

# Serviceorientierte Architekturen: Vom Konzept zum Einsatz in der Praxis

Roger Heutschi, Christine Legner, Hubert Österle

Institut für Wirtschaftsinformatik  
Universität St. Gallen  
Müller-Friedberg-Strasse 8  
9000 St. Gallen  
Roger.Heutschi@unisg.ch  
Christine.Legner@unisg.ch  
Hubert.Oesterle@unisg.ch

**Abstract:** Serviceorientierte Architekturen (SOA) werden gegenwärtig als Ansatz für eine einfachere und flexiblere inner- und zwischenbetriebliche Integration heterogener Anwendungssysteme diskutiert. Dieser Beitrag entwickelt ein SOA-Architekturmodell und untersucht seine Ausprägung in vier Praxisbeispielen. Er identifiziert aus den Fallstudien ausserdem typische Architekturtreiber, Einsatzziele, Nutzenpotenziale und Architekturmassnahmen einer SOA-Umsetzung in der Praxis.<sup>1</sup>

## 1 Einleitung

Obwohl das Konzept serviceorientierter Architekturen (SOA) seit ca. zwei Jahren intensiv diskutiert wird und die grossen Softwarehersteller ihre Lösungen daran ausrichten, sind die Erfahrungen mit der Umsetzung in den Unternehmen noch beschränkt [WKW05; Mo05]. Angesichts der Mächtigkeit des Konzeptes stellt sich aus Sicht der Unternehmen die Fragen, welchen Neuigkeitsgrad gegenüber herkömmlichen Architekturansätzen eine SOA bietet, für welche Problemstellungen sich ihr Einsatz eignet oder welche Fähigkeiten und Architekturmanagement-Massnahmen eine SOA innerhalb des IT-Bereichs erfordert [Ha05, 10; Qu04].

---

<sup>1</sup> Diese Arbeit entstand im Rahmen des Kompetenzzentrums Business Networking 3 des Instituts für Wirtschaftsinformatik der HSG (Universität St. Gallen) als Teil des Forschungsprogrammes Business Engineering HSG.

Ziel dieses Beitrags ist, die SOA-Konzepte aus Sicht der Wissenschaft den ersten Praxiserfahrungen bei SOA-Umsetzung gegenüberzustellen. Dazu arbeitet der Beitrag zunächst das SOA-Begriffsverständnis aus der Literatur auf und entwickelt daraus ein Architekturmodell, das SOA anhand von Architekturebenen, Gestaltungskomponenten und definierter Designprinzipien beschreibt (Kap. 2). Er stellt dem Architekturmodell vier SOA-Umsetzungen aus der Praxis gegenüber und identifiziert als Ergebnis Treiber, Ziele, Nutzenpotenziale und Architekturmassnahmen einer SOA (Kap. 3).

Der Beitrag verwendet als Forschungsansatz die qualitative Fallstudienforschung (vgl. Yi94). Die Erhebung und Strukturierung der Fallstudien basiert auf der Projektmethode Business Engineering Case Studies (PROMET BECS) von [SÖ04]. Die Fallstudien wurden 2005 anhand von Literatur- und Dokumentenanalysen sowie persönlichen Interviews erhoben und sind separat dokumentiert.

## **2 Architekturmodell**

Architekturen unterstützen die Realisierung physischer oder logischer Systeme, indem sie die grundlegenden Strukturen ihrer Systemkomponenten und ihre Beziehungen zueinander beschreiben sowie Richtlinien für ihr Design und ihre Weiterentwicklung geben [SG96]. Das im Rahmen dieser Arbeit verwendete Begriffsverständnis einer SOA orientiert sich an den Definitionen von [W304; Oa05; Zy02]:

*Services* stellen abstrakte Software-Elemente bzw. Schnittstellen dar, die anderen Applikationen unter Nutzung von verbreiteten Standards stabile, wieder verwendbare Software-Funktionalität auf einer anwendungsorientierten, fachlichen Granularitätsstufe anbieten.

Eine *SOA* ist eine mehrschichtige, verteilte Informationssystem (IS)-Architektur, die Teile der Applikationsarchitektur als *Services* kapselt und dabei eine Reihe von Designprinzipien berücksichtigt. Die Literatur ist sich über die grundlegenden Architekturkomponenten in einer SOA weitgehend einig. Die Meinungen bezüglich der wesentlichen SOA-Designprinzipien sind aber geteilt.

### **2.1 SOA-Architekturkomponenten**

SOA erweitert bestehende Ansätze zur Gestaltung von Informationssystemen wie bspw. Client/Server- oder komponentenorientierte Architekturen (vgl. Hasselbring 2006, S. 49, Mertens 2004, S. 315). Im folgenden werden die Architekturkomponenten einer SOA (s. Abb. 1) in einem Architekturmodell eingeordnet, das auf den Modellen von [Vo04; Sc04; Er05] basiert. Es beschreibt eine SOA als mehrschichtige Integrationsarchitektur,

welche die fachliche Prozessarchitektur mit der Applikationsarchitektur verbindet, und unterscheidet folgende Ebenen:<sup>2</sup>

- Die Ebene *Anwendungssystem* umfasst betriebliche Anwendungssysteme, die Applikationsfunktionen und Datensammlungen implementieren [Vo04]. Sie repräsentiert in diesem Modell die Ressourcensicht, welche die im IS vorhandene, potenziell zur Unterstützung betrieblicher Aktivitäten nutzbare, fachliche Funktionalität betrachtet.
- *Services* strukturieren und kapseln die auf Ebene Anwendungssystem vorhandenen Daten und Funktionen nach den Anforderungen systemübergreifender Prozesse und bilden eine organisationsweit standardisierte Schnittstellen- und Kommunikationsschicht [Er05]. Services kommunizieren über Nachrichten und werden durch Servicespezifikationen technisch und fachlich beschrieben. Servicespezifikationen werden an zentraler Stelle, in einem Serviceverzeichnis, veröffentlicht, über das potenzielle Nutzer geeignete Services identifizieren können. Applikationsdomänen gruppieren architekturweit fachlich zusammengehörige Funktionen und Daten möglichst redundanzfrei und kapseln sie über Services nach aussen [Sc04]. Der Zugriff der Serviceschnittstellen auf die Funktionalität der Ebene Anwendungssystem erfolgt dabei unter Nutzung von Middlediensten zur Funktions-, Daten- oder Präsentationsintegration. Solche Middledienste stellen bspw. Application Server oder Message Broker zur Verfügung [Al03].
- Die Ablaufsteuerung eines rollen- und applikationsübergreifenden Geschäftsprozesses ist Gegenstand der *Ebene Workflowintegration*. Im Rahmen einer Workflowintegration ist die Ablauflogik des Integrationssystems in Form eines ausführbaren Prozessmodells – eines Workflows – definiert. Ein Workflow stellt zur Ausführungszeit einen automatisierten (Teil-) Prozess dar, der Dokumente, Informationen oder Aufgaben anhand bestimmter Regeln von einer bearbeitenden Ressource zur nächsten weiterleitet [WF99].
- Die *Ebene Desktopintegration* bringt die zur Erfüllung von Aufgaben notwendigen betrieblichen Anwendungen an einem Arbeitsplatz zusammen. Sie stellt die Sichtweise einer Mitarbeiter- bzw. Benutzerrolle in den Vordergrund. Portale oder Composite Applications (CA) stellen die heute typische Form der Desktopintegration dar. Für den Steuer- und Datenfluss zwischen Arbeitsschritten einer desktopintegrierten Applikation verwendet [Vo04], den Begriff Taskflow.

Bei der Beschreibung der Ebenen unterscheidet das Modell in Anlehnung an [Sc04], eine anwendungsbezogene Sicht und eine anwendungsneutrale Sicht. Die anwendungsbezogene Sicht konzentriert sich auf Architekturkomponenten, die fachliche Logik umsetzen (Applikationslogik, Workflows etc.). Die anwendungsneutrale Sicht beschreibt Integrationsmechanismen und Infrastrukturkomponenten ohne unmittelbaren fachlichen Bezug, die Dienste und Protokolle zur Implementierung der Systemintegration bereitstellen.

---

<sup>2</sup> Ein detaillierteres Modell mit ausführlicher Beschreibung der Ebenen, Architekturkomponenten und ihrer Beziehungen zueinander findet sich in [Sc06].

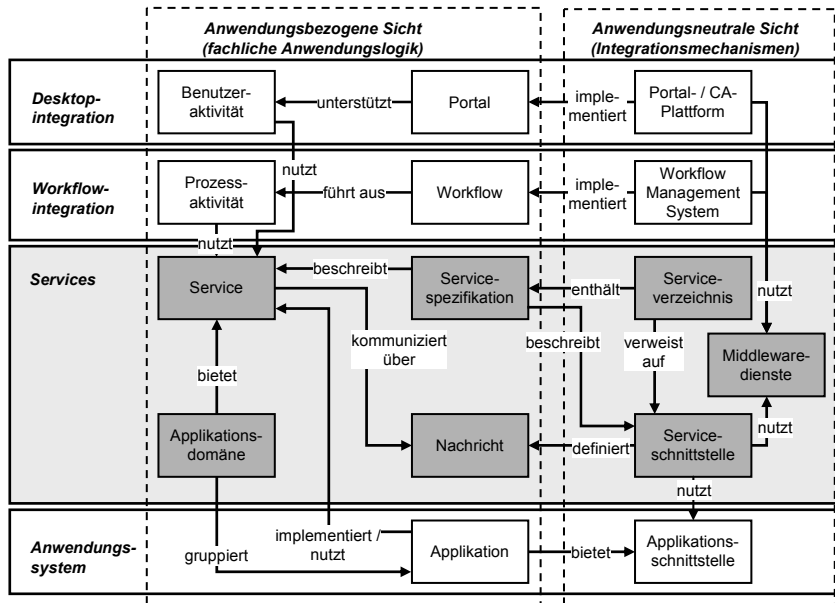


Abbildung 1: SOA-Ebenen und Architekturkomponenten

## 2.2 SOA-Designprinzipien

Neben den Architekturkomponenten werden in der Literatur eine Reihe von Designprinzipien genannt, die eine SOA erfüllen sollte. Eine Herausforderung bei der Analyse der Literatur besteht darin, dass

- unterschiedliche Autoren verschiedene Designprinzipien betonen,
- sie die Designprinzipien verschieden detailliert definieren,
- sich mit unterschiedlichen Begriffen bezeichnete Prinzipien inhaltlich überlappen oder
- inhaltlich gleiche Prinzipien unterschiedlich benannt werden.

Der folgende Abschnitt nimmt daher eine Systematisierung und Begriffsklärung zu den von Wissenschaft, Standardisierungsgremien und Beratung meistgenannten Designprinzipien vor. Tabelle 1 gibt einen Überblick über die teils explizit, teils implizit genannten Designprinzipien in den verwendeten Quellen.

Designprinzip	[Ba05]	[BJK02]	[Er05]	[Fr04]	[KWS05]	[Mc03]	[NL04.]	[Pa03]	[W304]
Abstraktion von der Serviceimplementierung									
Umfassende, einheitliche Servicespezifikation									
Stabile, gemanagte Servicekontrakte									
Technische Standardisierung									
Fachliche Standardisierung									
Verwendung offener und verbreiteter Industriestandards									
Hohe Servicekohäsion und schwache logische Kopplung									
Lose gekoppelte Kommunikation									
An Geschäftskonzepten orientierte Servicegranularität									
Generalisierung der Serviceleistung									
Quellen mit grau gefüllte Zellen führen das jeweilige Designprinzip auf									

Tabelle 1: Nennung der Designprinzipien in den untersuchten Quellen

### 2.2.1 Abstraktion von der Serviceimplementierung

Services sind abstrahierte, logische Sichten auf Softwareelemente [W304]. Eine Abstraktion von der Implementierungslogik wird erreicht, indem Services gemäss dem Prinzip des „Information Hiding“ [Pa72] Details der Serviceimplementierung wie z.B. interne Datenstrukturen oder Algorithmen verbergen. Durch eine programmiersprachen-, plattform- und middlewareneutrale Servicebeschreibung [Do05b] abstrahieren Service-schnittstellen zur Design- und Entwicklungszeit auch von der technischen Implementierung eines Services.

### **2.2.2 Umfassende, einheitliche Servicespezifikation**

Servicebenutzer sollten keine über die Servicespezifikation hinausgehenden Informationen benötigen, um die Servicefunktionalität verwenden zu können [Ba05]. Eine umfassende Servicespezifikation enthält nicht nur eine technische Schnittstellenbeschreibung (Netzwerkprotokolle, Nachrichtenformate, Netzwerkadresse etc.) sondern beschreibt einen Service auch auf Verhaltensebene (Vor- / Nachbedingungen und Invarianten der Servicebenutzung), Abstimmungsebene (Dialogsequenzen, Synchronisationserfordernisse etc.), Aufgabenebene (Beschreibung in geschäftlicher Terminologie), Qualitätsebene (z.B. Service Level Agreements, SLAs) und Vermarktungsebene (z.B. organisatorische Verantwortung, Nutzungskonditionen) [Tu02].

### **2.2.3 Stabile, gemanagte Servicekontrakte**

Serviceschnittstellen in einer SOA sind stabile, verbindliche Kontrakte zwischen Serviceanbietern und Nutzern. Sie stellen eigenständige, gemanagte Architekturelemente dar, die in einem zentralen Repository verwaltet und nur in klar definierten Änderungszyklen über gesteuerte Prozesse und Qualitätschecks sowie unter Berücksichtigung der bestehenden Nutzer angepasst werden [KWS05].

### **2.2.4 Technische Standardisierung**

Services besitzen einheitliche Schnittstellenbeschreibungen und kommunizieren über einheitliche Protokolle und Datenformate [Pa03]. Da Services i.d.R. in unterschiedlichen Technologien implementiert sind, ist eine gemeinsame Middlewareinfrastruktur notwendig, die einheitliche technische Standards definiert. Sie stellt zentrale Dienste für die Serviceentwicklung, -publikation und -nutzung zur Verfügung und übersetzt plattform-spezifische interne Datenformate und Kommunikationsprotokolle in die gemeinsamen Formate und Protokolle auf Serviceebene [Mc03].

### **2.2.5 Fachliche Standardisierung**

Einheitliche plattformübergreifende Technologien erhöhen nur die technische Interoperabilität von Services. Um die Wiederverwendung von Services und die Komposition von Services in unterschiedlichen Anwendungskontexten zu erleichtern, sollten SOA auch Vorgaben bezüglich semantischer Schnittstelleneigenschaften formulieren [NL04]. Fachliche Standards [Tu01] bzw. „Domänenstandards“ [SGM02] definieren einheitliche Begriffssysteme und standardisieren Semantiken für betriebliche Aufgaben und Daten. Beispiele für solche Standards in einer SOA reichen von einheitlichen Datenmodellen und -wertebereichen für Geschäftsentitäten bis hin zu festgelegten Codes für Fehlermeldungen.

### **2.2.6 Verwendung offener und verbreiteter Industriestandards**

SOA sollten für die technische und fachliche Standardisierung wenn möglich offene und verbreitete Industriestandards verwenden [Fr04]. [ID04] bezeichnet einen Standard dann als offen, wenn er von einer non-profit Organisation in einem allen interessierten Parteien offen stehenden Abstimmungsprozess veröffentlicht und weiterentwickelt wird, frei oder über eine nominelle Gebühr erhältlich ist und beliebig vervielfältigt und genutzt werden kann. Während Web Services einen weit verbreiteten und viel versprechenden Ansatz für hersteller- und plattformunabhängige Standards auf technischer Ebene darstellen, sind fachliche bzw. Domänenstandards weniger verbreitet bzw. in Softwareprodukten umgesetzt [Al04]. Gründe dafür sind u.a. die für eine Standardisierung zu hohe Änderungsgeschwindigkeit bzw. Variantenvielfalt der betrieblichen Anwendungsdomäne oder ungünstige Machtkonstellationen in Geschäftsnetzwerken.

### **2.2.7 Hohe Servicekohäsion und schwache logische Kopplung**

Kohäsion und Kopplung sind Masse, welche die Abhängigkeiten bzw. logische Zusammengehörigkeit von Daten und Funktionen bewerten. Die Servicekohäsion betrachtet die "Innenbeziehung" der von einem Service gekapselten Funktionen und Daten und bezeichnet den Grad, in dem sie dazu beitragen, einen einzigen, wohl definierten Zweck zu erfüllen [PY02]. Ein Service kontrolliert möglichst alle und nur die Funktionen und Daten, die er zur Unterstützung einer eigenständigen, logisch abgeschlossenen Aufgabe benötigt. Parallel zur Servicekohäsion wird in der Beziehung zwischen Services bzw. zwischen der durch Services gekapselten Fachlogik, eine schwache logische Kopplung angestrebt. Eine logische Kopplung besteht dann, wenn Services bzw. die sie implementierenden Softwarelemente ein gleiches Änderungsverhalten aufweisen, d.h. aufgrund gleicher geschäftlicher Anforderungen verändert werden müssen [GH98]. Dies ist etwa dann der Fall, wenn sie direkt auf gemeinsame Daten zugreifen, Funktionalität redundant implementieren oder Vererbungsbeziehungen untereinander aufweisen [NL04].

### **2.2.8 Lose gekoppelte Kommunikation**

Die lose gekoppelte Kommunikation zwischen Services und Servicenutzern ist ein Hilfsmittel zur Reduktion von Abhängigkeiten zur Laufzeit. Sie vermindert die Abhängigkeit vom Ort und von der Verfügbarkeit der Gegenpartei und verringert den Wiederaufsetzungsaufwand bei Systemausfällen. Eine lose Kopplung kann erreicht werden durch eine dynamische Serviceadressierung über einen logischen Namen (z.B. Uniform Resource Identifier, URI), eine asynchrone Kommunikation zwischen Servicenutzer und Service oder eine statuslose Serviceinteraktion [Er05; BJK02]. Servicenutzer und Service interagieren dann statuslos, wenn ein Serviceaufruf vom vorherigen unabhängig ist und der Service zwischen Aufrufen keine Interaktionsinformationen speichert.

Die lose gekoppelte Kommunikation nach diesen Prinzipien ist eine anzustrebende, aber keine zwingende Eigenschaften von SOA. Bspw. können Geschwindigkeitsanforderungen an den Serviceaufruf aus einer Benutzeroberfläche eine synchrone, RPC-ähnliche Kommunikation erforderlich machen [NL04].

### 2.2.9 An Geschäftskonzepten orientierte Servicegranularität

Aus Sicht des fachlichen Servicedesigns sind primär zwei Granularitätsarten von Bedeutung [Mc03]:

- Die Funktionsgranularität ist ein Mass dafür, wie viel Fachlogik ein Service umfasst [Gr98]. Ein Service mit grober Funktionsgranularität kapselt viel Fachlogik, d.h. er führt in einem Aufruf bzw. in einem Interaktionsschritt viele Verarbeitungsschritte durch.
- Die Schnittstellengranularität bezeichnet den Umfang des Datenmodells einer Serviceschnittstelle, d.h. die Anzahl Attribute, welche die zwischen Service und Servicenutzern ausgetauschten Nachrichten enthalten.

In verteilten Systemen ist es i.d.R. effizienter, viele Daten in einem Durchgang auszutauschen, als mehrere Serviceaufrufe auszuführen, die kleine Nachrichten mit wenig Daten liefern [Mc03]. Von Services angebotene Operationen sollten deshalb geschäftlichen Konzepten entsprechen, d.h. möglichst komplette Teilprozesse oder Prozessaktivitäten unterstützen bzw. Informationen zu kompletten Geschäftsentitäten anstatt einzelne Attribute lesen und bearbeiten.

Eine grobe Servicegranularität bringt aber auch Nachteile mit sich. Grosse zu übertragende Datenvolumina verlängern die Antwortzeiten eines Serviceaufrufs und damit die Wartezeiten der Servicenutzer. Services, die viel Fachlogik umfassen, sind ausserdem anfälliger auf durch neue fachliche Anforderungen ausgelöste Anpassungen [SW05] und weisen oft einen höheren Spezialisierungsgrad auf, was ihre Wiederverwendbarkeit reduziert [WV04].

### 2.2.10 Generalisierung der Serviceleistung

Der Grad an Generalisierung bzw. Spezialisierung eines Services bestimmt massgeblich seine Wiederverwendbarkeit durch unterschiedliche Nutzer. Der Spezialisierungsgrad eines Services kann durch seinen Anwendungskontext gegeben sein. Z.B. ist ein Service zur Bonitätsprüfung von Kunden in weniger Geschäftsprozessen einsetzbar als ein Service, der die Geschäftsentität „Kunde“ bearbeitet und liest. Er lässt sich aber auch durch das Servicedesign beeinflussen. Massnahmen zur Generalisierung von Services sind bspw. eine gröbere Schnittstellengranularität oder eine Verfeinerung der Funktionsgranularität [WV04] oder die Unterstützung unterschiedlicher Interaktionsstile (z.B. "Request/Response"-Stil und "Publish/Subscribe"-Stil) [NL04]. Die Generalisierung von Serviceleistungen erhöht zwar deren Wiederverwendbarkeit, ist aber auch mit steigenden Entwicklungs- und Nutzungskosten verbunden [Do05a].

### 3 SOA-Umsetzungen in der Praxis

Die folgenden Abschnitte untersuchen die SOA-Umsetzungen in vier Praxisbeispielen.<sup>3</sup> Diese befinden sich in verschiedenen Umsetzungsstadien und besitzen unterschiedliche Gestaltungsreichweiten (s. Tab. 2): Deutsche Post Brief und Credit Suisse stellen reife Umsetzungen einer unternehmensweiten SOA dar, die wesentliche Kerngeschäftsprozesse abdecken. T-Com befindet sich noch in einem frühen Umsetzungsstadium und konzentriert sich in einem ersten Schritt auf die Umsetzung einer SOA für den Prozessbereich Fulfillment, d.h. der Kundenauftragsabwicklung. Die Zuger Kantonalbank setzt SOA-Prinzipien für die servicebasierte Entwicklung eines integrierten Arbeitsplatzes für Kundenberater (BAP) ein, d.h. für ein konkretes Applikationsentwicklungs- und Integrationsprojekt.

Unternehmen	Branche	SOA-Reichweite	Status
Deutsche Post Brief (DPB)	Logistik	Unternehmensweit, alle Kerngeschäftsprozesse	Produktiv
Credit Suisse (CS)	Finanzen	Unternehmensweit, alle Kerngeschäftsprozesse	Produktiv
T-Com (TC)	Telekommunikation	Prozessbereich Fulfillment (Vertriebs- und Produktionsprozesse)	In Entwicklung
Zuger Kantonalbank (ZGKB)	Finanzen	Kundenberatungsprozess für Kontenprodukte	Produktiv

Tabelle 2: Überblick über die Fallstudien

#### 3.1 Architekturtreiber und Leidensdruck

Für die Umsetzung einer SOA zeigen sich die folgenden Treiber:

- Für Deutsche Post Brief stellten schlecht verfügbare Kerninformationen (z.B. Kundeninformationen), mangelhaft integrierte Geschäftsprozesse, hohe Wartungs- und Betriebskosten der IS-Architektur sowie zunehmend umfang- und risikoreichere IT-Projekte Treiber für eine SOA dar.
- Die über Jahre hinweg gewachsene und kaum koordinierte Applikationslandschaft führte bei Credit Suisse zu steigender Komplexität, welche bspw. den Aufbau von Internet-Kundenkanälen oder die Ablösung des Mainframe-Buchhaltungssystems durch eine objektorientierte Lösung mit vertretbarem Aufwand verunmöglichte.
- Wiederholte organisatorische Anpassungen, steigender Bedarf nach unternehmensübergreifend integrierten Kooperationsprozessen mit Vertriebspartnern und ein umfangreicher Ausbau des Produkt- bzw. Leistungsangebots waren Startpunkt der SOA-Überlegungen bei T-Com. Sie zeigten sich v.a. in Auftrags-

---

<sup>3</sup> Die ausführlichen Fallstudien sind über die Publikationsdatenbank des IWI-HSG online zugänglich: [http://web.iwi.unisg.ch/org/iwi/iwi\\_pub.nsf/wwwPublRecentGer?](http://web.iwi.unisg.ch/org/iwi/iwi_pub.nsf/wwwPublRecentGer?)

abwicklung und Produktion durch Redundanzen, technische Heterogenität und enge Abhängigkeiten der unterstützenden Informationssysteme.

- Auslöser des Projekts bei ZGKB war eine unzureichende Unterstützung der Kundenberatungsprozesse durch eine neu eingeführte Standardsoftware. Benutzeroberfläche und Prozessführung waren für die Zielgruppe der Kundenberater ungeeignet.

Ausserdem deckte die neue Lösung nicht alle benötigten Funktionen und Daten ab.

Zusammenfassend lässt sich festhalten, dass wichtige Treiber für die Umsetzung einer SOA die unzulängliche Unterstützung bestehender Geschäftsprozesse, eine mangelhafte Flexibilität beim Umsetzen neuer Geschäftsanforderungen, hohe Kosten des IT-Betriebs und Probleme bei notwendigen technischen Modernisierungen in der IT-Architektur waren. Tabelle 3 konsolidiert die wichtigsten der in den Fallstudien genannten Treiber in den bestehenden IS-Architekturen.

Architekturtreiber	DPB	CS	TC	ZGKB
Paralleler Einsatz verschiedener Technologien und Plattformen in der Applikationsarchitektur				
Vielfalt an inkompatiblen bzw. redundanten Integrationstechnologien				
Schlecht kontrollierte Applikationsschnittstellen und enge logische Abhängigkeiten zwischen Systembestandteilen				
Fachfunktionen mehrfach in unterschiedlichen Anwendungen implementiert				
Redundante Datenhaltung				
Enge Verknüpfung von Präsentations-, Ablaufsteuerungs- und Fachlogik in monolithischen Systemen				
Mangelhafte Fähigkeit, fehlende Funktionalität extern einzubinden bzw. externen Partnern Zugriff auf interne Funktionen zu geben				

Tabelle 3: In den Fallstudien genannte Treiber für SOA

### 3.2 Reichweite und Einsatzziele

Die SOA-Projekte von Credit Suisse und Deutsche Post Brief umfassten von Beginn weg die unternehmensweite Applikationsarchitektur bzw. alle Kerngeschäftsprozesse, wobei die eigentliche Implementierung von Services nach aktuellen Geschäftsanforderungen priorisiert und projektweise realisiert wird. Die Architekturentwicklung von T-Com konzentriert sich in einem ersten Schritt auf den Fulfillment-Bereich mit Schwerpunkt Auftragserstellung und -abwicklung. Das Unternehmen will die SOA erst in einem nächsten Schritt auf weitere Prozesse und Anwendungsdomänen ausdehnen. Mit der Entwicklung eines integrierten Arbeitsplatzes für die Kundenberatung im Bereich Konten und Kontokarten war das BAP-Projekt von Zuger Kantonalbank auf einen sehr begrenzten Teil der Unternehmensprozesse und Applikationsarchitektur ausgelegt.

Die durchgeführten Projekte konzentrierten sich zuerst auf die Entwicklung unternehmensintern genutzter Services. Die Integration externer Geschäftspartner für durchgängige unternehmensübergreifende Prozesse stellte i.d.R. einen zweiten, auf den internen Services aufbauenden Schritt dar. Nur für T-Com besass der Zugang auf unternehmensinterne Funktionen über Services für externe Nutzer eine hohe Priorität. Das Unternehmen will früh Services wie bspw. eine produktübergreifende Verfügbarkeitsprüfung entwickeln, die sowohl interne Vertriebskanäle als auch Vertriebspartner nutzen können.

Im Vergleich der Fallstudien zeichnen sich drei Einsatzziele für SOA ab (s. Tab. 4):

- Als *standardisierte Integrationsinfrastruktur* erhöht eine SOA die technische Konnektivität heterogener Applikationen, reduziert die Vielfalt der Schnittstellentechnologien und verringert damit Integrations- und Wartungskosten. Angesichts einer wachsenden Anzahl heterogener Applikationsplattformen, Betriebssysteme, Integrationsinfrastrukturen und Entwicklungswerkzeuge verfolgen Deutsche Post Brief, Credit Suisse und T-Com das Ziel, die Integration bestehender Anwendungssysteme zu standardisieren und dazu eine plattformübergreifende Integrationsinfrastruktur aufzubauen. Dies ist vergleichbar mit typischen EAI-Zielen.
- Der Einsatz von SOA zur *Entkopplung von Applikationsdomänen* zielt auf eine Reduktion von Abhängigkeiten und Redundanzen zwischen Applikationen. Credit Suisse, Deutsche Post Brief und T-Com senken Projektrisiken sowie Kosten für die Neu- bzw. Weiterentwicklung von Anwendungen und den IT-Betrieb, indem sie ihre Applikationslandschaft in redundanzfreie, fachlich abgegrenzte Domänen unterteilen. Diese bieten wiederverwendbare Services und werden eigenständig verantwortet und weiterentwickelt. Die Entkopplung der Domänenarchitektur ist beim Fallbeispiel ZGKB nicht so stark ausgeprägt bzw. besitzt wegen der Beschränkung auf einen kleinen Teilbereich der Applikationsarchitektur nicht den gleichen Stellenwert wie bei den übrigen Unternehmen.
- Eine einfachere und schnellere Anpassung bzw. Neuentwicklung anwendungsübergreifender Prozesse strebt die *flexible Benutzer- bzw. Geschäftsprozessintegration* an. Alle untersuchten Unternehmen integrieren Services in Portalen, um Benutzerprozesse besser zu unterstützen. Eine Externalisierung von Ablauf- und Steuerungslogik in Form von Workflows bzw. Taskflows stellte zum Zeitpunkt der Fallstudienenerhebung aber nur bei der Zuger Kantonalbank ein zentrales Ziel der SOA dar. Die restlichen Unternehmen programmierten Prozesslogik zur Servicenutzung i.d.R. fest in den nutzenden Anwendungen ein. Zwar waren auch bei Credit Suisse und Deutsche Post Brief und T-Com zum Zeitpunkt der Fallstudienenerhebung Workflowsysteme im Einsatz bzw. geplant. Die Serviceorchestrierung auf Basis von Workflows war dort aber eine Massnahme, die erst in Zukunft stärker gewichtet werden soll.

Primäre Einsatzziele	DPB	CS	TC	ZGKB
SOA als standardisierte Integrationsinfrastruktur	■	■	■	■
SOA für die Entkopplung von Applikationsdomänen	■	■	■	■
SOA für eine flexible Benutzer- / Geschäftsprozessintegration	■	■	■	■
■ angestrebt    ■ teilweise angestrebt				

Tabelle 4: Primäre Einsatzziele der SOA

### 3.3 Einordnung der Fallbeispiele im SOA-Ebenenmodell

Die Beispiele zeigen, dass die Umsetzung von SOA erst in den Anfängen steckt. Keines der Unternehmen bearbeitet sämtliche Ebenen des Ebenenmodells aus Kap. 2.1 (s. Tab. 2). Deutsche Post Brief, Credit Suisse und T-Com legten das Hauptgewicht auf die Entwicklung einer Domänenarchitektur sowie die Umsetzung einer von Implementierungsdetails abstrahierenden Serviceschicht auf einer zentralen, standardisierten Integrationsinfrastruktur. Sie bewegen sich damit primär auf der Anwendungssystem- und der Serviceebene.

Bei der Zuger Kantonalbank bildete die Externalisierung von Prozesslogik aus bestehenden Applikationen und die Komposition von Services in Workflows und insbesondere Taskflows den Schwerpunkt. Zwar wurden auch hier bestehende Applikationsfunktionen als Services gekapselt und in einem zentralen Verzeichnis beschrieben, auf die Implementierung einer plattformunabhängigen Serviceschicht mit standardisierten Schnittstellentechnologien auf Basis einer zentralen Middlewareinfrastruktur (einem Enterprise Service Bus) verzichtete das Unternehmen aber vorerst. Da ein Grossteil der Services durch eine Applikation (SAP Banking) implementiert wird, spielte auch die Definition von Applikationsdomänen eine untergeordnete Rolle.

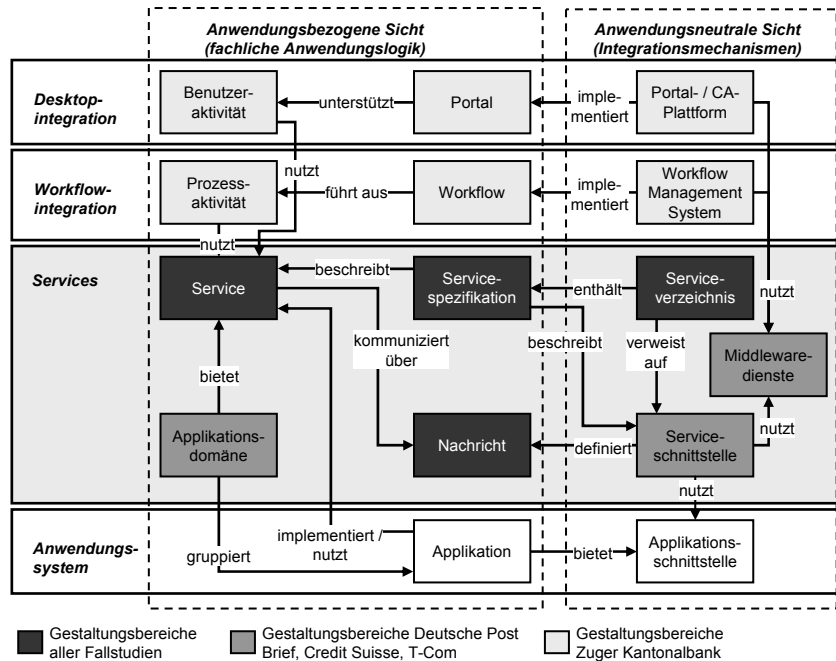


Abbildung 2: Hauptgestaltungsbereiche der Fallstudien

Tabelle 5 gibt einen Überblick über die Ausprägung der Architekturebenen in den Fallstudien zum Zeitpunkt ihrer Erhebung.

	DPB	CS	TC	ZGKB
Desktopintegration	U.a. Verwendung von Services in Mitarbeiter-, Kunden- oder Vertriebspartnerportalen, in Fallstudien nicht im Detail untersucht			Unterstützung der Kundenberatungsprozesse (Konten-anlage, Kontaktmanagement etc.) auf Basis SAP Enterprise Portal und Composite Application Framework
Workflow	Zum Zeitpunkt der Fallstudienenerhebung keine Workflow-basierte Servicenutzung			Unterstützung der Front-/Backoffice-Kommunikation auf Basis SAP Business Workflow
Ebene Services	<ul style="list-style-type: none"> <li>• 20-30 Services (ca. 100 geplant) in 13 Domänen</li> <li>• Zentrale, standardisierte Integrationsinfrastruktur auf Basis J2EE und Best-of-Breed-Produkte</li> <li>• Einheitliche Servicespezifikation in zentralem Serviceverzeichnis</li> </ul>	<ul style="list-style-type: none"> <li>• Ca. 300 Services (ca. 900 Serviceoperationen) in 20 Applikationsdomänen</li> <li>• Zentrale, standardisierte Integrationsinfrastruktur auf Basis CORBA und IBM WebSphere-Plattform</li> <li>• Einheitliche Servicespezifikation in zentralem Serviceverzeichnis</li> </ul>	<ul style="list-style-type: none"> <li>• Ca. 50-100 Services geplant, 5 Applikationsdomänen (nur Bereich Fulfillment)</li> <li>• Zentrale, standardisierte Integrationsinfrastruktur auf Basis IBM WebSphere-Plattform</li> <li>• Einheitliche Servicespezifikation in zentralem Serviceverzeichnis</li> </ul>	<ul style="list-style-type: none"> <li>• Ca. 20 Services, grossteils direkt auf SAP Banking Application Server implementiert</li> <li>• Keine zentrale Integrationsinfrastruktur, verschiedene Schnittstellentechnologien (ABAP Objects / SAP RFC, Java RMI)</li> <li>• Einheitliche Servicespezifikation, Dokumente auf Serververzeichnis abgelegt</li> </ul>

Tabelle 5: Ausprägung der Architekturebenen

### 3.4 Anwendung der SOA-Designprinzipien

Tabelle 6 gibt einen Überblick über die in den Fallstudien angewendeten SOA-Designprinzipien.

SOA-Designprinzipien	DPB	CS	TC	ZGKB
Abstraktion von der Serviceimplementierung				
Umfassende Servicespezifikation				
Stabile, gemanagte Servicekontrakte				
Technische Schnittstellenstandardisierung				
Fachliche Schnittstellenstandardisierung				
Verwendung offener und verbreiteter Industriestandards				
Hohe Servicekohäsion und schwache logische Kopplung				
Lose gekoppelte Kommunikation				
An Geschäftskonzepten orientierte Servicegranularität				
Generalisierung der Serviceleistung				
<span style="display: inline-block; width: 10px; height: 10px; background-color: black; border: 1px solid black;"></span> angewendet <span style="display: inline-block; width: 10px; height: 10px; background-color: gray; border: 1px solid black;"></span> teilweise angewendet				

Tabelle 6: Anwendung der SOA-Designprinzipien

Anhand der Designprinzipien zeigen sich die Unterschiede in der Anwendungsreichweite und in den Einsatzziele der SOA. Im Gegensatz zu den restlichen Fallbeispielen setzte Zuger Kantonalbank die SOA für einen kleinen und homogenen Teilbereich der Applikationsarchitektur um. Sowohl der Beraterarbeitsplatz als Servicenutzer als auch das Banking-System, das den überwiegenden Teil der genutzten Services implementierte, basierten auf der SAP-Produktfamilie und die Anzahl entwickelter Services war gering. Entsprechend untergeordnete Bedeutung besaßen hier die Abstraktion von der technischen und fachlichen Serviceimplementierung, das Management der Servicekontrakte sowie die technische und semantische Schnittstellenstandardisierung. Diese wird aber in Zukunft, mit einem Weiterausbau der SOA zunehmen. Die Unternehmen, die SOA für einen grösseren Bereich der Applikationsarchitektur umsetzen, wendeten diese Prinzipien stärker an. Sie erachteten auch eine fachlich-semantische Standardisierung der Serviceschnittstellen als wichtig.

Die Verwendung von Industriestandards ist in keinem der Fallbeispiele umfassend ausgeprägt. Grund dafür ist, dass hersteller- und technologieplattformübergreifende Standards momentan fehlen bzw. nicht reif genug sind. Im Bezug auf technische Standards geht Deutsche Post Brief am weitesten: Das Unternehmen achtet darauf, in der Integrationsinfrastruktur nur J2EE-Standards zu verwenden und herstellereigene Erweiterungen zu vermeiden, um sich möglichst wenig an einen bestimmten Softwareanbieter zu binden. Die anderen Unternehmen verfolgen zumindest in Teilbereichen der Integrationsinfrastruktur die Strategie, die Integrationsplattform von einem Hersteller zu beziehen und auch herstellereigene Erweiterungen zu nutzen. Auch die lose gekoppelte Kommunikation zwischen Services ist in keinem Fall durchgängig für alle Services umgesetzt. Zwar streben alle Unternehmen eine dynamische Serviceadressierung und eine statuslose Serviceinteraktion an, Service und Servicenutzer kommunizieren aber aus Performanzgründen häufig über synchrone Mechanismen.

### 3.5 Nutzenpotenziale und Herausforderungen

Tabelle 7 fasst die mit SOA verfolgten Nutzenpotenziale zusammen. Durch die Realisierung von Kosten-, Zeit- und Qualitätseffekten erhöhen die Unternehmen die Effizienz und Effektivität des IT-Einsatzes und die Anpassungsfähigkeit der IS-Architektur an neue Geschäftsanforderungen. Wegen der fortgeschrittenen Umsetzung der SOA konnten Deutsche Post Brief, Credit Suisse und Zuger Kantonalbank die Nutzenpotenziale bereits im täglichen IT-Betrieb und in einzelnen Entwicklungsprojekten beobachten und dazu quantitative oder qualitative Aussagen machen. T-Com hingegen befindet sich noch in einem frühen Umsetzungsstadium, und die angestrebten Nutzenpotenziale sind noch nicht verifizierbar.

Nutzenpotenziale		DPB	CS	TC	ZGKB
Kosten	Reduzierte Betriebskosten durch Harmonisierung von Plattformen und Technologien sowie Konzentration der technischen Fähigkeiten	realisiert	realisiert	gegenwärtig angestrebt	gegenwärtig angestrebt
	Senkung der Kosten für die Systemintegration durch technische und fachliche Schnittstellenstandardisierung	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Senkung von Entwicklungs- und Betriebskosten durch Wiederverwendung und Redundanzreduktion	realisiert	realisiert	gegenwärtig angestrebt	realisiert
Zeit	Verkürzung von Projektlaufzeiten und bessere Time-to-Market durch stärkere Wiederverwendung	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Schnellere Prozessanpassungen durch Trennung stabiler Geschäftslogik von dynamischer Prozesslogik	realisiert	realisiert	gegenwärtig angestrebt	realisiert
Qualität	Bessere Transparenz über Abhängigkeiten bzw. Schnittstellen in der Applikationsarchitektur	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Integrierte Prozesse und Investitionsschutz für bestehende Systeme durch plattformübergreifende Nutzung existierender Funktionen	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Besser beherrschbare bzw. besser isolierte IT-Projekte durch Domänen-Entkopplung	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Klar geregelte Verantwortlichkeiten für Fachfunktionen und Daten	realisiert	realisiert	gegenwärtig angestrebt	realisiert
	Vereinfachte Kommunikation zwischen IT- und Fachbereichen	realisiert	realisiert	gegenwärtig angestrebt	realisiert

■ realisiert    ■ gegenwärtig angestrebt

Tabelle 7: Realisierte bzw. angestrebte Nutzenpotenziale

Den Nutzenpotenzialen stehen auch Mehraufwände gegenüber. Diese bestehen u.a. in Anfangsinvestitionen für die Definition und Durchsetzung neuer Governance-Strukturen, Prozesse und Standards oder in der Implementierung der technischen Integrationsinfrastruktur. Auch im laufenden Betrieb und in Entwicklungsprojekten entstehen Zusatzaufwände, etwa bei der Entwicklung wieder verwendbarer Services oder in zentralen Projekt-Reviews. Neben den dadurch anfallenden Kosten bestehen auch verschiedene organisatorische und technische Herausforderungen.

### 3.6 Massnahmen zur Einführung einer SOA

Die untersuchten Unternehmen haben ihre SOA schrittweise in mehreren Teilprojekten geplant und eingeführt. Besonders aus den Fallbeispielen von Deutsche Post Brief, Credit Suisse und T-Com lassen sich grob zwei Gruppen von Aktivitäten identifizieren: Einerseits grundlegende projektübergreifende Architekturmanagement-Massnahmen, andererseits eine projektspezifische Identifikation, Entwicklung und Nutzung von Services in Applikationsentwicklungs- bzw. Applikationsintegrationsprojekten:

- In einem ersten Schritt verankerten die Unternehmen die Umsetzung der SOA in der IT-Organisation. Sie definierten dazu neue *Architekturrollen und Verantwortlichkeiten*, bzw. erweiterten die Aufgaben und Kompetenzen bestehender Rollen. Deutsche Post Brief bildete dazu den zentralen Bereich SOP-Group, T-Com einen ESB- / SOA-Ausschuss und ein SOA Solution Center. Auch Credit Suisse definierte Verantwortliche für die Projektunterstützung oder die Entwicklung und den Betrieb der Integrationsinfrastruktur im zentralen IT-Bereich.
- Aufgrund von Situations- und Potenzialanalysen bestimmten die Architekturverantwortlichen anschliessend die *Ziele und Einsatzbereiche* der SOA (s. Kap. 3.1 und 3.2).
- SOA-Ziele bildeten die Grundlage zur Formulierung von Architekturprinzipien. Diese machen zentrale Vorgaben für die Entwicklung und Nutzung von Services und der Integrationsinfrastruktur und umfassen etwa standardisierte Entwicklungs- und Reviewprozesse oder Prinzipien für das Servicedesign (s. Kap. 3.4).
- Mit der Entwicklung einer Domänenarchitektur strukturierten die Unternehmen die Applikationsarchitektur aus fachlicher Sicht und unterstützten damit Zuständigkeits- und Kopplungsentscheidungen: Domänengrenzen bestimmen, wo Applikationen lose gekoppelt über Services zu integrieren sind und wo alternative Kopplungsmechanismen zulässig sind.
- Bis auf ZGKB implementierten alle Unternehmen eine zentrale Integrationsinfrastruktur auf Service-Ebene. Diese Integrationsinfrastruktur vereinheitlicht Serviceschnittstellen, bietet zentrale Integrationsdienste (Verzeichnisdienst, Message-Bus etc.) und bildet damit die Grundlage für eine einfache plattformunabhängige Nutzung von Services. Während Unternehmen die früh mit der SOA-Umsetzung begonnen haben in der Vergangenheit ihre Integrationsinfrastruktur aus Produkten mehrerer Hersteller zusammenstellten, geht der Trend heute stärker in Richtung Verwendung einer umfassenden SOA-Plattform eines Herstellers (z.B. IBM WebSphere, SAP NetWeaver).
- Die laufende Überwachung und Steuerung der SOA über Kennzahlen war besonders im Fallbeispiel Credit Suisse ausgeprägt. Das Unternehmen definierte spezifisch für die SOA eine Architekturscorecard, mit welcher es die Architekturfortschritte periodisch misst. Auch Deutsche Post Brief erhob gewisse Metriken (z.B. Änderungshäufigkeit von Schnittstellen oder Wiederverwendungsgrad) zur Messung der Architekturqualität.

- Alle Fallbeispiele entwickelten und nutzten Services in fachlich getriebenen Applikationsentwicklungs- und Integrationsprojekten. Die Unternehmen sind aber bei der initialen Identifikation von zu entwickelnden Services unterschiedlich vorgegangen: Während Credit Suisse pro (domänenübergreifendem) Fachprojekt bestimmte, ob eine Schnittstelle als Service umgesetzt werden soll oder nicht, identifizierten Deutsche Post Brief oder T-Com Services vorausplanend pro Applikationsdomäne.

## 4 Zusammenfassung und Ausblick

Die Fallstudien zeigen, dass auch Unternehmen, die sich schon seit mehreren Jahren Erfahrung mit der SOA-Umsetzung sammeln, nicht sämtliche Aspekte des SOA-Konzepts parallel angehen. Vielmehr wählen sie einen fokussierten Ansatz und gewichten die SOA-Designprinzipien je nach Zielsetzung unterschiedlich. Inhalt weiterer Forschungsarbeiten wird deshalb die Identifikation typischer Muster für den SOA-Einsatz mit Architekturtreibern, Nutzenpotenzialen und zu berücksichtigenden Designprinzipien sein. Weiter sollte ein SOA-Qualitätsmodell entwickelt werden, welches Messgrößen für die Bewertung von Services und der SOA insgesamt definiert.

Die in den Fallstudien beobachteten Massnahmen zur Einführung von SOA korrespondieren gut mit den Aktivitäten, wie sie methodische Ansätze zur Gestaltung von Applikations- und Integrationsarchitekturen diskutieren [Sc04]. Diese behandeln die Aktivitäten allerdings auf einer generischen Ebene unter Berücksichtigung aller Arten von Integrationsansätzen. Weitere Forschungsarbeiten beschäftigen sich deshalb mit der Detaillierung der identifizierten Massnahmen und ihrer spezifischen Ausprägung für SOA.

## Literatur

- [Al03] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer, Berlin 2003.
- [Al04] Alt, R.: Überbetriebliches Prozessmanagement - Gestaltungsmodelle und Technologien zur Realisierung integrierter Prozessportale. Universität St. Gallen, St. Gallen, 2004.
- [Ba05] Baskerville, R. et al.: Extensible architectures: The strategic value of service-oriented architecture in banking. In: Bartmann, D.; Rajola, F.; Kallinikos, J.; Avison, D.; Winter, R.; Ein-Dor, P.; Becker, J.; Bodendorf, F.; Weinhardt, C. (Hrsg.): European Conference on Information Systems (ECIS). 2005
- [BJK02] Brown, A.; Johnston, S.; Kelly, K.: Using service-oriented architecture and component-based development to build web service applications. Rational Software Corporation, 2002.
- [Do05a] Dodd, J.: Practical Service Specification and Design Part 2b: Refining Service Requirements. In: CBDi Journal, (2005) Mai, S. 24-31.
- [Do05b] Dodd, J.: Practical Service Specification and Design Part 3: Specifying Services. In: CBDi Journal, (2005) Juni, S. 12-18.
- [Er05] Erl, T.: Service-Oriented Architecture. Prentice Hall, New York, 2005.

- [Fr04] Fritz, F.-J.: An Introduction to the Principles of Enterprise Services Architecture (ESA).  
[http://www.sapinsideronline.com/spijs/article.jsp?article\\_id=37906&volume\\_id=5147](http://www.sapinsideronline.com/spijs/article.jsp?article_id=37906&volume_id=5147), Abruf am 26.06.2004.
- [GH98] Gall, H.; Hajek, K.; Jazayeri, M.: Detection of logical coupling based on product release history. ICSM. 1998, S. 190-198
- [Gr98] Griffel, F.: Componentware: Konzepte und Techniken eines Softwareparadigmas. dpunkt, Heidelberg, 1998.
- [Ha05] Hayward, S.: Positions 2005: Service-oriented Architecture Adds Flexibility to Business Processes. Gartner inc., Stamford, 2005.
- [ID04] IDABC: European Interoperability Framework for Pan-European eGovernment Services. EU Enterprise and Industry Commission, 2004.
- [KWS05] Klesse, M.; Wortmann, F.; Schelp, J.: Erfolgsfaktoren der Applikationsintegration. In: Wirtschaftsinformatik, 47 (2005) 4, S. 259-267.
- [Mc03] McGovern, J. et al.: Service-Oriented Architecture. In: McGovern, J.; Tyagi, S.; Stevens, M.; Mathew, S. (Hrsg.): Java Web Services Architecture. Morgan Kaufmann, 2003, S. 35-63.
- [Mo05] Mougayar, W.: The SOA in IT Benchmark Report - What CIOs Should Know about How SOA Is Changing IT. 2005.
- [NL04] Newcomer, E.; Lomow, G.: Understanding SOA with Web Services. Addison-Wesley, Maryland, 2004.
- [Oa05] Oasis: Reference Model for Service Oriented Architectures, Working Draft 09. OASIS Open, 2005.
- [Pa03] Papazoglou, M. P.: Service-Oriented Computing: Concepts, Characteristics and Directions. 2003, S. 3-12
- [Pa72] Parnas, D. L.: On the criteria to be used in decomposing systems into modules. In: Communications of the ACM, 15 (1972) 12, S. 1053-1058.
- [PY02] Papazoglou, M. P.; Yang, J.: Design Methodology for Web Services and Business Processes. In: Buchmann, A.; et al. (Hrsg.): Tes 2002, LNCS 2444. Springer, Heidelberg, 2002, S. 54-64.
- [Qu04] Quack, K.: Service-orientierte Architektur, aber wie?  
<http://www.computerwoche.de/index.cfm?pageid=256&artid=61810&type=detail&kw=soa&rc=52>, Abruf am 31.03.2006.
- [Sc04] Schlamann, H.: Service Orientation: An Evolutionary Approach. In: Cutter IT Journal, 17 (2004) 5, S. 5-13.
- [Sc04] Schwinn, A.: Entwicklung einer Methode zur Gestaltung von Integrationsarchitekturen für Informationssysteme. Universität St. Gallen, Difo, Bamberg, 2006.
- [Sc06] Schemm, J. W. et al.: Framework zur Beschreibung Serviceorientierter Architekturen. 2006.
- [SG96] Shaw, M.; Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, New Jersey, 1996.
- [SGM02] Szyperski, C.; Gruntz, D.; Murer, S.: Component Software: Beyond Object-Oriented Programming. 2. Aufl., Addison Wesley Longman, London, 2002.
- [SÖ04] Senger, E.; Österle, H.: PROMET Business Engineering Case Studies (BECS) Version 2.0. 2004.

- [SW05] Schwinn, A.; Winter, R.: Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration. 7. Internationale Tagung Wirtschaftsinformatik 2005. Bamberg 2005, S. 587-606
- [Tu01] Turowski, K.: Spezifikation und Standardisierung von Fachkomponenten. In: Wirtschaftsinformatik, 43 (2001) 3, S. 269-281.
- [Tu02] Turowski, K. et al.: Vereinheitlichte Spezifikation von Fachkomponenten. 2002.
- [Vo04] Vogler, P.: Prozess- und Systemintegration: Evolutionäre Weiterentwicklung bestehender Informationssysteme mit Hilfe von Enterprise Application Integration. Universität St. Gallen, 2004.
- [W304] W3C: Web Services Architecture, W3C Working Group Note 11 February 2004. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, Abruf am 26.06.2004.
- [WF99] WFMC: Terminology & Glossary. [http://www.wfmc.org/standards/docs/TC-1011\\_term\\_glossary\\_v3.pdf](http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf), Abruf am 03.08.2004.
- [WKW05] Wilhelmi, C.; Klesse, M.; Wortmann, F.: Service-orientierte Architektur - Der richtige Weg? Institut für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen, 2005.
- [WV04] Wilkes, L.; Veryard, R.: Service-Oriented Architecture: Considerations for Agile Systems. In: Microsoft Architects Journal, (2004) 2, S. 11-23.
- [Yi94] Yin, R.: Case Study Research. Designs and Methods; SAGE Publications, London 1994.
- [Zy02] van Zyl, J.: A perspective on service based architecture. 2002, S. 249.