

2009

VORGEHENSMODELLE ZUR ENTWICKLUNG SERVICEORIENTIERTER SOFTWARESYSTEME

Oliver Thomas

Institut für Wirtschaftsinformatik im DFKI, Universität des Saarlandes

Katrina Leyking

Institut für Wirtschaftsinformatik im DFKI, Universität des Saarlandes

Michael Scheid

Institut für Wirtschaftsinformatik im DFKI, Universität des Saarlandes

Follow this and additional works at: <http://aisel.aisnet.org/wi2009>

Recommended Citation

Thomas, Oliver; Leyking, Katrina; and Scheid, Michael, "VORGEHENSMODELLE ZUR ENTWICKLUNG SERVICEORIENTIERTER SOFTWARESYSTEME" (2009). *Wirtschaftsinformatik Proceedings 2009*. 18.
<http://aisel.aisnet.org/wi2009/18>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2009 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

VORGEHENSMODELLE ZUR ENTWICKLUNG SERVICEORIENTIERTER SOFTWARESYSTEME

Oliver Thomas, Katrina Leyking, Michael Scheid¹

Kurzfassung

Bei der Entwicklung von Softwaresystemen auf Basis eines serviceorientierten Architekturparadigmas stellt sich die Frage, welches Vorgehensmodell zur Projektabwicklung herangezogen werden kann. In der Literatur werden unterschiedliche Vorgehensmodelle zur serviceorientierten Softwareentwicklung vorgeschlagen. Aus diesen werden 17 Modelle ausgewählt, charakterisiert sowie mit Hilfe eines allgemeinen Rahmens klassifiziert und verglichen. Dabei werden sowohl generelle Merkmale von Vorgehensmodellen als auch SOA-spezifische Vorgehensmerkmale herangezogen.

1. Einleitung

Führende Anbieter von IT-Lösungen sehen in der Serviceorientierten Architektur (SOA) das Paradigma der Zukunft. Mit diesem Management- und Systemarchitekturkonzept scheint es erstmalig möglich, die Modellierung betriebswirtschaftlich relevanter Abläufe mit deren physischer Ausführung über eine Softwareplattform zu verbinden [7; 20; 31; 33; 43]. Gartner prognostiziert, dass bis 2010 rund 65 Prozent der großen Unternehmen über 35 Prozent ihrer Anwendungsportfolios auf einer SOA-Architektur aufbauen werden [24]. Bei aller Euphorie ist zu beachten, dass in der Serviceorientierung nicht ein Softwareprodukt im Mittelpunkt steht, sondern eine langfristige, strategische Neuausrichtung der unternehmensweiten IT. Die entsprechenden Veränderungsprozesse sind durch eine hohe Komplexität gekennzeichnet. Zur Handhabarmachung dieser Komplexität werden in der Literatur verschiedene serviceorientierte Vorgehensmodelle vorgeschlagen. Trotz der Vielfalt an bestehenden SOA-Vorgehensmodellen existieren in der Literatur kaum Untersuchungen, die den Nachweis und die Dokumentation der Vorgehensmodelle im State-of-the-Art zum Gegenstand haben. Ebenso mangelt es darauf aufbauend an Arbeiten zur Behandlung der Frage, welche SOA-Vorgehensmodelle in welchen Anwendungssituationen zu nutzen sind. In Anlehnung an vergleichende Übersichten über Vorgehensmodelle des Software Engineering [2], komponentenorientierte Vorgehensmodelle [10] und objektorientierte Vorgehensmodelle [30] sowie Vorarbeiten zur Katalogisierung von Vorgehensmodellen [11] soll daher in diesem Beitrag ein vergleichender Überblick über ausgewählte Vorgehensmodelle zur serviceorientierten Entwicklung von Softwaresystemen gegeben werden. Dabei sollen sowohl generelle Merkmale von Vorgehensmodellen herangezogen werden als auch Aspekte, die im Hinblick auf die serviceorientierte Systementwicklung eine besondere Rolle spielen. Die einzelnen SOA-Vorgehensmodelle sollen nicht im Detail vorgestellt werden, zu ihrer ausführlichen Herleitung und Beschreibung wird auf die Originalquellen verwiesen. Der Vergleich der serviceorientierten Vorgehensmodelle ist systematisch und wird auf Basis eines einheitlichen Vergleichsrahmens durchgeführt.

¹ Institut für Wirtschaftsinformatik im DFKI, Universität des Saarlandes, Stuhlsatzenhausweg 3, 66123 Saarbrücken.

2. Neue Anforderungen an Vorgehensmodelle durch SOA

Vorgehensmodelle des Software Engineering beruhen auf der Idee des Software-Lifecycle und werden als ein Rahmenkonzept verstanden, dem jede Softwareentwicklung folgen soll [für einen Überblick vgl. 2, Kap. II 3.3]. Sie dienen der zeitlichen, logischen Strukturierung der durchzuführenden Aktivitäten innerhalb der Entwicklungs- und Einsatzphasen von Informationssystemen, der Zuordnung von Mitarbeitern zu diesen Aktivitäten sowie der Beschreibung der zu erzielenden Ergebnisse. Durch SOA soll es möglich werden, Anwendungssysteme flexibel aus eigenständigen Teilsystemen (Services) zusammenstellen zu können [20; 23]. Anwendungssysteme sollen nicht als Gesamtsysteme im Unternehmen vorgehalten werden, vielmehr sollen benötigte Dienste bedarfsgerecht über das Internet bezogen und unternehmensindividuell konfiguriert werden [12, S. 78f.; 21, S. 33]. Die Ziele einer SOA-Entwicklung unterscheiden sich somit von denjenigen „traditioneller“ Softwareprojekte. Die daraus resultierenden Herausforderungen sind bei der Auswahl von Vorgehensmodellen für eine SOA-Entwicklung zu berücksichtigen. In dieser Untersuchung müssen die Vorgehensmodelle folgenden im Rahmen einer Literaturrecherche identifizierten Kriterien – im Sinne SOA-spezifischer Gestaltungsprinzipien – gerecht werden:

- *Modularisierung*: SOA basieren auf einer Dekomposition bestehender Anwendungssysteme in funktional gekapselte Services [9, S. 290; 20, S. 242; 27, S. 42]. Dabei sind eine Kohäsion von Funktionalitäten und eine Autonomie unterstützender Services anzustreben. Einhergehend ist das Prinzip der losen Kopplung zu beachten, das auf eine Unabhängigkeit bei gleichzeitiger schnittstellenbasierter Interoperabilität der Services untereinander abzielt [6, S. 9; 45, S. 157].
- *Abstraktion*: Eine serviceorientierte Softwaregestaltung basiert auf einer Trennung von Schnittstelle und Implementierung, wodurch dem Softwarenutzer ein abstrakter Systemzugang gewährt wird und Codierungsdetails verborgen bleiben. Durch das Information-Hiding-Prinzip [35, S. 1056] wird eine plattformneutrale Servicebeschreibung realisiert, die die Servicenutzung unabhängig von der programmiertechnischen Ausgestaltung des in Anspruch nehmenden Systems macht. Die Auswahl der technischen Softwarelösung kann zur Laufzeit erfolgen.
- *Ausgewogene Granularität*: Bei SOA ist ergänzend das richtige Maß an fein- und grobgranularen Services zu finden [9, S. 299; 20, S. 242; 37, S. 127]. Die Granularität gibt Auskunft über Art und Umfang der Funktionalität, die durch ein Modul bereitgestellt wird. Sind Services grobgranular gestaltet, kann deren Wiederverwendung beeinträchtigt werden. Feingranulare Services können zu Ineffizienzen führen und einen fachaufgabenorientierten Einsatz gefährden.
- *Prozessorientierung*: Die Komposition von Services zu inner- und überbetrieblichen Geschäftsprozessen stellt einen zentralen Bestandteil einer SOA-Entwicklung dar. Sie steht in der Tradition der Konfiguration von Workflow-Management-Systemen und macht sich die Prinzipien des „Programming-in-the-large“ [5] zunutze. Die Prozessgestaltung kann dabei als Bindeglied zwischen dem Organisationssystem und dem Anwendungssystem betrachtet werden. Sie stellt ein durchgängiges „Business/IT Alignment“ [14] in Aussicht, welches Voraussetzung für die Zielsetzung flexibler und wiederverwendbarer Services ist.

Traditionelle Vorgehensmodelle des Software Engineering schenken diesen SOA-spezifischen Kriterien nicht immer Beachtung, was die Notwendigkeit serviceorientierter Vorgehensmodelle begründet. Allerdings sind in der Literatur auch serviceorientierte Vorgehensmodelle zu finden, die gleichsam wesentliche Charakteristiken einer SOA unberücksichtigt lassen. So gibt bspw. die Architectural und Organization Roadmap von Krafzig et al. [20] keine Empfehlungen für die Erzielung ausgewogener Granularität und abstrakter Serviceschnittstellen. Demgegenüber sieht das Konzept von Overhage, Turowski [32] keine explizite Modularisierung vor. Tsai [41] spricht in seinem Ansatz zwar von einer Servicekomposition zur Laufzeit, berücksichtigt aber keine prozessorientierte Serviceorchestrierung. Die im Rahmen einer Literaturrecherche identifizierten serviceorientierten Vorgehensmodelle, die den genannten Kriterien genügen, sind in Tabelle 1 zusammengestellt.

Tabelle 1: Vorgehensmodelle zur Entwicklung serviceorientierter Softwaresysteme

Quelle	Vorgehensmodell	Charakterisierung
[1]	Service-oriented Modeling and Architecture Method	Arsanjani (IBM) beschreibt den schematischen Aufbau einer SOA. Darauf aufbauend erfolgt die Erläuterung des Vorgehensmodells, bestehend aus den Hauptphasen „Identification“, „Specification“ und „Realisation“, die in die Subphasen „Service Identification“, „Service Classification or Categorization“, „Subsystem Analysis“, „Component Specification“, „Service Allocation“ und „Service Realisation“ segmentiert werden. Das Modell baut auf denselben konzeptionellen Grundlagen auf wie das Vorgehensmodell von Mitra.
[3]	Creating Serviceoriented Architectures	Barry & Associates, Inc. stellen ein fünfstufiges Vorgehensmodell für die Konzeption einer web-service-basierten SOA vor, das aus den Stufen „Experiment with Web Services“, „Adapt Existing Systems to Use Web Services“, „Remove Intersystem Dependencies“, „Establish an Internal SOA“ sowie „Establish an Internal SOA to include External Services“ besteht. Der Schwerpunkt liegt auf den technischen Aspekten, die mit SOA und der Realisation durch Web Services verbunden sind.
[4]	Service-Oriented Analysis and Design Activities	Bieberstein (IBM) stellt vier Phasen vor, die im Rahmen eines serviceorientierten Analyse- und Designprozesses durchzuführen sind: „Service Identification“, „Service Categorisation“, „Service Specification“ und „Service Realisation“. Neben Arsanjani und Mitra, ist dies der dritte in dieser Arbeit vorgestellte Autor, der für IBM tätig ist.
[13]	Enterprise-SOA-Roadmap-Methodik	Hack und Lindemann stellen die von SAP entwickelte Roadmap-Methodik vor. Sie besteht aus den Phasen „Strategische Rahmenbedingungen“, „Transparenz zur Geschäfts- und IT-Landschaft“, „Potenzialanalyse“, „Design“ und „Roadmap-Definition“, die weiter segmentiert sind. Ein Alleinstellungsmerkmal liegt in der Schwerpunktlegerung auf organisatorische und strategische Fragestellungen im Vorfeld der SOA-Entwicklung.
[15]	Creating of a Service Architecture	Der Ansatz von Jones und Morris, der als OASIS-Draft publiziert wurde, hat die Beschreibung eines „Service Discovery Approaches“ zum Ziel und orientiert sich an einer fortschreitenden Top-Down-Analyse ausgehend von einer Betrachtung des Geschäftsumfelds. Die vollständige Definition und die Implementierung der Services liegen außerhalb des Beitragsfokus.
[16]	Procedure Model for Service Identification	Dieses an der Universität Münster entwickelte Vorgehensmodell befasst sich vornehmlich mit der Serviceidentifikation. Es besteht aus den Hauptphasen „Preparation“, „Service Design“ und „Service Categorisation“. Im Rahmen einer Geschäftsprozessanalyse erfolgt im Gegensatz zu anderen Vorgehensmodellen eine Differenzierung zwischen manuellen, semiautomatischen und automatischen Funktionen.
[19]	Process Models for business-oriented Service Identification and Clustering	Kohlmann (Universität Leipzig) beschreibt ein Vorgehensmodell für Serviceidentifikation und Serviceclustering, welches in die Phasen „Preparation“, „Analysis“, „Verification“ und „Detailing“ untergliedert ist. In die Konstruktion dieses Vorgehensmodells sind die Erkenntnisse einer zuvor durchgeführten Analyse bereits existierender Vorgehensmodelle eingeflossen.
[22]	Procedure Model for a SOA-Based Integration of Enterprise Systems	Dieser Ansatz von sd&m beschreibt ein Vorgehensmodell für eine SOA-basierte Integration von Anwendungssystemen. Es besteht aus den Hauptphasen „Task based decomposition of systems“, „Preparation of the integration and mapping of redundant functions“, „Detection and assignment of services to code fragment“ und „Orchestration of web services“. Gleichzeitig wird das Vorgehensmodell anhand der Integration eines ERP-Systems und eines Content-Management-Systems beispielhaft angewendet.
[25]	Service Oriented System Engineering	Masak, der als Consultant und Autor tätig ist, nimmt eine modelltechnische Dreiteilung in Geschäftsmodell, Servicemodell und Implementierungsmodell vor. Der Konstruktionsprozess wird in die Phasen „Discover“, „Selection“, „Rating“, „Assembling“, „Execution“, „Monitoring“, „Accounting“ und „Disintegration“ unterteilt.
[26]	Vorgehensmodell zur Umsetzung von SOA-Systemen	Mathas (Siemens) führt ein Vorgehensmodell für SOA ein, das aus den Phasen „Betrachtungen zur Systemauslegung“, „Prozessanalyse“, „Funktionsanalyse“, „Systemdesign“, „Systementwicklung“, „Systemeinführung“, „Systembetrieb“ und „Wartung“ sowie „Anpassungen und Updates“ besteht, wobei er für jede Phase eine rollenspezifische Zuordnung vornimmt.
[28]	Business-driven SOA Development	Mitra (IBM) beschreibt ein Vorgehensmodell, das die Phasen „Business Component/Analysis“, „Service Identification“, „Service Spezifikation“, „Component Identification“, „Component Specification“, „Service Realisation Decision“ und „SOA Implementation“ umfasst. Neben einer anwenderorientierten business-getriebenen Ableitung der Services, wird eine gleichzeitige service-provider-orientierte Analyse bestehender Systeme vorgeschlagen.
[29]	Seven Steps to a Service-Oriented Evolution	Nadhan (EDS Consulting Group) definiert sieben Phasen, die im Rahmen eines SOA-Entwicklungsprozesses zu durchlaufen sind: „Identify“, „Locate“, „Group“, „Package“, „Orchestrate“, „Route“ und „Govern“. Dabei legt er einen Schwerpunkt auf die technischen Aspekte einer SOA-Entwicklung.
[34]	Service-oriented Design and Development Methodology	Das an der an der Universität Tilburg (NED) entwickelte Modell für das Design und die Entwicklung von Services besteht aus den Schritten „Planning“, „Analysis“, „Service Design“, „Service Construction“, „Service Test“, „Service Execution“ sowie „Service Management and Monitoring“. Die Vorgehensbeschreibung beinhaltet im Gegensatz zu anderen untersuchten Modellen, eine detaillierte Erläuterung zur BPEL-basierten Umsetzung von Services.
[36]	SOA-Entwicklungsprozess	Pingel (brick_AT_work GmbH) beschreibt einen aus den Schritten „Spezifikation“, „Analyse“, „Design“ sowie „Realisierung“ und „Tests“ bestehenden Ansatz für die SOA-Entwicklung. Die Schritte sind dabei in weitere Subphasen untergliedert. Über den gesamten Entwicklungsprozess wird UML zur Modellerstellung eingesetzt.
[17; 39]	Vorgehensmodell zur Entwicklung von Geschäfts-servicen	Das von der IDS Scheer AG entwickelte Vorgehensmodell umfasst insgesamt zehn Phasen. Es beschreibt einen modellgetriebenen Ansatz, der verstärkt auf die betriebswirtschaftlich relevanten Aspekte einer SOA fokussiert ist. Zudem wird für die Prozessschritte eine Zuständigkeitsverteilung für die Rollen „Business Analyst“, „Process Engineer“ und „Software Engineer“ vorgenommen.
[40]	Modellbasierte Gestaltung serviceorientierter Architekturen	Der an der Universität des Saarlandes in Zusammenarbeit mit der SAP entwickelte Ansatz ist modellgetrieben. Die SOA-Entwicklung erfolgt dabei anhand eines dreistufigen Ordnungsrahmens, bestehend aus den Phasen „Prozessgestaltung“, „Prozesskonfiguration“ und „Prozessausführung“ und verfolgt speziell das Ziel, die Lücke zwischen der konzeptionellen Modellierung und der serviceorientierten IT-Unterstützung zu schließen.
[44]	Service Identification and Packaging in Service Oriented Reengineering	Das am Software Technology Research Laboratory in Leicester (ENG) entwickelte Vorgehensmodell dient der Überführung und Integration bestehender Softwarekomponenten in serviceorientierte Informationssysteme. Die Hauptphase der „Service Identificaton“ ist dabei untergliedert in „Service identification in problem domain“, „Service identification in a legacy system“ sowie „Matching legacy functionalities to business functions in logical services“. Die zweite Hauptphase „Service Packaging“ ist die in die Subphasen „Legacy code refinement“, „New functional development“, „Service interface and description development“, „Transport mechanism integration“, „Connector development“ und „Service complexity reduction“ segmentiert.

3. Ein Vergleichsrahmen für serviceorientierte Vorgehensmodelle

Methodische Vorüberlegungen

Um die wesentlichen Aspekte zur Charakterisierung von SOA-Vorgehensmodellen herauszuarbeiten, wird eine Klassifikation entworfen. Allgemein liegt eine Klassifikation vor, wenn ein Untersuchungsgegenstand nach einem bestimmten Merkmal und dessen Ausprägungen gegliedert wird [8, S. 10ff.; 18, S. 142]. Zu einer solchen Klassifikation können durchaus mehrere Einteilungen nach jeweils anderen Kriterien gehören, ohne dass sich dadurch am Charakter einer klassifikatorischen Ordnung grundsätzlich etwas ändert. Die Kriterien stehen isoliert neben- oder hintereinander, wobei eine Verknüpfung der verschiedenen Einteilungskriterien nicht vorgenommen wird [18, S. 142]. Das in dieser Untersuchung verwendete Klassifikationssystem für SOA-Vorgehensmodelle besteht aus 31 Merkmalen (vgl. Tabelle 2). Um die Übersichtlichkeit des Klassifikationssystems zu wahren, sind die Merkmale zu fünf Merkmalsklassen zusammengefasst worden. Diese Klassen berücksichtigen einerseits, dass SOA-Vorgehensmodelle nicht nur entwickelt, sondern auch angewendet werden. Andererseits wird zwischen dem Vorgehen zur Vorgehensmodellkonstruktion (Konstruktionsprozess) und dessen Resultat (Konstruktionsergebnis) getrennt. Die Klassifikation orientiert sich ergänzend an den wesentlichen in der Literatur zu Vorgehensmodellen aufgeführten generellen Strukturmerkmalen und -ausprägungen [11], die um SOA-spezifische Aspekte erweitert sind.

Konstruktionsprozess und -ergebnisbezogene Merkmale

Die in dieser Klasse enthaltenen Merkmale charakterisieren den mit der Entwicklung eines Vorgehensmodells verbundenen Konstruktionsprozess und dessen Konstruktionsergebnis. *Herkunft*, *Erkenntnisweg* und *Repräsentationsform* sind im Sinne ihres in der Wissenschaft geläufigen Gebrauchs zu verstehen. Nach dem *Interaktionsgrad*, also dem Grad des wechselseitigen aufeinander Einwirkens der beteiligten Akteure und/oder Systeme, sind Vorgehensmodelle dahingehend zu unterscheiden, ob sie individuell, kooperativ (arbeitsteilig) oder kollaborativ (gemeinsam bearbeitend) erstellt werden. Diese Unterscheidung adressiert das in der Literatur diskutierte „Glaubwürdigkeitsproblem“ vieler individuell entwickelter Referenzmodelle. Die *Hierarchisierungsstufe* gibt an, ob sich die Phasen eines Vorgehensmodells in weitere Subphasen unterteilen. Das Merkmal *Individualität* legt fest, ob der Ersteller eines Vorgehensmodells dieses als Referenzmodell für eine Klasse von Anwendungsfällen betrachtet oder ob es im Rahmen eines fallspezifischen, individuellen SOA-Entwicklungsprojekts entstanden ist.

Merkmale mit Bezug zu SOA-Entwicklungszielen

Diese Merkmalsklasse dient dazu, die Ziele und Zwecke, für welche ein SOA-Vorgehensmodell entwickelt wurde, näher zu charakterisieren. Das Merkmal *Perspektive* beschreibt die potenzielle Zielgruppe eines Vorgehensmodells. Dabei erfolgt eine Dreiteilung in die Ausprägungen Serviceanbieter, -broker und -anwender. In Abhängigkeit zu entwickelnden Informationssystemen können Vorgehensmodelle danach unterschieden werden, ob sie eher die Anwendungs- oder die Organisationssystemkomponente im Sinne eines *Informationssubsystems* betrachten. Darüber hinaus sind für Vorgehensmodelle zwei unterschiedliche Schwerpunkte in Bezug auf den *Verwendungszweck* zu differenzieren. Vorgehensmodelle, die auf Business Engineering ausgelegt sind, fokussieren die flexible Anpassung von Geschäftsprozessen an sich ständig verändernde wirtschaftliche Anforderungen. Im Gegensatz dazu liegt das Hauptaugenmerk von software-engineering-orientierten Vorgehensmodellen auf der Gestaltung des Zusammenspiels zwischen den Geschäftsprozessen und der sie unterstützenden Anwendungssysteme. Da für die Entwicklung von SOA mehrere technische Optionen bestehen, sind Vorgehensmodelle dahingehend zu unterscheiden, ob sie in Bezug auf ihre *technologische Ausrichtung* unabhängig oder gebunden (z.B. an den Web-Service-Technology-Stack) sind. Ein weiteres Merkmal trifft eine Aussage darüber, ob ein Vorgehensmodell einen konkreten *Produktbezug* aufweist. Insbesondere aus der Praxis stammende Vorgehensmodelle sind

häufig auf konkrete Softwareprodukte (z. B. Modellierungswerkzeuge oder Entwicklungsumgebungen) ausgerichtet. Die Merkmale *Organisations-* und *Aufgabenbezug* beschreiben die verschiedenen Projektgrenzen, für die ein Vorgehensmodell auslegt sein kann. Es ist zu unterscheiden, ob ein Vorgehensmodell nur organisationsinterne SOA berücksichtigt, oder ob es zusätzlich die Entwicklung von unternehmensübergreifenden SOA unterstützt. Analog kann ein Vorgehensmodell dazu dienen, lediglich eine einzelne Aufgabe innerhalb eines Unternehmens als SOA abzubilden, oder es kann zur Entwicklung einer SOA dienen, die einen kompletten Unternehmensteilbereich abdeckt.

SOA-unspezifische Vorgehensmerkmale

Diese Merkmalsklasse umfasst generelle Eigenschaften, mit denen sich Vorgehensmodelle charakterisieren lassen. Die *Anwendungsdomäne* eines Vorgehensmodells kann entweder spezifisch oder neutral ausgeprägt sein. Domänenneutrale Vorgehensmodelle sind auf beliebige Entwicklungsszenarien anwendbar, während domänenspezifische Modelle nur auf eine Klasse von Anwendungsfällen (z. B. Branche) passen. Das Merkmal *Vorgehenssteuerung* beschreibt die verschiedenen Arten der Prozesssteuerung für Vorgehensmodelle. Ein Vorgehensmodell ist aktivitätsorientiert, wenn es eine festgelegte Abfolge von Aktivitäten beschreibt. Es ist ergebnisorientiert, wenn sein Schwerpunkt auf der phasenübergreifenden Transformation der zu erzielenden Ergebnisse liegt. Ein entscheidungsorientiertes Vorgehensmodell definiert Bedingungen, die in Abhängigkeit ihres Eintretens die Art und die Reihenfolge der durchzuführenden Aktivitäten bestimmen. Das Merkmal *Problemlösung* orientiert sich in seinen Ausprägungen iterativ, inkrementell und rekursiv an den korrespondierenden Vorgehensmodellbeschreibungen aus dem Bereich des Software Engineering [11]. Die *Phasenanzordnung* eines Vorgehensmodells kann entweder streng sukzessiv aufgebaut sein, d. h. mit einer neuen Phase kann erst dann begonnen werden, wenn die vorangehende Phase abgeschlossen ist, aber auch mehrere parallel auszuführende Phasen aufweisen. Die Merkmale *Sprachempfehlung*, *Methodenempfehlung* und *Werkzeugunterstützung* treffen Aussagen über Instrumente, denen sich das Vorgehensmodell bedient. Das Spektrum von Sprachempfehlungen reicht vom UML-Aktivitätsdiagramm über die EPK bis hin zu BPMN oder BPEL. Als Empfehlung für eine homogene Gesamtmethode kommen beispielsweise ARIS oder UML in Frage. Damit einhergehend kann zusätzlich eine Werkzeugempfehlung für einzelne Phasen oder den Gesamtprozess ausgesprochen werden. Die Ergebnisse der einzelnen Phasen werden i. d. R. in einer *Ergebnisdokumentation* festgehalten, deren Gestaltung textuell oder grafisch (modellorientiert) erfolgen kann.

SOA-spezifische Vorgehensmerkmale

Neben den zuvor beschriebenen, generellen Charakteristika, die auf Vorgehensmodelle anwendbar sind, existiert darüber hinaus eine Reihe SOA-spezifischer Merkmale. So ist im Rahmen einer SOA-Entwicklung zu Beginn i. d. R. eine Projektinitialisierung durchzuführen, um einen Anforderungskatalog zu definieren. Als *Gegenstand der Initialisierung* kommen dabei entweder eine Geschäftsprozessanalyse, die Untersuchung eines bereits existierenden Anwendungssystems oder eine abstrakte Betrachtung der Unternehmensstrategie, aus der sich konkrete Anforderungen ableiten lassen, in Frage. Eine SOA-Einführung kann analog zu den im Software Engineering bestehenden Optionen durch eine Aneinanderreihung einzelner Teilprojekte oder durch einen „Big-Bang“ erfolgen. Im ersteren Fall ist die *Migrationsstrategie* evolutionär, im letzteren Fall revolutionär. Die *Entwicklungsstrategie* eines SOA-Vorgehensmodells lässt sich analog zum Software Engineering in die Ausprägungen top-down, middle-out und bottom-up untergliedern. Die Funktionen, die innerhalb eines Geschäftsprozesses auftreten, sind nach ihrer potenziellen IT-Unterstützung in automatisch, semiautomatisiert und manuell zu differenzieren. Das Merkmal *Automatisierungsgrad* gibt an, welche der entsprechenden Servicearten ein Vorgehensmodell berücksichtigt. Zu einem Zeitpunkt, zu dem bereits die Anforderungen und die erforderliche Funktionalität für ein Projekt festgelegt wurden, ist zu entscheiden, welche Services mit welchem Funktionsumfang diese Anforderungen abdecken sollen. Grundsätzlich bestehen zwei Möglichkeiten zur Durchführung dieser *Service-*

identifikation. Eine fachliche Serviceidentifikation leitet die erforderlichen Services aus den zuvor erkannten Funktionen der Geschäftsprozesse ab, eine technische Serviceidentifikation erfolgt durch eine Überführung der Funktionalität von Komponenten bereits existierender Anwendungssysteme in Services. Darüber hinaus können SOA-Vorgehensmodelle als Merkmal die konkrete Darstellung eines Konstruktionsprozesses für die *technische Serviceentwicklung* enthalten, die eine detaillierte Beschreibung für die Entwicklung einzelner Services umfasst. Neben der Identifikation von Services ist ein Abgleich zwischen den identifizierten und den real verfügbaren Services vorzunehmen. Das Ziel eines *SOA-Alignment-Prozesses* liegt darin, evtl. bestehende Abweichungen zwischen den identifizierten und den real verfügbaren Services zu beseitigen. Dies kann durch eine direkte Zuordnung von identifizierten Services zu realen Services erfolgen. Für business-getriebene Vorgehensmodelle besteht darüber hinaus die Möglichkeit einer indirekten Zuordnung. Dabei werden die Geschäftsprozessmodelle in einem Zwischenschritt zu technisch-fachlichen Anwendungsmodellen transformiert, anschließend erfolgt eine Zuordnung zu den real existierenden Services. Ein Beispiel für einen indirekten Transformationsprozess ist die Überführung von EPK- in BPMN-Modelle [40]. Zu einem nachgelagerten Zeitpunkt einer SOA-Einführung ist eine Auswahlentscheidung zwischen den infrage kommenden Services zu treffen. SOA-Vorgehensmodelle können für die *Serviceselektion* eine Entscheidungsunterstützung auf zwei Arten zur Verfügung stellen: anhand von Quality-of-Service-Merkmalen [38] oder alternativ anhand wirtschaftlicher Gütekriterien [42]. Um aus einzelnen Services eine Applikation erstellen zu können, müssen diese zusammengefügt werden. Diesen Vorgang bezeichnet man auch als *Servicekomposition*. Grundsätzlich bestehen für die Komposition zwei grundlegende Strategien. Zum einen können die Services prozessorientiert gesteuert werden. Für den Fall, dass unternehmensübergreifende Applikationen implementiert werden sollen, ist eine Koordination zwischen den Anwendungssystemen der beteiligten Partner zu gewährleisten (Service-Choreographie). Alternativ zur Prozessorientierung lassen sich Services auch zu funktionsorientierten Komponenten aggregieren. In diesem Fall werden nur die Schnittstellen dieser Komponenten über eine Orchestrierung gesteuert, während innerhalb der Komponenten z.B. objektorientierte Techniken verwendet werden. In Bezug auf ihre Anwendung sind SOA dadurch gekennzeichnet, dass in den meisten Fällen bereits zur Buildtime einer SOA definiert wird, welche Services zur Runtime aufzurufen sind. SOA erlauben es allerdings auch, die Entscheidung darüber, welche Services aufzurufen sind, in die Runtime zu verschieben. Der Entscheidungsprozess dieses „dynamic binding“ kann entweder ereignisgesteuert oder manuell ablaufen. Vorgehensmodelle lassen sich insofern unterscheiden, ob sie diese verschiedenen Arten einer *Servicesteuerung* berücksichtigen.

4. Anwendung des Vergleichsrahmens

In Tabelle 2 sind die ausgewählten Vorgehensmodelle in den hergeleiteten Merkmalen und Merkmalsausprägungen vergleichend gegenübergestellt. Es ist erkennbar, dass die serviceorientierten Vorgehensmodelle eine große Vielfalt von Anwendungsszenarien bedienen. Diese inhaltliche Heterogenität erschwert ihren Vergleich. Das Spektrum reicht von der einfachen Serviceidentifikation auf Basis bestehender Softwarelandschaften [29], der Erarbeitung fachlich orientierter Servicearchitekturen [15], der serviceorientierten Integration zweier Softwaresysteme [22] über die unternehmensweite Einführung von SOA [13] bis hin zu Reorganisationsprojekten auf Basis von SOA sowie der Entwicklung serviceorientierter Softwaresysteme [44]. Die Breite des Spektrums verdeutlicht, dass über die Entwicklungsprozesse für serviceorientierte Architekturen bislang kein Konsens herrscht. Daher wird in dieser Untersuchung bewusst auf eine terminologische Eingrenzung der Vorgehensmodelle zugunsten einer kriterienbasierten Auswahl verzichtet.

Während frühe Vorgehensmodelle zur serviceorientierten Systementwicklung an der technischen Realisierung ausgerichtet sind [3; 29], betonen neuere Ansätze die Relevanz betriebswirtschaftlich-fachlicher Aspekte für den Entwicklungsprozess [34; 44]. Beispielhaft sei die Diskussion um eine

Entscheidungsunterstützung bei der Auswahl von Web Services aufgegriffen, die sich häufig an der Dienstgüte (Quality of Service), einem eher technisch ausgerichteten Kriterium, orientiert [38]. Fragen der wirtschaftlichen Nutzung von SOA werden erst im neueren Schrifttum berücksichtigt [42].

Tabelle 2: Klassifikation serviceorientierter Vorgehensmodelle

Klasse	Merkmal	Merkmalsausprägung	[1]	[3]	[4]	[13]	[15]	[16]	[19]	[22]	[25]	[26]	[28]	[29]	[34]	[36]	[39]	[40]	[44]	
Konstruktionsprozess	Herkunft	Wissenschaft	–	–	–	–	(✓)	(✓)	✓	✓	✓	–	–	–	✓	–	(✓)	✓	✓	
		Praxis	✓	✓	✓	✓	✓	–	–	✓	–	✓	✓	✓	–	✓	✓	✓	–	
	Erkenntnisweg	Deduktiv	✓	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	–	✓	–	✓	✓	✓
		Induktiv	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓	–	✓	✓	✓	✓	–
	Interaktionsgrad	Individuell	–	✓	–	–	–	–	–	–	✓	–	–	–	–	–	–	–	–	–
		Kooperativ	✓	–	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–
	Kollaborativ	–	–	–	–	–	–	–	–	✓	–	–	–	–	–	–	–	–	✓	
Konstruktionsergebnis	Repräsentationsform	Textuelles Modell	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Grafisches Modell	✓	–	–	✓	–	–	✓	✓	✓	–	✓	✓	–	–	✓	✓	✓	✓
	Hierarchisierung	Hierarchisiert	✓	✓	–	✓	–	–	✓	✓	–	–	–	–	–	–	–	–	–	✓
		Nicht Hierarchisiert	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Individualität	Spezifisches Modell	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
	Referenzmodell	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
SOA-Entwicklungsziel	Perspektive	Serviceanbieter	(✓)	–	(✓)	–	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	–	(✓)
		Servicebroker	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Serviceanwender	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Informationssystem	Organisationssystem	–	–	–	✓	✓	✓	–	–	–	–	–	–	–	–	✓	✓	✓	–
		Anwendungssystem	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Verwendungszweck	Business Engineering	✓	–	–	✓	✓	✓	✓	–	–	–	–	(✓)	–	–	–	–	✓	✓
Software Engineering		✓	✓	✓	(✓)	–	–	–	–	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	✓	
Technologische Ausrichtung	Unabhängig	–	–	✓	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	
	Gebunden	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
Produktbezug	Produktunabhängig	✓	✓	✓	–	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	
	Produktabhängig	–	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
Organisationsbezug	Organisationsübergreifend	✓	–	✓	✓	✓	✓	–	–	–	–	–	(✓)	–	–	–	–	–	–	
	Organisationsintern	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	
Aufgabenbezug	Aufgabenbereich	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	
	Einzelaufgabe	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	
Anwendungsdomäne	Domänenspezifisch	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
	Domänenneutral	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	
	Aktivitätsorientiert	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Ergebnisorientiert	–	–	–	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	
Vorgehenssteuerung	Ergebnisorientiert	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
	Entscheidungsorientiert	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
SOA-unspezifische Vorgehensmerkmale	Problemlösung	Rekursiv	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
		Iterativ	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Inkrementell	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–
	Phasenordnung	Parallel	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Sukzessiv	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Methodenempfehlung	Heterogen	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–
		Homogen	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Sprachempfehlung	Keine	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Singular	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Pluralistisch	–	–	–	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–
	Werkzeugunterstützung	Keine	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Phasenbezogen	–	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Durchgängig	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	
Ergebnisdokumentation	Textuell	✓	–	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	
	Grafisch	✓	–	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	
SOA-spezifische Vorgehensmerkmale	Gegenstand der Initialisierung	Geschäftsprozess	✓	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Anwendungssystem	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Unternehmensstrategie	–	–	–	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–
	SOA-Migrationsstrategie	Evolutionär	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		Revolutionär	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	SOA-Entwicklungsstrategie	Top-Down	✓	–	✓	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–
Middle-Out		✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	

Klasse	Merkmal	Merkmalsausprägung	[1]	[3]	[4]	[13]	[15]	[16]	[19]	[22]	[25]	[26]	[28]	[29]	[34]	[36]	[39]	[40]	[44]		
Auto-matisierungs-grad		Bottom-Up	✓	✓	✓	–	–	✓	✓	✓	✓	✓	✓	✓	✓	–	–	–	✓		
		Manuell	–	–	✓	–	0	✓	0	–	–	–	–	–	–	–	✓	–	–	–	
		Semiautomatisiert	–	–	✓	–	0	✓	0	–	–	–	–	–	–	–	✓	–	–	–	
		Automatisiert	✓	✓	✓	✓	0	✓	0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Service-identifikation		Fachlich	✓	–	✓	✓	✓	✓	–	–	✓	✓	✓	–	✓	✓	✓	✓	✓		
		Technisch	✓	✓	✓	–	–	–	–	✓	✓	✓	✓	✓	✓	–	–	–	–	✓	
Technische Service-entwicklung		Berücksichtigt	✓	✓	✓	–	0	–	0	✓	–	–	✓	–	✓	–	–	–	–	✓	
		Nichtberücksichtigt	–	–	–	✓	0	✓	0	–	✓	✓	–	✓	–	✓	✓	✓	✓	–	
SOA-Alignment-Prozess		Keiner	–	✓	–	–	0	–	–	✓	–	✓	–	–	–	–	–	–	–	–	
		Direkt	✓	–	✓	✓	0	(✓)	–	–	–	–	–	✓	✓	✓	✓	✓	✓	✓	
		Indirekt	–	–	–	–	0	–	–	✓	–	✓	–	–	–	–	–	–	–	✓	–
Service-selektion		Keine	✓	✓	–	0	0	0	0	✓	–	✓	✓	✓	–	✓	✓	✓	✓	✓	
		Quality of Service	–	–	✓	0	0	0	0	–	✓	–	–	–	–	✓	–	–	–	–	–
		Wirtschaftlichkeit	–	–	–	0	0	0	0	–	–	–	–	–	–	–	–	–	–	–	–
		Prozessorientiert	✓	✓	0	✓	0	✓	0	✓	0	✓	✓	✓	✓	✓	✓	✓	✓	✓	0
Service-komposition		Orchestrierung	0	0	0	0	0	0	0	✓	0	✓	✓	✓	✓	✓	–	0	✓	0	
		Choreographie	0	0	0	0	0	0	0	–	0	✓	–	✓	–	✓	–	0	0	0	
		Komponentenorientiert	–	–	0	✓	0	–	0	–	0	–	–	–	–	–	–	–	–	–	0
Service-steuerung		Manuell	0	–	0	–	0	0	0	–	–	0	–	–	–	–	–	–	–	0	
		Vordefiniert	0	✓	0	✓	0	0	0	✓	✓	0	✓	✓	✓	✓	✓	✓	✓	✓	0
		Eventorientiert	0	–	0	–	0	0	0	–	–	0	–	–	–	–	–	–	–	–	0

Legende: ✓ Merkmal erfüllt, – Merkmal nicht erfüllt, (✓) Merkmal teilweise erfüllt, 0 Merkmal nicht zu beurteilen

Eine Mehrheit von dreizehn Vorgehensmodellen sieht eine direkte [z.B. 36] oder indirekte [z.B. 40] Überführung fachlicher Serviceanforderungen des Organisationssystems in die serviceorientierte Umsetzung des Anwendungssystems vor. Jedoch bleibt die umgekehrte Rückführung von technischen Anforderungen in die Fachbeschreibungen von nahezu allen Vorgehensmodellen unberücksichtigt. In der Folge fehlt es insgesamt an wechselseitigen Synchronisationstechniken, die für eine Integration von Organisations- und Anwendungssystemen erforderlich sind.

Gleichwohl bei SOA i. d. R. drei Stakeholder zu unterscheiden sind, bleiben in vielen Vorgehensmodellen Perspektiven auf die serviceorientierte Systementwicklung unberücksichtigt. Die vorliegenden Empfehlungen nehmen fast ausnahmslos die Rolle der Service-Konsumenten, d.h. der anwendenden Organisation, ein und lassen die Sichtweise der Service-Anbieter und -Broker außen vor. Dies steht im Widerspruch dazu, dass Softwareprojekte i. d. R. nicht von den Konsumenten der zu entwickelnden Anwendungssysteme durchgeführt werden. Für die erfolgreiche Abwicklung eines Softwareprojekts sind zudem Erfahrung und Fachwissen vorauszusetzen. Nicht jeder Service-Konsument, der eine SOA-Einführung beabsichtigt, verfügt zwangsläufig über ein ausreichendes Maß an erforderlichem Fachwissen. In diesem Zusammenhang ist in Zukunft eine Veränderung des Fokus von reinen Anwendermodellen hin zu anbieterorientierten Vorgehensmodellen zu fordern.

Nur wenige Vorgehensmodelle berücksichtigen organisationsübergreifende Geschäftsszenarien für die Anwendung einer SOA. Dabei verspricht die Serviceorientierung gerade für kollaborative Geschäftsprozesse größere Flexibilität und Integration. Gleichwohl gehen nur drei Vorgehensmodelle [26; 28; 34] explizit auf die Möglichkeit ein, unternehmensübergreifende Prozesse durch die Choreographie von Services abzubilden. Die übrigen Modelle beschränken sich auf eine unternehmensinterne Serviceorchestrierung bzw. nehmen keine Unterscheidung vor.

Auffällig ist weiterhin, dass über die Hälfte der untersuchten Vorgehensmodelle die Frage nach einer Migrationsstrategie vernachlässigt und keine Aussage darüber trifft, wie die zu entwickelnde SOA in die bestehende Anwendungslandschaft eingebunden werden kann. In der Praxis ist diese Problematik der Integration jedoch immer aufzulösen. Für die weitere Gestaltung von Vorgehensmodellen ist die Einbindung einer Migrationsstrategie als obligatorischer Bestandteil zu fordern.

Darüber hinaus zeigt sich, dass eine Reihe von Vorgehensmodellen eine technische Serviceidentifikation durch die Analyse der bestehenden Anwendungssysteme vorsieht. Für die Implementierung ist dabei eine Transformation von Fragmenten der Altsysteme in Services vorgesehen. Eine solche Vorgehensweise ist nur für Serviceprovider und nicht für Serviceanwender praktikabel, da die anwendenden Organisationen in der Regel fremdbezogene, betriebliche Standardsoftware einsetzen, für die ihnen i. d. R. die erforderlichen Softwarelizenzen fehlen.

Aus der Perspektive des Prozessmanagements ist zu beachten, dass nur die wenigsten Geschäftsprozesse durchgängig mit softwaregestützten Services zu automatisieren sind. Manuelle Tätigkeiten und eine Mensch-Maschine-Interaktion werden vor dem Hintergrund der zunehmenden Bedeutung von Dienstleistungs- und Wissensarbeit auch zukünftig eine bedeutende Rolle bei der serviceorientierten Systementwicklung spielen. Lediglich drei der untersuchten Vorgehensmodelle beziehen entsprechende Überlegungen mit ein [4; 16; 36].

5. Konklusion und Ausblick

Insgesamt kann das Profil der serviceorientierten Vorgehensmodelle kaum zufrieden stellen. Trotz des breiten Spektrums der Modelle offenbart die Analyse einen Rückstand des State-of-the-Art gegenüber den Potenzialen, die gemeinhin mit einer serviceorientierten Systemgestaltung verbunden werden. Dementsprechend liefert die in diesem Beitrag durchgeführte Untersuchung erstmalig eine umfassende Bewertung von SOA-Vorgehensmodellen und zeigt Verbesserungspotenziale für das Vorgehen in serviceorientierten Implementierungsprojekten auf. Der entwickelte Vergleichsrahmen bietet eine Vielzahl an SOA-spezifischen und -unspezifischen Merkmalen und ermöglicht die kritische Würdigung neu konstruierter serviceorientierter Vorgehensmodelle. Die insb. im Rahmen der Merkmalsklassifikation gewonnenen Erkenntnisse können in einem konkreten SOA-Einführungsprojekt als Entscheidungsunterstützung für die Auswahl eines Vorgehens herangezogen werden. Darüber hinaus lassen sich funktionale Anforderungen an eine IT-Unterstützung für die Einführung von SOA ableiten. Aufgrund der flexiblen Struktur kann der Vergleichsrahmen zukünftig jederzeit erweitert und um relevante Merkmale für einen spezifischen Untersuchungskontext ergänzt werden. Eine mögliche Erweiterung des Vergleichsrahmens wäre die Evaluation der Vorgehensmodelle in ihrem praktischen Einsatz im Rahmen von Anwenderbewertungen.

6. Literatur

- [1] Arsanjani, A.: *Service-oriented modeling and architecture*. Markham, Canada : IBM, 2004
- [2] Balzert, H.: *Lehrbuch der Software-Technik : Software-Entwicklung*. 2. Aufl. Heidelberg : Spektrum, 2001
- [3] Barry, D. K.: *Web services and service-oriented architectures*. Amsterdam : Morgan Kaufmann, 2003
- [4] Bieberstein, N.: *Service-oriented architecture (SOA) compass*. Upper Saddle River, NJ : IBM Press, 2006
- [5] DeRemer, F.; Kron, H.: Programming-in-the large versus programming-in-the-small. In: *Proceedings of the international conference on Reliable software*. Los Angeles, CA : ACM, 1975, S. 114–121
- [6] Dostal, W.; Jeckle, M.; Melzer, I.; Zengler, B.: *Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis*. Heidelberg : Elsevier, 2005
- [7] Dustdar, S.; Krämer, B. J.: Introduction to special issue on service oriented computing (SOC). In: *ACM Transactions on the Web* 2 (2008), Nr. 2, S. 1–2
- [8] Engeli, G.: *Der Begriff der Klassifikation*. Hamburg : Buske, 1971
- [9] Erl, T.: *Service-oriented architecture*. Upper Saddle River : Prentice Hall PTR, 2006
- [10] Fettke, P.; Intorsureanu, I.; Loos, P.: Komponentenorientierte Vorgehensmodelle im Vergleich. In: Turowski, K. (Hrsg.): *4. Workshop Komponentenorientierte betriebliche Anwendungssysteme*. Augsburg, 2002, S. 19–43
- [11] Filß, C. et al.: Rahmen zur Auswahl von Vorgehensmodellen. In: Petrasch, R. et al. (Hrsg.): *12. Workshop der Fachgruppe WI-VM der GI*. Aachen : Shaker, 2005, S. 183–227
- [12] Fremantle, P.; Weerawarana, S.; Khalaf, R.: Enterprise services. In: *Communications of the ACM* 45 (2002), Nr. 10, S. 77–82
- [13] Hack, S.; Lindemann, M. A.: *Enterprise SOA einführen*. Bonn : Galileo Press, 2007
- [14] Henderson, J. C.; Venkatraman, N.: Strategic Alignment: Leveraging Information Technology for Transforming Organisations. In: *IBM Systems Journal* 32 (1993), Nr. 1, S. 272–284
- [15] Jones, S.; Morris, M.: *A Methodology for Service Architectures*. London : Capgemini, 2005

- [16] Klose, K.; Knackstedt, R.; Beverungen, D.: Identification of Services – A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning. In: Österle, H. et al. (Hrsg.): *Proceedings of the 15th European Conference on Information Systems, St. Gallen, Juni 2007*, S. 1802–1814
- [17] Klückmann, J.: *10 Steps to Business-Driven SOA*. Saarbrücken : IDS Scheer AG, 2007
- [18] Knoblich, H.: Die typologische Methode in der Betriebswirtschaftslehre. In: *Wirtschaftswissenschaftliches Studium* 1 (1972), Nr. 4, S. 141–147
- [19] Kohlmann, F.: Service Identification and Design – A Hybrid Approach in Decomposed Financial Value Chains. In: Reichert, M.; Strecker, S.; Turowski, K. (Hrsg.): *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures, St. Goar, Germany, October 2007*, S. 205–218
- [20] Krafzig, D.; Banke, K.; Slama, D.: *Enterprise SOA*. 5. print. Upper Saddle River : Prentice Hall PTR, 2006
- [21] Kreger, H.: Fulfilling the Web Services Promise. In: *Communications of the ACM* 46 (2003), Nr. 6, S. 29–34
- [22] Lämmer, A.; Eggert, S.; Gronau, N.: A Procedure Model for a SOA-based Integration of Enterprise systems. In: *International Journal of Enterprise Information Systems* 4 (2008), Nr. 2, S. 1–12
- [23] Leymann, F.: Web services: Distributed applications without limits. In: Weikum, G.; Schöning, H.; Rahm, E. (Hrsg.): *BTW 2003, Datenbanksysteme für Business, Technologie und Web, Tagungsband der 10. BTW-Konferenz, 26.–28. Februar 2003, Leipzig*. Leipzig : Springer, 2003, S. 2–23
- [24] Malinverno, P.: *Service-Oriented Architecture Craves Governance*. Stamford, CT : Gartner Research, 2006
- [25] Masak, D.: *SOA?: Serviceorientierung in Business und Software*. Berlin : Springer, 2007
- [26] Mathas, C.: *SOA intern: Praxiswissen zu serviceorientierten IT-Systemen*. München : Hanser, 2008
- [27] McGovern, J.; Tyagi, S.; Stevens, M.; Mathew, S.: *Java web services architecture*. San Francisco : Kaufmann, 2003
- [28] Mitra, T.: *Business-driven development*. IBM : Markham, Canada, 2005
- [29] Nadhan, E. G.: Seven Steps to a Service-Oriented Evolution. In: *Business Integration Journal* (January) (2004), S. 41–44
- [30] Noack, J.; Schienmann, B.: Objektorientierte Vorgehensmodelle im Vergleich. In: *Informatik-Spektrum* 22 (1999), Nr. 3, S. 166–180
- [31] Nüttgens, M.; Iskender, D.: Geschäftsmodelle für dienstebasierte Informationssysteme – Ein strategischer Ansatz zur Vermarktung von Webservices. In: *Wirtschaftsinformatik* 50 (2008), Nr. 1, S. 31–38
- [32] Overhage, S.; Turowski, K.: Serviceorientierte Architekturen – Konzept und methodische Herausforderung. In: Nissen, V.; Petsch, M.; Schorcht, H. (Hrsg.): *Service-orientierte Architekturen – Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen*. Wiesbaden : DUV, 2007, S. 3–17
- [33] Papazoglou, M. P.; Traverso, P.; Dustdar, S.; Leymann, F.: Service-Oriented Computing: A Research Roadmap. In: *International Journal of Cooperative Information Systems* 17 (2008), Nr. 2, S. 223–255
- [34] Papazoglou, M. P.; van den Heuvel, W.-J.: Service-oriented design and development methodology. In: *International Journal Web Engineering and Technology* 2 (2006), Nr. 4, S. 412–442
- [35] Parnas, D. L.: On the criteria to be used in decomposing systems into modules. In: *Communications of the ACM* 15 (1972), Nr. 12, S. 1053–1058
- [36] Pingel, D.: Der SOA-Entwicklungsprozess. In: Starke, G.; Tilkov, S. (Hrsg.): *SOA-Expertenwissen: Methoden, Konzepte und Praxis serviceorientierter Architekturen*. Heidelberg : dpunkt, 2007, S. 187–200
- [37] Sametinger, J.: *Software engineering with reusable components*. Berlin : Springer, 1997
- [38] Spahn, M.; Berbner, R.; Heckmann, O.; Steinmetz, R.: Ein heuristisches Optimierungsverfahren zur dienstgütebasierten Komposition von Web-Service-Workflows. In: Steinmetz, R. (Hrsg.): *Technical Report, Multimedia Communications Lab*, Nr. KOM-TR-2006-02, TU Darmstadt, 2006
- [39] Stein, S.; Ivanov, K.: Vorgehensmodell zur Entwicklung von Geschäfts servicen. In: Fähnrich, K.-P.; Thränert, M. (Hrsg.): *Integration Engineering – Motivation, Begriffe, Methoden und Anwendungsfälle, Leipziger Beiträge zur Informatik VI*. Leipzig : Eigenverlag Leipziger Informatik-Verband, 2007
- [40] Thomas, O.; Leyking, K.; Dreifus, F.: Prozessmodellierung im Kontext serviceorientierter Architekturen. In: *HMD – Praxis der Wirtschaftsinformatik* 43 (2007), Nr. 253, S. 37–46
- [41] Tsai, W. T.: Service-Oriented System Engineering: A New Paradigm. In: *Service-Oriented System Engineering, 2005 (SOSE 2005) : IEEE International Workshop, 20–21 Oct. 2005 ; Proceedings*. IEEE, 2005, S. 3–6
- [42] vom Brocke, J.: *Serviceorientierte Architekturen – SOA*. München : Vahlen, 2008
- [43] Woods, D.; Mattern, T.: *Enterprise SOA: designing IT for business innovation*. Beijing : O'Reilly, 2006
- [44] Zhang, Z.; Liu, R.; Yang, H.: Service Identification and Packaging in Service Oriented Reengineering. In: Chu, W. C. et al. (Hrsg.): *Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'2005), Taipei, Taiwan, Republic of China, July 14–16, 2005*. Taipei, 2008, S. 620–625
- [45] Zhu, H.: *Software design methodology*. Amsterdam : Elsevier, 2005