

BUSINESS SERVICE MODELLIERUNG

Effiziente SOA durch den konsequenten Einsatz fachlich getriebener Modelle

SOA (serviceorientierte Architektur) gehört zu den in den letzten Jahren wohl am meisten diskutierten Schlagwörtern. Nach der anfänglichen Euphorie hat sich eine – wohl heilsame – Ernüchterung eingestellt. SOA wird zunehmend kritischer betrachtet, und negative Stimmen häufen sich.

*Umso erstaunlicher ist es, dass ein ganz wesentlicher Themenbereich bisher ignoriert oder zumindest stiefmütterlich behandelt wurde: Die Frage, **welche** Funktionen als Services realisiert werden sollten; und **wie** die Services strukturiert sein sollten. Viele der bemängelten Probleme bei der SOA-Umsetzung lassen sich auf der Vernachlässigung dieser Themen zurückführen.*

*Die Business Service Modellierung beschreibt einen Ansatz, der **Serviceidentifikation** und **Service design** vereint und aus konsequent fachlich orientierter Sicht behandelt. Die Durchführung der Business Service Modellierung ist dabei primär eine Herausforderung für IT Architekten.*

1 Ausgangssituation

SOA (Service-orientierte Architektur) gehört zu den in den letzten Jahren wohl am meisten diskutierten Schlagwörtern. Nach der anfänglichen Euphorie hat sich eine – wohl heilsame - Ernüchterung eingestellt, die mit inzwischen fundierten praktischen Erfahrungen einhergeht. Themen wie Web Services, ESB, Registry, etc. sind sicherlich nicht mehr als Neuland zu bezeichnen. Die technischen Grundlagen für SOA-Implementierungen werden mehr und mehr beherrscht, aber das "Plateau der Produktivität" scheint, obwohl in Sichtweite, immer noch entfernt [1]. Gleichzeitig wird die Enttäuschung über die bisher nicht erreichte Realisierung der vielen Versprechungen thematisiert und sorgt nach dem übertriebenen Hype der letzten Jahre für eine kritischere Einstellung zu SOA [2].

Umso erstaunlicher ist es, dass ein ganz wesentlicher Themenbereich bisher ignoriert oder zumindest stiefmütterlich behandelt wurde: Die Frage, **welche** Funktionen als Services realisiert werden sollten; und **wie** die Services strukturiert sein sollten. Ohne klare Richtlinien ist der Aufbau eines effizienten Service Portfolios kaum denkbar. Und obwohl SOA dazu geeignet scheint, die Kluft zwischen der IT und den Fachseiten zu überbrücken, tun sich genau hier wieder Gräben auf. Die IT wartet meist vergeblich auf konkrete fachliche Vorgaben zur **Serviceidentifikation**, und mancher fachlich Verantwortliche fühlt sich von der IT erst mit technischen Details überfrachtet und dann allein gelassen. Viele der bemängelten Probleme bei der SOA-Umsetzung lassen sich auf die Vernachlässigung dieser Themen zurückführen.

Wie sollen also Services identifiziert werden? Wie stellt man sicher, dass es die richtigen sind? Wie entwickelt man das Serviceportfolio? Und wer im Unternehmen soll dafür zuständig sein? Das sind genau die Fragen, zu denen es durchaus Antworten gibt – wir bezeichnen diesen Themenkomplex als **Business Service Modellierung**.

2 SOA in praktischen Projektsituationen

Nach Lehrbuch versteht man SOA fast immer als Unternehmensarchitektur, also als Ansatz für die übergreifende Strukturierung der IT eines ganzen Unternehmens. In der Praxis findet man heute – von einigen Ausnahmen abgesehen – selten eine so konsequente und umfassende Umsetzung. Häufig findet man SOA-Ansätze im wesentlich kleineren Rahmen vor: Serviceorientierung wird innerhalb von Teilbereichen eines Unternehmens, oder aber sogar nur im Rahmen einzelner Applikationen bzw. Lösungen verwendet.

Obwohl das gesamte Instrumentarium der SOA prinzipiell in Richtung Unternehmensarchitektur zielt, müssen solche Projekte mit kleinerem Anwendungsbereich („Scope“) nicht negativ bewertet werden. Ganz im Sinne der iterativen Einführung der serviceorientierten Architektur kann durchaus eine Unternehmens-SOA aus der Zusammenführung mehrerer kleinerer Ansätze entstehen.

Dies ist aber nur dann möglich, wenn frühzeitig auf eine saubere Modellierung, klare Begrifflichkeiten und ein aktives Management der jeweiligen Serviceportfolios Wert gelegt wird. Dies alles sind



Dr. Philipp Hoernes ist IT Architekt bei [ipt]. Er berät Unternehmen vorrangig bei der Konzeption und der Implementierung von SOA und Services. In der IT Branche ist er seit 1997 als Entwickler, IT Architekt und Business Consultant tätig

Aspekte der SOA Governance, also der Steuerung der Entwicklung einer SOA. Es ist bewährt, dabei zwischen Design-Time, Run-Time und Change-Time Governance zu differenzieren. Sowohl **Serviceidentifikation** als auch **Service design** sind Aspekte der Design-Time Governance, also der Festlegung von Regeln für an das Business angepasste Architektur und effizientes Design.

Die im Folgenden dargestellte Vorgehensweise ist zum einen geeignet, im jeweiligen definierten Anwendungsgebiet ein effizientes Service-Portfolio zu entwickeln. Sie bietet darüber hinaus aber auch die Möglichkeit, von vornherein mit einem größeren Horizont zu planen und - etwa durch Erstellung eines unternehmensweiten Domänenmodells – das konkrete Anwendungsgebiet in einem Projekt von Anfang an in Bezug zum Umfeld zu planen. Dadurch wird die Grundlage für eine spätere Erweiterung des Scopes gelegt.

3 Eine praktische Vorgehensweise zur Business Service Modellierung

3.1 Prozessmodellierung

Der Zusammenhang zwischen Geschäftsprozessen und Services wird immer wieder betont. Services sollen letztlich - durch „Orchestrierung“ – so kombiniert werden können, dass sie Prozesse bzw. Teile von Prozessen direkt unterstützen. Auf der anderen Seite wird „Wiederverwendung“ groß geschrieben: Ein und derselbe Service soll idealerweise im Kontext mehrerer unterschiedlicher Prozesse verwendet werden können.

Während man bei der Prozesssicht also von oben (Wertschöpfungskette, Kernprozesse) kommt und hierarchisch weiter detailliert (Teilprozesse, Prozessschritte, etc.) beginnt man mit feingranularen Services, die durch Orchestrierung zu immer komplexeren Services zusammengesetzt werden können. Dabei kann durchaus ein und dieselbe geschäftliche Aktion einmal als Prozess (Innensicht, Fokus auf den Ablauf), einmal als Service (Außensicht, Fokus auf die Ein- und Ausgangsartefakte) aufgefasst werden.

Von der Prozessmodellierung kann angesichts dieses komplexen Zusammenhangs nicht direkt auf Services geschlossen werden. Ein detailliertes Prozessmodell - also eines, das die Prozesse bis hinunter auf eine tiefe Ebene zerlegt - ist trotzdem ein wichtiges Werkzeug bei der Business Service Modellierung. Aus einem solchen detaillierten Prozessmodell kann nämlich abgeleitet werden, welche **Informationen** für jeweils einen Prozessschritt relevant sind. Jede Aktion (im Prozess) muss sich auf (eine) konkrete Entität(en) beziehen, und die Beschreibung dieser Entität(en) gehört zur Darstellung der unternehmenswichtigen Informationen. Beispiel: Ein Prozess „Kaufvertrag erstellen“ bezieht sich auf die Informationsobjekte „Kunde“, „Produkt“ und erzeugt ein Objekt „Kaufvertrag“.

3.2 Informationsmodellierung

Die im vorigen Abschnitt beschriebene Identifikation unternehmenswichtiger Informationsobjekte aus einem Prozessmodell ist nicht die einzige Alternative. In Ermangelung eines detaillierten Prozessmodells können Informationsobjekte anhand von Interfacebeschreibungen oder aus Interviews mit fachlichen Experten identifiziert werden. Es geht darum, möglichst vollständig aufzustellen, welche Informationen für den betrachteten Anwendungsbereich („Scope“) relevant sind.

Bei der „Informationsmodellierung“ geht es nun darum, aus den vorliegenden - in der Regel nicht strukturierten und vereinheitlichten - Beschreibungen von Informationsobjekten redundanzfreie Modelle zu erzeugen. Es gibt dabei mehrere Ausprägungen auf unterschiedlichen Ebenen:

1. **Businessdomänenmodell (BDM):** Eine Aufteilung der gesamten Informationslandschaft in (überschaubar viele) dis-

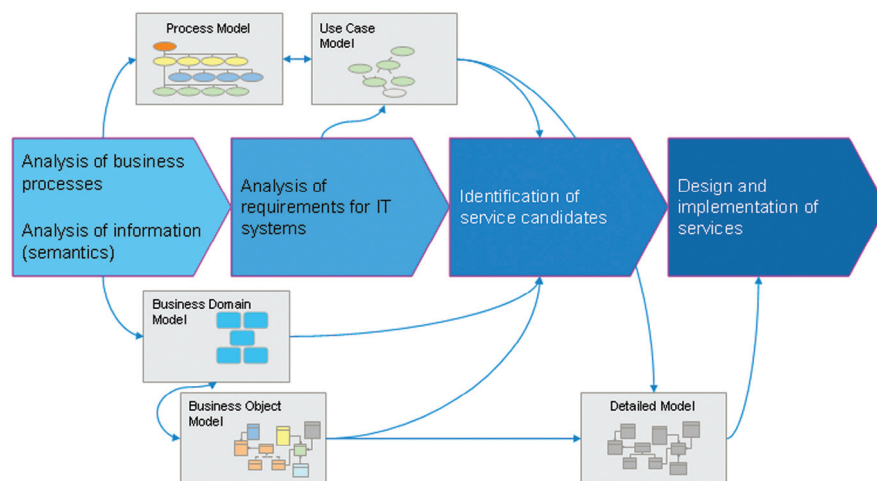


Abbildung 1: Übersicht über den Ansatz zur Business Service Modellierung

junkte „Businessdomänen“, in denen jeweils einheitliche Begrifflichkeiten verwendet werden. Businessdomänen können dabei bestimmten Organisationseinheiten, Funktionen oder auch Prozessen im Unternehmen zugeordnet werden, müssen aber nicht. Ein BDM ist vor allem im Hinblick auf einen unternehmensweiten bzw. zumindest lösungsübergreifenden Anwendungsbereich sinnvoll.

2. **Businessobjektmodell (BOM):** Ein Modell, das die aus dem Prozessmodell oder aus Interfaces extrahierten Informationen in Form von eindeutig definierten, redundanzfreien Businessobjekten und deren Beziehungen darstellt. Das BOM wird oft als UML-Klassendiagramm modelliert. Die einzelnen Businessobjekte müssen keine Attribute besitzen.
3. **Detailliertes Objektmodell:** Ein detailliertes Klassenmodell, auf dessen Basis die Interfaces von Services vollständig beschrieben werden können. Es baut auf dem BOM auf und detailliert dieses u.a. durch die Modellierung aller fachlich relevanten Attribute der Businessobjekte.

Typischerweise beginnt man mit dem BDM (1) und BOM (2) parallel. Beides sind Modelle, die zwar am besten durch IT-Architekten erstellt werden, die aber primär fachliche Bedeutung haben und von den Fachseiten verstanden werden müssen. Technische Objekte und Details haben darin nichts zu suchen, es geht ausschließlich um die Strukturierung fachlich bedeutsamer

Informationen.

Das detaillierte Objektmodell wird später erstellt, wenn das BOM bereits entwickelt ist und auch schon Services identifiziert wurden. Es dient zur detaillierten Darstellung der Service-Schnittstellen, für die das BOM zu wenig genau ist.

Im Zusammenhang mit der Erstellung des Businessdomänenmodells wird manchmal eine Diskussion über deren Zweckmäßigkeit geführt. Das Thema einer übergreifenden Informationsmodellierung ruft ungute Gedanken an die meist gescheiterten Versuche zur „Unternehmensweiten Datenmodellierung“ wach. Man sollte die Gegenargumente kennen:

- Ein Businessobjektmodell modelliert nicht alle unternehmensrelevanten Informationen, sondern nur diejenigen, die im Rahmen von Geschäftsprozessen verarbeitet werden, die also für Services relevant sind. Nur die Informationen im Anwendungsbereich der SOA werden betrachtet.
- Wenn man Services entwirft, wird durch deren Schnittstellendesign implizit automatisch ein Informationsmodell erstellt. Ein BOM ist nur dessen explizite Darstellung, die für Übersichtlichkeit und Verbindlichkeit sorgt und die Redundanzfreiheit der Modelle sicherstellt.

3.3 Identifikation von Service-Kandidaten

Wie im vorigen Abschnitt dargelegt, dient das BOM zum Design der Schnittstellen von Services. Es kann aber auch sehr gut als Hilfe zur Identifikation von Services



herangezogen werden! Doch zunächst eine Begriffsklärung. Unter einem „Service-Kandidaten“ versteht man einen potentiell nützlichen Service, für dessen Funktionalität fachliche Anforderungen existieren. Ein „richtiger“ Service wird aus dem Service-Kandidaten dann durch Design und Implementierung. Viele Fragestellungen – insbesondere die nach Granularität und Effizienz des Service Portfolios – können aber bereits auf der Ebene der Service Kandidaten geklärt werden.

Aus den Businessobjekten im BOM kann eine Struktur von Service-Kandidaten abgeleitet werden. Die Idee dahinter ist, dass Businessobjekte die "feinste" Sicht darstellen, die fachlich betrachtet wird (alle weiteren Detaillierungen finden nur noch auf der technischen Ebene statt). Grundsätzlich können auf einem Businessobjekt die folgenden Operationen ausgeführt werden:

1. Create: Anlegen (einer Instanz) des Businessobjekts
2. Read: Lesen (einer Instanz) des Businessobjekts
3. Update: Ändern (einer Instanz) des Businessobjekts
4. Delete: Löschen (einer Instanz) des Businessobjekts

„CRUD“ ist eine Faustregel, die pro Businessobjekt sinnvolle Service-Kandidaten bzw. Operationen derselben definiert. Bei ihrer Anwendung ist zu beachten:

- C, R, U, D, dürfen nicht wörtlich interpretiert werden. Statt nur „Lesen“ ist oft auch „Suchen“ und „Lesen“ bzw. „Listen“ sinnvoll. Auch „Delete“ muss nicht immer physikalisches Löschen heißen, sondern wird je nach fachlichem Zusammenhang besser „Stornieren“ oder „Deaktivieren“ genannt.
- Die CRUD-Regel impliziert keinen Automatismus. Die vier Servicekandidaten (Operationen) sind nur dann sinnvoll, wenn es auch entsprechende fachliche Anforderungen für die Funktionalität gibt. Die Erfahrung zeigt aber, dass primitive Schreib- und Leseoperationen sehr oft benötigt werden.

Das BOM hilft allerdings nur dabei, feingranulare („primitive“) Servicekandidaten zu identifizieren. Im früher verwendeten Beispiel also vielleicht „KaufvertragAn-

legen“ oder „KaufverträgeSuchen“. Komplexere Servicekandidaten ergeben sich dagegen aus formulierten fachlichen Anforderungen oder aus dem Prozessmodell (Beispiel: „BonitätFürKaufvertragPrüfen“).

In der Praxis kann oft ein überraschend großer Anteil der benötigten Funktionen durch einfache „CRUD“-Services abgedeckt werden; höhere Aufgaben - etwa inhaltliche Prüfungen - können in den einfachen Services zum Anlegen und Ändern von Businessobjekten "versteckt" werden.

3.4 Servicedesign

Nachdem Servicekandidaten identifiziert wurden, ist folgendes bekannt:

- Bezeichnung des Service-Kandidaten
- Kurze Beschreibung der Funktionalität
- Eingehende Businessobjekte (Informationen im „Request“)
- Ausgehende Businessobjekte (Informationen im „Response“)

Die Aufgabe im (fachlichen) Servicedesign ist es, daraus eine eindeutige und für die Implementierung taugliche Beschreibung der Funktionalität und des Interfaces zu machen. Das technische Servicedesign bzw. die Servicespezifikation – nicht Gegenstand der Ausführungen hier – spezifiziert dann die technischen Details der Implementierung.

Das primäre Werkzeug für das Servicedesign ist das oben bereits erwähnte detaillierte Modell, das auf dem BOM aufbaut. Es ist durch die vollständige Ausmodellierung aller (fachlichen) Attribute und der Relationen zwischen den Objekten charakterisiert. Außerdem können Businessobjekte zerlegt, hierarchisch strukturiert oder durch Hilfsobjekte ergänzt werden. So enthält es etwa die Ausprägungen derselben Businessobjekte zum Schreiben, Ändern und zum Lesen, die sich etwa durch verschiedene Kardinalitäten ihrer Attribute unterscheiden. Das Service-Interface, das durch die Nachrichten Request, Response und ggf. auch Fault gekennzeichnet ist, lässt sich so durch Verwendung von Objekten aus dem Detailmodell darstellen.

Es ist eine gute Praxis, ein- und dasselbe Modellierungstool für alle Informationsmodelle zu verwenden. Gut geeignet sind etwa UML-Werkzeuge wie der Enterprise Architect (Sparx Systems); Rational RSA/RSM oder einfach das im Unternehmen jeweils beliebte Tool. Wichtig ist die Möglichkeit, Transformationen selbst

erstellen oder modifizieren zu können. Die Services bzw. Serviceoperationen können in einem solchen Tool durch UML-Interfaces modelliert werden, deren Parameter sich auf Objekte im Detailmodell beziehen. Die für die Modellierung notwendigen Artefakte wie XML Schema- und WSDL-Dokumente (oder sogar plattformspezifischer Code für die Implementierung von Services) können dann durch Transformationen aus den Modellen generiert werden.

4 Die Rolle des Architekten:

4.1 Service Portfolio Management

Die beschriebenen Verfahren zur Business Service Modellierung sind nicht als einmalige Tätigkeit zu verstehen. SOA ist nicht als Ansatz für die Architektur isolierter bzw. einmaliger Lösungen zu sehen (wäre dafür wohl zu komplex), sondern dient zur dauerhaften und übergreifenden Umstellung der Art und Weise, wie IT-Lösungen im Unternehmen entwickelt werden. Ein ganz wichtiger Aspekt dabei ist das aktive Management des Service Portfolios - einem zentralen Bestandteil der SOA-Governance.

Serviceportfolio Management bedeutet, langfristig die Entwicklung der Serviceportfolios gezielt zu steuern und dessen Effizienz aufrechtzuerhalten. Die Ziele des Portfolio Managements sind:

- Sicherstellung der Umsetzung neuer Anforderungen
- Sicherstellung der Nutzung vorhandener Services
- Anstoß zu Weiterentwicklung vorhandener Services
- Ausbau des Serviceportfolios durch neue Services
- Vermeidung von Redundanz
- Identifikation nicht mehr benötigter Services und gesteuerte Außerdienststellung

Es handelt sich also beim Service Portfolio Management im Wesentlichen um die fortlaufende Anwendung der beschriebenen Vorgehensweise auf die jeweils neuen Anforderungen. Konkret dienen die beschriebenen Modelle dazu, die Übersicht über das jeweilige Serviceportfolio zu behalten und Planungen durchführen zu können. Eine gerade für Managementpräsentationen beliebte Darstellung ist etwa ein Overlay von Domänenmodell, Businessobjektmodell und den Servicekandidaten. Durch Farben kann zwischen

realisierter, projektiertes und geplanter Implementierung unterschieden werden. Ein solcher „Bebauungsplan“ ermöglicht die Beurteilung der Entwicklung des Serviceportfolios mit einem Blick.

4.2 Herausforderungen für Architekten

Die Anforderungen an die Tätigkeit eines Architekten ändern sich durch SOA - das wird anhand des Vorgehens bei der Business Service Modellierung besonders deutlich. So kann man nicht mehr davon ausgehen, dass die Arbeit des Architekten mit wohl definierten fachlichen Anforderungen beginnt und mit der Erstellung eines technischen Designs (bzw. mit dem Akzeptanztest der Fachseiten) endet. Wir haben gesehen, dass die fachlichen Anforderungen direkte Auswirkungen auf die Service Modellierung haben. Der Architekt muss also frühzeitig im Bereich des Requirements Engineering tätig werden, aktiv das Business verstehen und mit den Fachseiten gemeinsam Modelle entwickeln. Die Anforderungen in Richtung Kommunikation, Darstellung komplexer Sachverhalte und Konfliktfähigkeit sind

hier vielfach ausgeprägter als beim immer noch anzutreffenden Typus eines Architekten, der sich nur in der IT-Welt bewegt.

Man darf aber nicht den Eindruck bekommen, dass ein Architekt in der Welt der SOA nur noch auf der Ebene fachlicher Modelle arbeiten müsste. Im Gegenteil, aus den fachlichen Modellen müssen ja schrittweise immer detailliertere Modelle werden, aus denen schließlich direkt technisch nutzbare Artefakte (z. B.: WSDL-Dokumente) erzeugt werden können. Die fachliche Verständlichkeit darf also nicht auf Kosten der technischen Exaktheit gehen, beides muss zusammen verwirklicht werden. Der Architekt der Zukunft bewegt sich sowohl auf fachlichem als auf technischem Grund sicher.

Vor allem von Softwareherstellern wird manchmal die Ansicht vertreten, dass in Zukunft dank der immer besseren Tools die fachlichen Experten Modelle (z. B.: Prozessmodelle) erstellen können, die dann mehr oder weniger automatisiert zu ausführbaren Applikationen werden. Betrachtet man die Komplexität der Business Service Modellierung, so erscheint diese

Aussage klar als das, was sie ist: als wenig glaubwürdiger Werbeslogan. Der Autor ist sich im Gegenteil mit vielen Praktikern einig, dass viel eher die IT-Architekten fachliches Territorium betreten werden, als dass das Umgekehrte der Fall sein wird. Modellierung erfordert solides technisches Wissen und Erfahrung, die nicht einfach angelernt und schon gar nicht durch Tools ersetzt werden können. Eine Herausforderung für Arbeitgeber und Trainingsanbieter ist es aber sicherlich, Architekten in die Lage zu versetzen, die neuen Aspekte ihrer Aufgabe in Zukunft erfolgreich wahrzunehmen. ■

References

- [1] Hype Cycle for Application Infrastructure Middleware, Platforms and Architecture, 2007 Paolo Malinverno et al. Gartner Group, 2007
- [2] SOA-Mythen im Reality Check Kerstin Kaiser, Dieter Masak Computerwoche online, 6.6.2007