

# Zweitveröffentlichung



Sinz, Elmar J.

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

Datum der Zweitveröffentlichung: 26.08.2024

Akzeptiertes Manuskript (Postprint), Zeitschriftenartikel

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-975553

### Erstveröffentlichung

Sinz, Elmar J. (2008): „SOA und die bewährten methodischen Grundlagen der Entwicklung betrieblicher IT-Systeme“. In: Wirtschaftsinformatik : WI, Jg. 50, Nr. 1, S. 70-72, Wiesbaden: Springer Gabler, doi: 10.1007/PL00022213.

### Verlagshinweis

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <http://dx.doi.org/10.1007/PL00022213>.

### Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt das deutsche Urheberrecht.

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

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 Web-Services. 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 Methoden-Kontext 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 in ihrem Buch “Reengineering the Corporation” 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 Web-Services 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 Workflow-Schema beschreibt ein Netz von Arbeitsschritten als Lösungsverfahren für betriebliche Aufgaben. Ein BPEL-Prozess spezifiziert die automatisierte Steuerung von Web-Services 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 interagieren, 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 Web-Service 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 Drei-Ebenen-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. überlappende 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 aus 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 Web-Services 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 Service-Schnittstellen 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 Zustandkonsistenz des verteilten Systems sind die Folge.

Fazit: Serviceorientierung und SOA bieten große Potenziale für Fortschritte in den eingangs skizzieren 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.

*Elmar J. Sinz  
Universität Bamberg*

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.