From the case studies, three focus areas of SOA adoption can be derived, namely (1) SOA as standardized integration infrastructure, (2) SOA for decoupling application domains and (3) SOA for flexible user / business process integration. Each of these focus areas of SOA adoption is characterized by a set of specific objectives and related benefits from the company perspective. These benefits may either relate to the costs of application integration (in the case of SOA as standardized integration infrastructure), to the manageability of application development and operations (in the case of SOA for decoupling application domains) or to business benefits related to faster realization of IS support (in the case of SOA for flexible user / business process integration). The focus area of SOA adoption also has significant implications on the applied architectural principles and the measures for SOA implementation. The object of further research will be to validate these areas of SOA adoption based on a larger number of cases. Furthermore, these focus areas need to be complemented by suggestions for detailed design principles and metrics for evaluating service design and the SOA as a whole.

With regard to the necessary steps toward SOA adoption, three main set of measures have been identified from the four case studies: (1) Introduction of new roles and processes related to IT organization and governance, (2) formulation of architectural guidelines and (3) SOA realization in application development and integration infrastructure projects. Our investigation suggests that with the growing expansion of the objectives and fields of use for SOAs, consistent architecture management will become increasingly important. Alignment of the IS architecture with the business process architecture will be necessary for the business-oriented design of SOAs with the aim to flexibly adapt to information system design to business requirements. Future work should therefore be conducted on comprehensive enterprise architecture models as well as architecture management and IS development methods which incorporate service-based concepts.

Acknowledgement

This research has been conducted in the context of the Competence Center Business Networking 3 (CC BN3) and the research program “Business Engineering” at University of St. Gallen, Switzerland. The authors would like to thank Degussa AG, ETA SA, Fraport AG, LBA, SAP AG and T-Com for their cooperation and funding.

References

Alonso, G., Casati, F., Kuno, H. and Machiraju, V. (2003). Web Services: Concepts, Architectures and Applications, Springer, Berlin.

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M. and Virili, F. (2005). Extensible architectures: The strategic value of service-oriented architecture in banking. In Proceedings of European Conference on Information Systems (ECIS) (Bartmann, D. and Rajola, F. and Kallinikos, J. and Avison, D. and Winter, R. and Ein-Dor, P. and Becker, J. and Bodendorf, F. and Weinhardt, C., Eds).

Brown, A., Johnston, S. and Kelly, K. (2002). Using service-oriented architecture and component-based development to build web service applications. Rational Software Corporation.

Erl, T. (2005). Service-Oriented Architecture. Prentice Hall, New York.

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine.
Fritz, F.-J. (2004). An Introduction to the Principles of Enterprise Services Architecture (ESA). SAPInsider April-June 2004.
Gioldasis, N., Moumoutzis, N., Kazasis, F., Pappas, N. and Christodoulakis, S, (2003). A Service Oriented Architecture for Managing Operational Strategies, In Proceedings of ICWS-Europe 2003, LNCS 2853, 11-23.
Klesse, M., Wortmann, F. and Schelp, J. (2005). Erfolgsfaktoren der Applikationsintegration. Wirtschaftsinformatik 47 (4), 259-267.
Kossmann, D. and Leymann, F. (2004). Web Services. Informatik-Spektrum 27 (2), 117-128.
McGovern, J., Tyagi, S., Stevens, M. and Mathew, S. (2003). Service-Oriented Architecture. In Java Web Services Architecture (Mcgovern, J. and Tyagi, S. and Stevens, M. and Mathew, S., Eds), 35-63, Morgan Kaufmann, San Francisco.
Mougayar, W. (2005). The SOA in IT Benchmark Report - What CIOs Should Know about How SOA Is Changing IT. Aberdeen Group, Boston.
Newcomer, E. and Lomow, G. (2004). Understanding SOA with Web Services. Addison-Wesley, Maryland.
OASIS (2005). Reference Model for Service Oriented Architectures, Working Draft 09. OASIS Open.
Papazoglou, M. P. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003) (IEEE Computer Society, Ed.), 3-12, Rome, Italy.
Papazoglou, M. P. and Georgakopoulos, D. (2003). Service-Oriented Computing, In Communications of The ACM 46 (10), 25-28
Papazoglou, M. P. and Yang, J. (2002). Design Methodology for Web Services and Business Processes. In Technologies for E-Services : Third International Workshop, TES 2002 (Buchmann, A. C., F.; Fiege, L.; Hsu, M.-C.; Shan, M.-C., Ed), 54-64, Springer, Berlin.
Richter, J.-P., Haller, H. and Schrey, P. (2005). Serviceorientierte Architektur. Informatik Spektrum 28 (5), 413-416.
Schelp, J. and Schwinn, A. (2005): Extending the Business Engineering Framework for Application Integration Purposes, In Proceedings of the 2005 ACM Symposium on Applied Computing, 1333-1337, ACM Press, New York.
Shaw, M. and Garlan, D. (1996). Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, New Jersey.
Turowski, K. (2001) Spezifikation und Standardisierung von Fachkomponenten. Wirtschafts-informatik 43 (3), 269-281.
Van Zyl, J. (2002). A perspective on service based architecture. In Proceedings of SAICSIT, 249.
Vinoski, S. (2005). Old Measures for New Services, IEEE Internet Computing 9 (6), 72-74.
Vogler, P. (2004). Prozess- und Systemintegration: Evolutionäre Weiterentwicklung bestehender Informationssysteme mit Hilfe von Enterprise Application Integration. Habilitation, University of St. Gallen.
W3C (2004). Web Services Architecture, W3C Working Group Note 11 February 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/.
WFMC (1999). Terminology & Glossary. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.
Wilhelmi, C., Klesse, M. and Wortmann, F. (2005). Service-orientierte Architektur - Der richtige Weg? Working Paper, University of St. Gallen.
Yin, R. K. (2002). Case Study Research. Design and Methods. Sage Publications, London.