

WI – Dissertationen

Die Rubrik Dissertationen ist unter der Adresse <http://www.wirtschaftsinformatik.de> online zu erreichen. Auf diesen Seiten ist eine Übersicht mit Doktorandenseminaren bei Tagungen und Konferenzen verfügbar. Zudem wird eine Datenbank mit abgeschlossenen sowie laufenden Dissertationen auf dem Gebiet der Wirtschaftsinformatik aufgebaut. Alle Doktoranden werden gebeten, sich unter der oben genannten Adresse zu registrieren. Zwei der Einträge in die Datenbank abgeschlossener Dissertationen sind nachstehend aufgeführt.

Outsourcing Relationships: Designing Contracts for Successful Outsourcing – An Analysis of the German Banking Industry

Gellings, Cornelia

Promotion am 2007-07-18, erschienen bei Books on Demand, Norderstedt, 212 Seiten, 49,95 €

Based on a mathematical principal-agent model this cumulative PhD thesis shows that an efficient contract can set the stage for achieving a true sharing of risk and return between the parties involved in an outsourcing deal. Furthermore, this re-

search analyses how outsourcing risks can concretely be mitigated by agreeing on respective contract clauses. The results highlight that contract clauses regarding problem resolution and SLA quality are of utmost importance to mitigate risk and to ensure the long term success of an outsourcing project.

Gutachter: Prof. Dr. Wolfgang König, Prof. Dr. Roland Holten, Universität Frankfurt

Stichworte: Outsourcing-Vertrag, Risikomanagement bei Outsourcing, Service Level Agreements (SLA), Principal-Agent-Theorie, Fallstudie, Strukturgleichungsmodell, PLS

E-Mail: gellings@wiwi.uni-frankfurt.de

Strategische Liefernetze – Evaluierung, Auswahl, kritische Knoten

Müssigman, Nikolaus

Promotion am 2006-11-22, erschienen bei DUV Gabler, Wiesbaden, 273 Seiten, 49,90 €, ISBN 9783835007413

Outsourcing über die gesamte Lieferantensstruktur nimmt in vielen Branchen zu.

Dadurch entstehen große, komplexe Liefernetze, die von der Beschaffungsfunktion im Unternehmen beherrscht werden müssen. Der Paradigmenwechsel vom direkten Lieferanten zum Liefernetz sowie der Wandel des Einkaufs von einer administrativen Bestellabwicklungsfunktion zu einer strategischen Beschaffungsfunktion erfordern neue Prozesse und Konzepte zur Unterstützung der effektiven und effizienten Versorgung im Unternehmen.

Nikolaus Müssigmann erarbeitet Methoden und Verfahren zur Identifikation und Evaluierung strategischer Liefernetze. Besonderen Wert legt er dabei auf die Berücksichtigung kritischer Knoten. Er greift Disziplinen übergreifende Vorarbeiten auf und setzt sich unter wirtschaftsinformatischen Aspekten mit interorganisationalen Anwendungssystemen auseinander. Praxisrelevante Hinweise zur Planung und Steuerung strategischer Liefernetze runden die Arbeit ab.

Gutachter: Prof. Dr. Klaus Turowski, Prof. Dr. Axel Tuma, Prof. Dr. Bernhard Fleischmann, Universität Augsburg

Stichworte: Strategische Liefernetze, Unternehmensnetzwerke

E-Mail: nikolaus.muessigmann@t-online.de

WI – Meinung/Dialog

SOA – Ein neues Paradigma der Gestaltung verteilter Informationssysteme?

Einleitung

In der Informatik sind in den letzten Jahren mit den Begriffen der Serviceorientierten Architektur (SOA) bzw. der Webservices (WS) neue Konzepte für die Gestaltung und Nutzung modularer, insbesondere verteilter Informationssysteme entstanden. Im engeren Sinn beschreibt SOA ein Konzept verteilter Informationstechnologie, in dem autonome, plattformunabhängige Softwaremodule (Dienste) zu kollaborativen Softwareapplikationen vernetzt werden können. Standardisierte WS-Protokolle ermöglichen eine anwendungs- und realisierungsunabhängige Beschreibung, Dienstfindung und Orchestrierung dieser Dienste.

Der Begriff der Services oder Dienste findet sich aber auch in den Wirtschaftswissenschaften an unterschiedlichen Stellen wieder. Auf einer makroökonomischen Ebene stellt ein hoher Anteil angebotener und nachgefragter Dienstleistungen ein wesentliches Charakteristikum entwickelter Wirtschaftssysteme dar. Eine mikroökonomische Betrachtung würde die flexiblere Unterstützung von Prozessen in den Mittelpunkt stellen, die erst durch eine Dienstorientierung möglich wird. Im Bereich des Dienstleistungsmanagements werden Services als wissensintensive, häufig kundenindividuelle und auftragsbezogene Interaktion zwischen Anbieter und Konsument betrachtet. Schließlich verbindet sich mit dem Schlagwort „Service Science“ eine Diskussion über eine entsprechende Ausrichtung in der Betriebswirtschaftslehre, die in der Rubrik „WI – Schlagwort“ dieses Hefts aufgenommen wird.

Es stellt sich die Frage, ob es sich bei dieser Begriffsähnlichkeit um mehr handelt als bloß sprachliche Nähe. Elektronische Dienstleistungen nutzen zur Erstellung und Ausführung die gleichen Informations- und Kommunikationstechnologien wie elektronische Produkte (off-the-shelf-software), aber der Unterschied scheint bisher im Wesentlichen nur in der Abrechnungsmethode „on demand“ greifbar zu werden. Reicht das zur inhaltlichen Abgrenzung bereits aus? Existieren darüber hinaus auch inhaltliche Gemeinsamkeiten, die Gegenstand von Forschung und Anwendung in

der Wirtschaftsinformatik sein können? Handelt es sich beim Grundkonzept von SOA tatsächlich um die Übertragung des ökonomischen Dienstleistungsparadigmas auf die Informatik? Kann umgekehrt die Betriebswirtschaftslehre aus den Innovationen und Realisierungen von Services in der Informatik lernen?

In der derzeitigen Forschung zu SOA und WS stehen noch technische Fragen verteilter Informationssysteme, wie Dienstspezifizierung, Dienstzusammenstellung, Architekturen und Realisierung von Workflows im Vordergrund. Aus der Sicht der Wirtschaftsinformatik stellen sich zusätzlich Fragen der Prozess- und Netzwerkgestaltung, Entwurfs- und Implementierungskonzepte sowie neue Formen und Größenordnungen einer intra- und interorganisatorischen Integration, Koordination und Verfügbarkeit von Ressourcen. Diese werden zukünftig komplementiert durch betriebswirtschaftliche Probleme ähnlich denen im Dienstleistungsmanagement: Messung der Dienstqualität, Vorteile verbesserter Skalierbarkeit, dezentrales Management, Fehler-toleranz, neue Kalkulations-, Verrechnungs- und Organisationskonzepte.

Insgesamt stellt sich die Frage, ob und wie weitgehend die Vision SOA tatsächlich Realität werden wird. Handelt es sich dabei um einen weiteren von Analysten und Marketingstrategen heraufbeschworenen Hype oder wird es tatsächlich fundamentale Änderungen geben in der Art, wie in Zukunft Informationssysteme entwickelt, genutzt und gesteuert werden? Beispielfhaft seien hier einige zu diskutierende Fragestellungen genannt:

- Besitzen serviceorientierte Architekturen und verwandte Konzepte, wie z. B. Service-oriented Computing oder Webservices, tatsächlich revolutionäre Eigenschaften?
- In vielen Realisierungen des SOA-Konzeptes stehen derzeit autonome, plattformunabhängige Softwaremodule zur Gestaltung und Nutzung von Workflows in verteilten Systemen im Vordergrund. Ist damit das Innovationspotenzial von SOA in realistischer Weise beschrieben?
- Sind Ähnlichkeiten erkennbar zwischen dem technischen „Service“-Begriff der SOA und dem betriebswirtschaftlichen Paradigma, welches unserer Leistungsgesellschaft zugrunde liegt, oder handelt es sich nur um eine sprachliche Koinzidenz?
- Welche weiteren Schritte sind im internationalen Wettbewerb sinnvoll, um diese

Technologie voran zu bringen? Nutzen solche Schritte im Zeitalter des Offshoring von Softwareentwicklungen nicht nur den Werkbänken der Softwareindustrie in Osteuropa und Asien?

Über diese Fragen diskutieren:

- Prof. Dr. Elmar J. Sinz (Universität Bamberg)
- Dr. Jürgen Laartz (McKinsey & Co.)
- Karl-Heinz Streibich (BITKOM e.V. und Software AG)
- Prof. Dr. Michael N. Huhns (University of South Carolina)
- Dr. Orestis Terzidis, Dr. York Sure und Christian Brelage (SAP Research, CEC Karlsruhe)

Wenn auch Sie sich an dieser Debatte beteiligen wollen, dann senden Sie Ihre Stellungnahme (max. 4.000 Zeichen) bitte an hans-ulrich.buhl@wiwi.uni-augsburg.de.

Prof. Dr. Torsten Eymann
Universität Bayreuth
Prof. Dr. Robert Winter
Universität St. Gallen

SOA und die bewährten methodischen Grundlagen der Entwicklung betrieblicher IT-Systeme

Elmar J. Sinz

Serviceorientierung und serviceorientierte Architekturen (SOA) gehören zu den derzeit meist diskutierten Themen im Bereich der Gestaltung betrieblicher IT-Systeme. Flexibilisierung, Wiederverwendbarkeit, Komponentenorientierung und Integration, Bewältigung von IT-Altlasten sowie Unterstützung des IT/Business-Alignment sind Ziele und Erwartungen, die mit SOA verknüpft werden.

Steht SOA wirklich für einen fundamentalen Paradigmenwandel bei der Gestaltung der betrieblichen IT-Systeme oder handelt es sich größtenteils um „alten Wein in neuen Schläuchen“? Siedersleben (2007, S. 111) nennt drei kennzeichnende Merkmale von SOA: Komponentenorientierung, lose Kopplung und Workflow. Komponentenorientierung ist ein Ziel, das im Software Engineering als Ingenieurdisziplin seit mehr als vier Jahrzehnten verfolgt wird und in dem auch lange vor SOA sichtbare Erfolge erzielt wurden (z. B. CORBA der Object Management Group OMG). Lose Kopplung ist ein konstituierendes Merkmal verteilter Systeme (z. B. Enslow

1978) und auch die Workflow Management Coalition (WfMC) wurde bereits im Jahr 1993 gegründet.

Obwohl eine Reihe von Arbeiten zur ganzheitlichen Gestaltung von SOA vorliegen (z. B. Erl 2007; Krafzig et al. 2005; Pulier und Taylor 2006; Starke und Tilkov 2007; Woods und Mattern 2006), stellt SOA immer noch eher eine unscharf abgegrenzte Leitidee oder Vision dar als ein konkretes Paradigma zur Gestaltung von IT-Systemen. Das technologische Umfeld von SOA wird bestimmt durch eine Vielzahl von Standards für die Spezifikation, die Ausführung und das Zusammenwirken von Webservices. Es besteht die Gefahr, dass sich der Blick zu sehr auf diese Standards verengt und diese zum Bezugspunkt der Methodenentwicklung im Bereich SOA werden. Eine umfassende SOA-Gestaltungsmethodik erfordert aber den Blick für das Ganze und die Einbeziehung des gewachsenen Wissens der Systementwicklung. Im Folgenden soll versucht werden, einige der in der aktuellen SOA-Euphorie eher vernachlässigten Aspekte aus dem Bereich der bewährten methodischen Grundlagen der Entwicklung betrieblicher IT-Systeme aufzugreifen und in den Methodikontext von SOA einzuordnen.

SOA und die Identifikation von Enterprise-Services

Eine der Grundaussagen der SOA-Bewegung lautet, dass Services zur Unterstützung von Geschäftsprozessen, die so genannten Enterprise-Services (Woods und Mattern 2006, S. 20), direkt aus den Geschäftsprozessen bzw. aus hinreichend verfeinerten Geschäftsprozessmodellen abgeleitet werden sollen.

Tatsächlich liegt Geschäftsprozessen seit jeher ein serviceorientiertes Verständnis zugrunde. Bereits Hammer und Champy (1993, S. 35) definieren einen Geschäftsprozess als "collection of activities that takes one or more kinds of input and creates an output that is of value to the customer". Ein Geschäftsprozess erzeugt einen Wert, einen Nutzen und damit einen Service für einen nachfragenden Abnehmer. Betriebliche Leistungserstellung findet in einem Netz aus Haupt- und Serviceprozessen statt, an deren Schnittstellen Services (Güter, Dienstleistungen) bereitgestellt und übergeben werden.

Warum entstehen die Spezifikationen von Enterprise-Services dennoch nicht quasi automatisch aus den Blattknoten eines hinreichend verfeinerten Geschäftsprozessmodells? Die Antwort liegt in der

Differenzierung zwischen Aufgaben- und Aufgabenträgerebene eines betrieblichen Systems. Die Beschreibung eines Geschäftsprozesses erfolgt auf der Aufgabenebene. Die einzelnen Aufgaben dienen der Leistungserstellung oder ihrer Lenkung und sind anhand ihrer Aufgabenobjekte und Ziele spezifiziert. Enterprise-Services sowie bei nicht oder teilautomatisierten Aufgaben auch Personen sind hingegen der Aufgabenträgerebene zugeordnet. Sie stellen die Lösungsverfahren für die betrieblichen Aufgaben bereit.

Die Spezifikation von Enterprise-Services anhand der Aufgaben von Geschäftsprozessen erfordert damit Festlegungen hinsichtlich des Automatisierungsgrades der Aufgaben und der Wahl des Lösungsverfahrens für die automatisierbaren Aufgabenteile. Hier bestehen Freiheitsgrade und damit Gestaltungspotenziale für das betriebliche System. Diese können z. B. genutzt werden, um (generische) Enterprise-Services für Klassen ähnlicher Aufgaben zu entwerfen. Die strikte Trennung zwischen Aufgaben- und Aufgabenträgerebene erschließt somit Flexibilitäts- und Wiederverwendungspotenziale von SOA.

SOA und die Unterscheidung zwischen Geschäftsprozessen und Workflows.

Auch wenn die populäre Orchestrierungssprache für Webservices die Bezeichnung „Business Process Execution Language (BPEL)“ trägt, geht es dabei um die Ausführung automatisierter Workflows und nicht um Geschäftsprozesse. Ein Geschäftsprozessmodell spezifiziert ein Netz von Aufgaben, deren Durchführung eine betriebliche Leistung erzeugt. Ein Workflowschema beschreibt ein Netz von Arbeitsschritten als Lösungsverfahren für betriebliche Aufgaben. Ein BPEL-Prozess spezifiziert die automatisierte Steuerung von Webservices und damit den automatisierten Teil eines Workflows. Die Verkürzung der Beziehung Geschäftsprozess – Aufgabe – Workflow – Automatisierung – BPEL-Prozess auf ihre Ränder verstellt den Blick für methodische Zusammenhänge und verschenkt Gestaltungspotenziale. Die Auswirkungen dieser Verkürzung zeigt die aktuelle Diskussion um BPEL4People. Damit wird sozusagen im Nachhinein versucht, personelle Aufgabenträger in die Steuerung von BPEL-Prozessen einzubeziehen.

SOA und verteilte IT-Systeme

Verteilte Systeme umfassen autonome Komponenten, die durch Nachrichten inter-

agieren, d. h. lose gekoppelt sind. Keine Komponente besitzt die globale Kontrolle über das Gesamtsystem. Das z. B. von BPEL unterstützte Konzept der Orchestrierung beruht dagegen auf zentraler Koordination. Da jeder BPEL-Prozess aus Außensicht einen Webservice darstellt, der wiederum in einen BPEL-Prozess eingebunden sein kann, entsteht eine mehrstufig hierarchische Koordination. Nun stellen aber Geschäftsprozesse in der Regel selbst verteilte Systeme dar. Sie können autonome betriebliche Objekte umfassen, wie z. B. einzelne Produktionsstufen, die in einem Produktionsprozess zusammenwirken. Daraus folgt, dass Orchestrierung als alleiniges Konzept zur Spezifikation der Koordination in Geschäftsprozessen nicht ausreicht. Orchestrierung ist um ein komplementäres Konzept der Koordination autonomer Komponenten zu ergänzen. Ein solches Konzept ist die Choreographie, mithilfe derer die Zusammenarbeit von Peers beschrieben wird und die u. a. von der Sprache WS-CDL (Web Services Choreography Description Language) des W3C unterstützt wird. Im SOA-Umfeld wird Choreographie derzeit überwiegend für die Zusammenarbeit zwischen Geschäftsprozessen diskutiert, weniger jedoch innerhalb eines Geschäftsprozesses. In eine umfassende SOA-Entwurfsmethodik sind aber beide Formen der Koordination einzubeziehen. Die methodischen Grundlagen hierfür sind wiederum seit langem gegeben. So unterstützt die SOM-Methodik zur Modellierung von Geschäftsprozessen (Ferstl und Sinz 1995) eine Modellierung von Geschäftsprozessen als verteilte Systeme auf der Basis von nicht hierarchischer und hierarchischer Koordination.

SOA und persistente Daten

Über Jahrzehnte war die Datenintegration das bestimmende Integrationskonzept (Ferstl und Sinz 2006, S. 229 – 237) für betriebliche IT-Systeme. Der bekannten Dreiebenen-Schemaarchitektur folgend, werden die das Unternehmen abbildenden Datenstrukturen global in einem (unternehmensweit) integrierten konzeptuellen Datenschema beschrieben. Über diesem werden anwendungsspezifische externe Datenschemata spezifiziert, welche die Datenstrukturen für die einzelnen Anwendungsfunktionen bereitstellen. In den SQL-Sprachelementen CREATE {SCHEMA | VIEW} sind die „Versteinerungen“ dieses Konzepts bis heute gegenwärtig. Die einzelnen Anwendungsfunktionen sind über gemeinsame persistente Daten, d. h. überlap-

pende externe Schemata, eng gekoppelt. Über diese gemeinsamen Daten erfolgt auch die Kommunikation zwischen den Anwendungsfunktionen: Daten, die von einer Anwendungsfunktion geschrieben werden, werden von einer anderen gelesen.

Gegen Ende der 1980er-Jahre wurden die Grenzen dieses Integrationskonzepts – zumindest auf unternehmensweiter Ebene – sichtbar. Gleichzeitig zeichneten sich mit der Objektorientierung und der Komponentenorientierung Alternativen ab. Speziell die Objektorientierung beruht auf Datensicht auf dem seit den 1970er-Jahren bekannten Konzept des Abstrakten Datentyps (ADT) (Parnas 1972; Liskov und Zilles 1974). Die in einer ADT-Instanz gekapselten Daten werden ausschließlich von den Operatoren des ADT manipuliert. Seiteneffekte auf Daten außerhalb der ADT-Instanz oder ADT-Instanzen mit gemeinsamen Daten sind ausgeschlossen.

Das Konzept der Webservices stellt lediglich die angebotenen Dienste in den Mittelpunkt, lässt jedoch deren Implementierung und damit die Frage der persistenten Daten offen. Die aktuelle SOA-Diskussion geht in Richtung einer Komponentenarchitektur (siehe z. B. das von namhaften Unternehmen unterstützte Konzept Service Component Architecture SCA). Dennoch bleiben viele Fragen offen, z. B. wenn Altsysteme (Legacy Systems) unter Nutzung von Serviceschnittstellen verpackt und für die Weiterverwendung in einer SOA aufbereitet werden. Was hinter diesen „vorgemauerten Fassaden“ geschieht, dürfte in den wenigsten Fällen wirklich kontrollierbar sein. Obwohl Datenseparierung als SOA-Ziel genannt wird, ist eine Datenkapselung im strengen Sinne in vielen Fällen wohl nicht erreichbar. Enge Kopplung über gemeinsame Daten, Seiteneffekte von Diensten und damit Probleme mit der globalen Zustandskonsistenz des verteilten Systems sind die Folge.

Fazit: Serviceorientierung und SOA bieten große Potenziale für Fortschritte in den eingangs skizzierten Themenfeldern. Angesichts der Unschärfe des Begriffs SOA und der zu beobachtenden methodischen Verkürzungen droht jedoch die Gefahr, dass SOA auf eine Ernüchterungsphase zusteuert. Eine ganzheitliche SOA-Gestaltungsmethodik muss auf den bewährten Grundlagen der Entwicklung betrieblicher IT-Systeme aufbauen und diese im SOA-Kontext neu interpretieren.

Prof. Dr. Elmar J. Sinz
Universität Bamberg

Literatur

- Enslow, Philip H. (1978): What is a ‚Distributed‘ Data Processing System? In: IEEE Computer 11 (1), S. 13 – 21.
- Erl, Thomas (2007): SOA. Principles of Service Design. Prentice Hall, Upper Saddle River, NJ.
- Ferstl, Otto K.; Sinz, Elmar J. (1995): Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: WIRTSCHAFTSINFORMATIK 37 (3), S. 209 – 220.
- Ferstl, Otto K.; Sinz, Elmar J. (2006): Grundlagen der Wirtschaftsinformatik. 5. Aufl., Oldenbourg, München.
- Hammer, Michael; Champy, James (1993): Reengineering the Corporation. A Manifesto for Business Revolution. HarperBusiness, New York.
- Krafzig, Dirk; Banke, Karl; Slama, Dirk (2005): Enterprise SOA. Service-Oriented Architecture Best Practice. Prentice Hall, Upper Saddle River, NJ.
- Liskov, Barbara; Zilles Stephen (1974): Programming with Abstract Data Types. In: SIGPLAN Notices 9 (4), S. 50 – 59.
- Parnas, David L. (1972): On the criteria to be used in decomposing systems into modules. In: Communications of the ACM 15 (12), S. 1053 – 1058.
- Pulier, Eric; Taylor, Hugh (2006): Understanding Enterprise SOA. Manning, Greenwich.
- Siedersleben, Johannes (2007): SOA revisited: Komponentenorientierung bei Systemlandschaften. In: WIRTSCHAFTSINFORMATIK 49 (Sonderheft), S. 110 – 117.
- Starke, Gernot; Tilkov, Stefan (Hrsg.) (2007): SOA-Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg.
- Woods, Dan; Mattern Thomas (2006): Enterprise SOA: Designing IT for Business Innovation. O‘Reilly, Sebastopol CA.

SOA revolutioniert das Management

Jürgen Laartz

In technischer Hinsicht sind serviceorientierte Architekturen kein revolutionärer Neuansatz: IT-Entwickler verfolgen seit vielen Jahren das Ziel, Funktionalität und damit Services sowohl orts- als auch plattformunabhängig kapseln und unternehmensübergreifend anbieten zu können. Sie knüpfen dabei an durchaus erprobte Softwarekonzepte zur Komplexitätsbewältigung an. Revolutionär ist dieses neue Instrument jedoch aus betriebswirtschaftlicher Sicht. Denn dank SOA können Unternehmen deutlich effizienter agieren sowie schneller und flexibler auf Veränderungen im Wettbewerb reagieren. Zudem hilft SOA, Kosten zu sparen.

1 SOA ermöglicht Unternehmen wesentliche Verbesserungen

Was ist für das Management so neu an SOA? Das Entscheidende ist die Tatsache, dass serviceorientierte Architekturen die Fixierung auf Prozesse durchbrechen, zugleich aber die technische und geschäftliche Komplexität im Unternehmen reduzieren. So lassen sich beispielsweise Betriebsabläufe und IT als Blueprint abbilden, ohne dass hierfür eine Systemsprachenkenntnis der IT-Seite notwendig wäre. Grundlage für solche Verbesserungen sind vor allem vier Fähigkeiten:

1. SOA segmentiert Geschäftsabläufe in abgeschlossene Einheiten (Services), die untereinander mit klarem Interface kommunizieren – und optimiert so die Transparenz, Vergleichbarkeit und letztlich Austauschbarkeit von fachlicher Funktionalität.
2. SOA kann vorhandene Services wiederverwenden und weiterentwickeln – und spart damit Zeit, Geld und Mühe.
3. SOA bildet Servicecluster (Domains) aus natürlich zusammenhängenden Einheiten – und minimiert so die Kommunikation zwischen den Einheiten.
4. SOA vereinheitlicht Sprache und Kommunikation zwischen IT und Management auf Grund einer Governance, die Standards setzt und das Einhalten von Servicegraden reguliert – und hilft folglich, Missverständnisse und somit Nacharbeiten zu vermeiden.

Diese Fähigkeiten tragen insgesamt zu mehr Effizienz in der Interaktion zwischen IT- und Geschäftsseite bei, weil Letztere Verantwortung für die Domains übernimmt und damit Kontrolle über die entsprechenden IT-Services und -Daten erhält.

2 SOA dient Unternehmen als Managementtool

Doch mit serviceorientierten Architekturen können Unternehmen noch einen Schritt weiter gehen: Jenseits der rein technischen Möglichkeiten erweist sich SOA als veritables Managementinstrument. Denn nachdem man für jeden Service ermittelt hat, welchen Wert er für das Unternehmen besitzt, lässt sich eine IT-Strategie ableiten, die Priorisierungs- oder Sourcingentscheidungen vereinfacht sowie eindeutige Anforderungen an die IT und die einzelnen Domains stellt. Zwei Services gilt es dabei zu unterscheiden:

- Standardservices (auch Commodity Services genannt): Hier können Firmen ihre Kosten senken, indem sie die Wiederver-

wendung maximieren und die Komplexität reduzieren (d.h. Redundanzen tilgen oder Änderungen punktuell durchführen, also ohne den gesamten Code zu modifizieren).

- Differenzierte Services (im Hinblick auf Wettbewerber): Hier erreichen Unternehmen mehr Flexibilität und Agilität (z.B. bei der Einführung neuer Produkte), weil die Kapselung der einzelnen Services nur noch lokale Updates erforderlich macht. Mit Blick auf die Einführung von SOA bietet die Kapselung der Services einen weiteren Vorteil: Sie ermöglicht ein schrittweises Vorgehen, bei dem ein Service (z.B. für Kundendaten oder Terminvergabe) zunächst als Pilot zur Erprobung des Gesamtkonzepts dienen kann. Die Erfahrung lehrt, dass bereits dieser Pilot auf einen konkreten Geschäftsbedarf hin ausgerichtet sein sollte, weil sich der Nutzen einer rein technischen Umsetzung ohne sichtbaren Geschäftsbezug dem Management kaum erschließt.

3 Zum Durchbruch fehlt es SOA-Netzwerken noch an kritischer Masse

Zu guter Letzt bergen serviceorientierte Architekturen auch das Potenzial für einen noch weiter reichenden Einsatz. Denn sie lassen sich über Unternehmensgrenzen und IT-gestützte Prozesse hinaus als Instrument zur Beschreibung von Services allgemein nutzen – ganz so, wie es in den Service Sciences angedacht wird.

Wie sähe ein entsprechender Markt für die Wiederverwendung von Services aus? Im Mittelpunkt stünden wahrscheinlich weniger zentrale Registries als eine Reihe von industriespezifischen „Servicebibliotheken“, die von Vermittlern angeboten werden. Um sich in der Flut der Angebote zurechtzufinden, nutzen die Akteure spezielle Suchmaschinen, die Ähnlichkeiten von Services erkennen und mögliche Interessenten identifizieren. Hierbei könnten zum Beispiel „social techniques“, wie etwa eine selbstorganisierte Bewertung der Services durch Abonnenten, eine Hilfe sein.

Von einem solchen Angebot würden alle beteiligten Unternehmen profitieren, indem sie schneller, kosteneffizienter und somit wettbewerbsfähiger agieren könnten. Für kleinere spezialisierte Anbieter täte sich ein Absatzmarkt auf, was wiederum dem Mittelstand zugute käme. Schließlich würden Möglichkeiten wie das Offshoring von Services erleichtert, wodurch sich Mehrwert schaffen und höhere Produktqualität erreichen ließe.

Die Frage drängt sich auf: Warum werden bislang noch kaum Services unternehmensübergreifend angeboten? Die Antwort ist unspektakulär: Ein nutzbringender breiter Einsatz von SOA erfordert allgemein akzeptierte Standards. Doch zur Etablierung solcher Normen fehlt es bislang an schierer Masse. Auf einigen Feldern könnten Softwareanbieter die Aufgabe des Standardisierens übernehmen – aber noch ist es keinem Anbieter gelungen, genügend Vertrauen aufzubauen, um dies erfolgreich zu meistern.

Wie sich ein solcher Standard dennoch verankern lässt, zeigt das Beispiel RosettaNet: Als eine globale, von mehr als 500 Unternehmen getragene Organisation entwickelt RosettaNet Standards für das Supply Chain Management. Treibende Kraft war und ist die Einsicht, dass diese Normen allen Beteiligten nützen. Die XML-Technologie ermöglicht deren Verankerung. Die Standards selbst haben sich aus über 2.000 Implementierungen herausgebildet, sind also unmittelbar aus den Anforderungen der Industrie abgeleitet – und werden kontinuierlich an neue Bedingungen angepasst. Ihr jeweiliger Nutzen wird allen Mitgliedern rasch anhand von Beispielen vermittelt. Unterm Strich erzielen die beteiligten Unternehmen dank Standardisierung spürbare Effizienzsteigerungen im Supply Chain Management.

Fazit: Serviceorientierte Architekturen übertragen Best Practices der IT auf Geschäftsprozesse. Dies minimiert die Reibungsverluste an der Schnittstelle zwischen Management und IT. Konsequenter eingesetzt, ermöglicht SOA den Unternehmen sowohl höhere Effizienz als auch mehr Flexibilität im Wettbewerb.

Dr. Jürgen Laartz

Director Business Technology Office
McKinsey & Co., Berlin

Der Paradigmenwechsel ist in vollem Gange

Karl-Heinz Streibich

Auch wenn in Anbetracht von geschätzten 9 Milliarden Euro Umsatz weltweit für 2006 und einem bisher noch relativ geringen Anteil an Integrationsprojekten mit SOA-Anteil in Deutschland von etwa 15 Prozent der Anschein erweckt werden könnte, SOA wäre lediglich ein Marketing- und Medienhype, so ist der Paradigmenwechsel bei-

spielsweise im Finanz- oder Logistiksektor bereits in vollem Gange. Hersteller, Analysten und in steigendem Maße Anwenderunternehmen bewerten serviceorientierte Architekturen als Treiber für Wachstum und Innovation.

SOA ist keine Technologie und kein Produkt, welches sich in kürzester Zeit in einem Unternehmen einführen lässt. Das Konzept steht vielmehr für die effiziente Anpassung der Anwendungssysteme an sich ändernde Geschäftssysteme. Der Erfolg eines Unternehmens hängt in zunehmendem Maße von der Flexibilität der Organisationsstrukturen und der Geschäftsprozesse sowie der genutzten IT-Landschaft ab. Änderungen der Geschäftsprozesse schnell umsetzen zu können, ist unter anderem der Grund für eine in vielen Branchen rapide wachsende Marktdynamik.

Kunden und Anwender, welche in den vergangenen Jahrzehnten mit einer nicht enden wollenden Kette von aufwändigen Entwicklungsprojekten konfrontiert waren, nehmen heute ihr Schicksal selbst in die Hand und modernisieren die vorhandenen Anwendungssysteme anstelle riskanter und umfangreicher Neuentwicklungen von unternehmenskritischen Anwendungssystemen.

Die entscheidende Neuerung an serviceorientierten Architekturen gegenüber vielen bisherigen Ansätzen aus dem IT-Umfeld ist die kompromisslose Fokussierung auf den Internet-Technologiestandard Webservices.

Für den erfolgreichen Paradigmenwechsel ist die Antwort auf die Frage notwendig, wie das zugrunde liegende Konzept der Serviceorientierung durch alle Beteiligten – Management, Fachabteilungen und IT-Verantwortliche auf Kundenseite – akzeptiert und umgesetzt wird. Aktuell ist das Wissen rund um das SOA-Konzept in vielen Unternehmen, insbesondere in den Fachabteilungen, noch nicht ausreichend für eine umfassende Einführung.

Die Entscheidung für SOA ist die Wahl einer IT-Strategie mit deutlichen Auswirkungen auf die Unternehmensorganisation, die zukünftig das Zusammenspiel zwischen IT und Business bestimmen wird. Anforderungen aus den Fachabteilungen und dem Management werden zum Treiber der IT. Die ständige und gleichzeitige effiziente Anpassung von bestehenden Systemen basierend auf unternehmensweitem Geschäftsprozessmanagement wird in vielen Branchen den Umgang mit IT verstärkt beeinflussen. Klare Strukturen

mit exakt definierten Verantwortlichkeiten sind notwendig. Change-Management auf Organisationsebene nimmt somit eine Schlüsselrolle bei der erfolgreichen SOA-Umsetzung ein.

Sind die organisatorischen Rahmenbedingungen auf Kundenseite geschaffen, muss aber nicht die IT-Landschaft ersetzt werden. Eine Vielzahl von bestehenden Anwendungen mit ihren Daten und bereits implementierten Prozessen kann in die neu strukturierten IT-Landschaften integriert werden. Legacy-Systeme werden dadurch mittels Services-Layer über deren geplante Lebenszeit hinaus nutzbar.

Die erfolgreiche Einführung von SOA führt so auch zu einer Steigerung der Investitionssicherheit in Bezug auf die verschiedenen Komponenten einer unternehmensweiten IT-Landschaft. Zusätzlich helfen serviceorientierte Architekturen bei der Beherrschung der immensen Komplexität in modernen Unternehmen und reduzieren den Anteil von IT-Kosten bei der Anpassung von Prozessen auch über Unternehmensgrenzen hinweg.

Das Konzept SOA ist nach anfänglich sehr technisch geprägter Diskussion erwachsen geworden. Ein zunehmendes Verständnis auf Kundenseite in Kombination mit umfassenden Lösungen rund um serviceorientierte Architekturen wird den kompletten Bereich der unternehmenskritischen Anwendungssoftware nachhaltig verändern. Software erreicht mit SOA einen neuen Reifegrad gemäß der gemeinsamen von Herstellern und Kunden formulierten Zielsetzung „Business drives IT“.

Karl-Heinz Streibich
Mitglied des Präsidiums des
BITKOM e.V., Berlin
Vorsitzender des Vorstands,
Software AG, Darmstadt

Services Must Become More Agent-Like

Michael N. Huhns

The design or reengineering of information systems according to the concepts of service-oriented architectures (SOA) and the development or deployment of systems according to the principles of service-oriented computing (SOC) offer great potential benefits. However, except in special cases, the benefits have not been realized. This essay describes several impediments

that must be overcome in order to achieve the benefits and makes the claim that an agent-oriented basis for services can remove the impediments.

1 Introduction

The practice of information technology is rapidly evolving toward service-oriented computing, where components are understood as services that can be developed independently, selected and composed dynamically, and executed in a context sensitive manner according to the prevailing context. The vision is that such services will be widely available in our home, office, and recreational environments. To this end, the following are some of the major classes of applications where services arise:

- Heterogeneous information management within and, increasingly, across enterprises, thereby facilitating superior process management and e-business scenarios.
- Scientific computing, involving large, dynamically reconfigurable resources, thereby supporting the solution to grander scientific problems, such as in physics or bioinformatics.
- Mobile computing, where users are mobile and need to access information resources that may be fixed or mobile, thereby providing the right information at the right time to users.
- Pervasive computing, where computational resources are associated with components of our surrounding physical infrastructure, thereby leading to optimized management of resources and an improved experience for users.

The widespread deployment of such applications requires an infrastructure that is being steadily provided, albeit more slowly than desired or expected, by the Semantic Web initiative. In particular, the Semantic Web is making available widespread and more comprehensive ontologies, which will form models for numerous real-world entities and systems, as well as the mean-

ings for documents and content. But this will not address all problems.

2 Impediments to widespread adoption of SOA and SOC

There are several major problems with the current view of services and their use, as follows:

- There is both a conceptual gap and an engineering gap between the perception of how services are composed into a workflow and the reality of how the composition actually takes place (see Figure 1)
- A common misconception is that services in a workflow interact directly, but they actually interact via a controller that provides synchronization, intermediate storage, and error recovery.
- There has been some success with the use of SOA within a local organization, where the services being deployed can be designed, managed, and controlled by the local IT staff, but the success has not translated to SOA across enterprises:
 - Companies that have repackaged internal functionality as Web services have not yet figured out how to market them externally to yield revenue streams.
 - Companies with needs for external Web services have not figured out how to deal with their autonomy, i.e., they are under the control of an external organization.
 - Legal contracts for external services have not been well defined and standardized.
 - The discovery, selection, and programmed invocation of services is still done manually in advance by developers, rather than done automatically and as needed. Moreover, services are being developed by putting a SOAP/WSDL veneer on existing applications or components, which considers only their deployment.
 - Quality of service (QoS) is mentioned widely, but there is no clear definition for it and its parameters, so that contracts for services cannot be based on it.

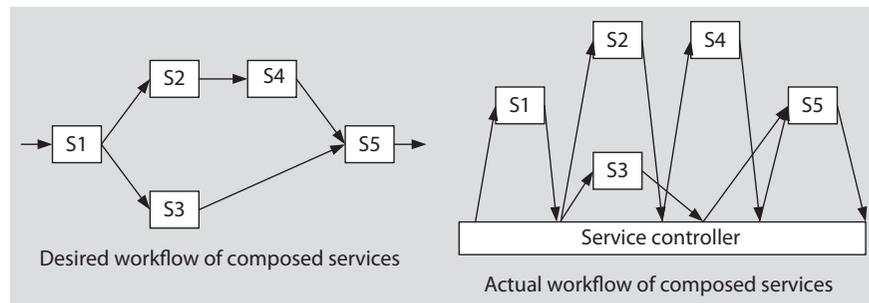


Figure 1 Composed services

My students have been able, in just one day, to encode an application that includes software developed by hundreds of programmers from different organizations across the Web. As in traditional software engineering, their application might have made use of generic modules from standard software libraries, which they would have then customized and specialized for their needs. But more than this, their application can include domain-specific modules (Web services) that can be invoked as is, i.e., without any customization. Moreover, the modules will be maintained by others. The capability to do this is unprecedented.

However, the result has none of the usual qualities by which software is measured, such as the „ilities“ from software engineering. And, the delegation of maintenance to the service provider means no control or supervision over how and when maintenance is done.

3 Agents can overcome the impediments

The impediments described above can be overcome by converting individual services into software agents, and the composition of services into multiagent systems (Zavala and Huhns 2004; Singh and Huhns 2005). How might this happen? The answer lies in a consideration of the three primary development phases for a SOA-based system, shown in Figure 2 and described as follows:

- Candidate service discovery is the distributed search for available services that can (potentially) accomplish some set of a client's internal goals or objectives. One architectural approach to this phase is interaction with a semantic matchmaker or registry agent, or a peer agent serving either function.
- Service engagement includes the process of interpreting candidate Web service enactment constraints (partly or fully described in each service's published self-description), and then negotiating with prospective services until an agreement is reached. This phase concludes when both service and client agree to the service-provision terms in an explicit or implicit contract. Negotiations can include service price, the quality and timeliness of the service, and security and privacy issues.
- Service enactment is the process that completes the mutually agreed objectives of the client and service by following the service's published protocols. If the contract's primary objectives are not accomplished, then the client and service can use a compensation protocol to restore fi-



Figure 2 Service interaction process

ancial equity or a stable operating state. If the underlying message transport mechanisms allow for asynchronous interactions, then the client can use protocols to monitor the service's status during execution. The enactment phase includes the ongoing management of the service and its use. The three main phases of interaction (discovery, engagement, and enactment) are driven by alternative characterizations of client and service provider goal descriptions communicated during message exchanges using a set of widely shared semantic concepts.

The following attributes of multiagent systems can make each phase "industrial strength:"

- Agents are often self-aware at a metalevel, and through learning and model building gain awareness of other agents and their capabilities as interactions among the agents occur. A Web service knows about itself, but not about its users/clients/customers. An agent-based service could take advantage of new capabilities in its environment and customize its service to a client, such as by providing improved services to repeat customers.
- Agents can be designed to reconcile ontologies, whereas if the client and provider of a conventional service happened to use different ontologies, then the result of invoking the service would be incomprehensible to the client.
- Via agent-based software engineering, MAS can form the fundamental building blocks for software systems, even if the software systems do not themselves require any agent-like behaviors.
- Agents are inherently communicative, whereas Web services are passive until invoked. Agents can provide alerts and updates when new information becomes available.
- Agent-based systems have made use of service level agreements and quality of service measures that build upon work in the areas of automated negotiation and flexible service execution in dynamic and uncertain environments.
- Agents are cooperative and, by forming teams and coalitions, can provide higher-level and more comprehensive services dynamically.

- Autonomy is a characteristic of agents, and it is also a characteristic of many envisioned Internet-based applications. An agent-based service would attempt to coordinate with other services where appropriate and to keep its commitments as much as possible, but it would exercise its autonomy in entering into those commitments in the first place, thus enabling QoS guarantees.

4 Conclusions

Multiagent systems offer a set of computational abstractions – such as cooperation, negotiation, and commitment – based on fundamental concepts derived from societal interactions and economies. These enable the description, simulation, and understanding of human-scale systems, which is precisely what is needed for Internet-scale service-based systems. As inter-enterprise applications become more complex, it will be increasingly difficult for one server to provide a total solution and increasingly difficult for one client to integrate solutions from many servers. Web services currently involve a single client accessing a single server, but soon applications will demand federated servers with multiple clients sharing results. Cooperative peer-to-peer solutions will have to be managed, and this is an area where agents have excelled. In doing so, agents can balance cooperation with the interests of their owner.

Prof. Dr. Michael N. Huhns
University of South Carolina, Columbia
Center for Information Technology

References

- Singh, Munindar P.; Huhns, Michael N.* (2005): *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Hoboken.
- Müller, Günter* (2005): Günter Müller referiert: *Service-Oriented Computing: Key Concepts and Principles*. In: *WIRTSCHAFTSINFORMATIK* 47 (5), pp. 448 – 449.
- Zavala, Rosa Laura; Huhns, Michael N.* (2004): *On Building Robust Web Service-Based Applications*. In: *Cavedon, Lawrence; Maamar, Zakaria; Martin, David; Benatallah, Boualem* (Ed.): *Extending Web Services Technologies: The Use of Multi-Agent Approaches*. Kluwer Academic Publishers, New York, pp. 293 – 310.

SOA – Flexibility and Agility

Orestis Terzidis, York Sure,
Christian Brelage

Whether Service-Oriented Architecture (SOA) is new or not or even a useful concept at all is discussed since it is being proposed by software vendors as a concept to manage growing complexity and heterogeneous, distributed enterprise software infrastructures. It is necessary to review current challenges and trends as well as to discuss technical and business aspects of SOA in order to analyze its value proposition.

1 Challenges and trends: drivers of SOA

Companies are facing a shift from hard-wired value chains towards flexible Business Value Networks. Traditionally, value chains were backed-up by long-running master agreements and stable production lines, customers and distribution channels. Core to the creation of value was the increase of the production process efficiency. Information systems of companies reflected the stable situation by hard-wiring all relevant business processes.

The trend of specialisation and reduction to core competencies led to outsourcing of services and processes. This trend is continuing with an increased pace. Today companies are organized in flexible business value networks and are acting on a global scale. The production machineries adapt quickly to changing market situations. The logistic chains and distribution channels are continuously optimized. Flexibility and agility are sine qua non for companies (Kagermann and Österle 2006).

Powerful drivers for long-term growth and competitive advantage of companies are the acceleration of innovation, the development of operational excellence and the empowering of the information worker. Flexible information system architectures are the key enablers to execute these strategies better and faster than competitors and critical for the success of a company.

Ideally the information system architecture executes the business model of a company and allows for efficient and effective change management at low costs. Examples for change management needs include the following: A supplier is required to implement an RFID-based logistic chain within few months to avoid price reductions, or, a company needs to quickly react on the need for compliance with the Sarbanes-Oxley-Act, IFRS or FDA. SOA is both,

a technical and a business concept, which allows for flexible information system architectures and enables agility.

2 Technical and business dimensions

SOA modularizes monolithic software packages into manageable application components which are offered as standardized Web Services in an expandable repository. SOA supports the flexible and dynamic composition of processes, interfaces, reports and electronic forms. By adopting SOA, companies can implement business improvements more quickly, while still supporting existing business processes. With SOA, organizations can (SAP 2007):

- Link stand-alone processes to quickly compose new end-to-end processes,
- Selectively redesign existing processes to make changes faster, and
- Facilitate seamless process design and execution across company boundaries to leverage the core competencies and best practices of business partners.

SOA is nearly as old as computing itself from a technical perspective. The underlying technical concepts, including modularization, standardisation and abstraction, (re)appear regularly in new programming languages, standards and products and are not new as such. However, SOA also has a new technical aspect which is the primary reason for its success and quick adoption: it is completely based on existing internet standards and, thus, blends seamlessly into any web based infrastructure.

Although SOA is primarily seen as a technical concept to design, implement and manage software architectures it also has business implications. From a technical perspective SOA is agnostic to business content, i.e. business services, processes and data, and does not provide direct value as such. However, it clearly is an enabling technology for the challenges outlined above and, thus, can generate substantial value. Additionally, SOA can be extended by a business dimension. For instance, SAP proposes the Enterprise SOA (ESOA) concept which can be seen as a combination of the (technical) SOA concept and business content in form of services, processes and business objects which standardize business operations and represent substantial value for organizations.

3 Future trends

Although SOA is just beginning to find adoption in the marketplace, it is possible to identify future trends from a research perspective which may indicate future developments.

Considerable effort is being made in order to make web services accessible for humans as well as computers by applying semantics (Hitzler et al. 2007; Studer et al. 2007). Currently, services and their descriptions (mainly WSDL) describe the functionality on a mere technical level. Semantic annotations of web services, however, try to describe what the web services does and which pre and post conditions are relevant for its execution. Thus, computers and humans are supposed to better “understand” the web service and its functionality. Moreover, semantic annotations can be used to aid software engineers in discovering and composing web services.

New, disruptive applications of SOA principles and concepts are discussed under terms like “Internet of Services” and “Service Science”. To fully enable future business value networks, services, similar to products, need to become tradable goods. Thus, services need to be composable into (complex) value-added services which can be offered, discovered and consumed via the internet.

4 Summary

Given the challenges outlined we believe that the adoption of SOA concepts is a strategic necessity for organizations. Today, SOA is a crucial IT enabler for new business models, processes and flexibility and, thus, provides indirect value for organizations. Whether SOA is completely new or not is of no practical relevance respectively the wrong question to the right answer. However, SOA is only the next logical step in a continuous innovation cycle which has to be augmented technically, for instance by extending SOA by semantics, as well as from a business perspective, e.g. through the provision of additional business content in form of useful services, in order to provide additional business value.

Dr. Orestis Terzidis, Dr. York Sure, Dr. Christian Brelage, SAP Research, CEC Karlsruhe

References

- Hitzler, P.; Krötzsch, M.; Rudolph, S.; Sure, Y. (2007): *Semantic Web – Grundlagen*. Springer, Berlin.
- Kagermann, H.; Österle, H. (2006): *Geschäftsmodelle 2010. Wie CEOs Unternehmen transformieren*. Frankfurter Allgemeine Buchverlag, Frankfurt.
- SAP (2007): *Enterprise Service-Oriented Architecture from a Business Perspective*. SAP White Paper.
- Studer, R.; Grimm, S.; Abecker, A. (2007): *Semantic Web Services: Concepts, Technologies, and Applications*. Springer, Berlin.

Kompakt, verständlich, IT-gestützt



Jürgen Bauer, Egbert Hayessen

Controlling für Industrieunternehmen

Kompakt und IT-unterstützt - Mit SAP®-Fallstudie

2006. XV, 240 S. mit 170 Abb. Br. EUR 39,90
ISBN 978-3-8348-0067-1

Der Inhalt

Strategisches Controlling - Grundlagen Kostenmanagement und Kostencontrolling mit und ohne SAP-Einsatz - Balanced Scorecard - Grundlagen Value Controlling - Anwendung Deckungsbeitragsrechnung und BEP-Analyse - Grundlagen und SAP-Unterstützung in Materialcontrolling, Termin- und Kapazitätscontrolling - Projektcontrolling mit MS-Project und SAP-Systemen - Investitionscontrolling - Controllingreorganisation

Das Buch

Ein kompakter und leicht verständlicher Einstieg in das IT-gestützte Controlling eines jeden Industrieunternehmens. Das Buch zeigt in kompakter Form die Grundzüge des modernen Controllings unter Nutzung von IT-Ressourcen - am Beispiel von SAP. Die Methoden und Werkzeuge werden anhand von Praxisbeispielen erläutert. Studierende und Praktiker profitieren von der guten Übersichtlichkeit und dem durchgängigen Anwendungsbezug. Zu den Inhalten gehört u. a. das rationelle TCO-Kostenmanagement und Werkzeuge eines Investitions- und Projektcontrollings. Eine Fallstudie vermittelt in nachvollziehbaren Schritten das Vorgehen beim Kosten- und Logistikcontrolling.

FAX-Bestellung 0611.7878-439

Ja, hiermit bestelle ich:

Änderungen vorbehalten

Bauer/Hayessen
Controlling für Industrieunternehmen
2006. EUR 39,90 (zzgl. Versand)
ISBN 978-3-8348-0067-1



Abraham-Lincoln-Str. 46
65189 Wiesbaden
www.vieweg.de
Fax: 0611.7878-439

Geschäftsführer Andreas Kösters, Dr. Ralf Birkelbach
AG Wiesbaden HRB 9754

Vorname/Name	

Firma	Abteilung
_____	_____
Straße	

PLZ/Ort	

Telefon/Fax	

Unterschrift	Datum
_____	_____