

**Ausarbeitung im Seminar
Agile Workflows und selbstorganisierende Systeme**

SOA Governance

24. Februar 2010

Markus Alexander Kuppe
8kuppe@informatik.uni-hamburg.de
2. Fachsemester M.Sc. Informatik
Matr.-Nr. 6095945

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	1
2.1	Kernkonzepte einer SOA	1
2.1.1	SOA Dreieck	2
2.1.2	SOA Vision - Verschiedene Dimensionen einer SOA	2
3	SOA Governance	6
3.1	SOA Governance Modelle	7
3.1.1	SOA und Service Lifecycle	8
3.1.2	Rollen und Gruppen in der SOA Governance	11
3.1.3	Reifegradmodelle	12
3.1.4	Richtlinien und Leitfäden	14
4	SOA Governance Architektur	14
4.1	Policy Framework	15
4.1.1	IETF Policy Architecture Meta Model	16
4.2	Anforderungen an Policy Frameworks	17
5	Zusammenfassung und Ausblick	18

1 Einleitung

Governance-Modelle spielen auf allen Ebenen eines Unternehmens eine immer stärker werdende Rolle. Sie dienen dabei sowohl der Optimierung der organisatorischen Abläufe als auch der optimalen Ausnutzung technischer (IT) Mittel. Die Autoren Weill und Ross (2004) haben gezeigt, dass Unternehmen mit einer funktionierenden (IT) Governance größere Gewinne erzielen, als Konkurrenten ohne (IT) Governance. Unter der Annahme, dass diese Erkenntnis nicht nur für IT Governance gilt, sondern sich auch auf die SOA Governance übertragen lässt, sollte die Einführung eines SOA Governance Modells Hand in Hand mit der Einführung einer SOA Initiative selbst gehen. Daher dient diese Seminararbeit der Vorstellung der Begriffe SOA und SOA Governance, sowie der konzeptionellen Definition von SOA Governance Modellen. Weiterhin wird der Nutzen der SOA Governance als Meta-Architektur für die SOA typische verteilte Entwicklung dargelegt.

2 Grundlagen

Bevor in Abschnitt 3 SOA Governance Modelle vorgestellt und analysiert werden, sollen im Folgenden die Kernkonzepte der SOA knapp erläutert werden, um die Grundlage für die aus einer SOA resultierenden Herausforderungen erkennbar zu machen. Anschließend diskutiert diese Ausarbeitung mögliche Definitionen für den Begriff SOA Governance und grenzt ihn bezüglich seiner Eigenschaften von bewährten Konzepten wie Corporate und IT Governance ab.

2.1 Kernkonzepte einer SOA

Service-orientierte Architekturen (SOA) sind ein Software Architektur Paradigma (bei (Melzer, 2007, S. 2) aber auch als Programmierkonzept im Großen bezeichnet), das in Organisationen zum Einsatz kommt um *Geschäftsprozesse* mit IT-Mitteln zu realisieren. Vor dem Hintergrund moderner, stark IT-abhängiger Unternehmen, bietet eine SOA die nötige Flexibilität und Agilität, um auf die aus den Geschäftszielen stetig wechselnden (und wachsenden) Anforderungen zu reagieren, sie bestmöglich zu unterstützen und teilweise selbst als Unternehmensmotor aktiv zu werden. Eine SOA ist ein kontinuierlicher Ansatz, dessen Erfolge sich nach mehreren Iterationen einstellen.

Technisches Kernstück einer SOA ist dabei die Repräsentation von Methoden oder (Alt-) Anwendungen - IT-Mittel - als *Dienste*, die offene Schnittstellen anbieten und Standardisierungen befolgen. Aber auch Mitarbeiter des Unternehmens stellen Dienste in einer SOA dar. Ziele sind die allgemeine Nutzbarkeit und Wiederverwendbarkeit in verteilten

Anwendungen (vgl. Geschäftsprozesse), um eine optimale Ausnutzung von Ressourcen zu erreichen. Weiterhin werden die Dienste einer SOA nicht zentralistisch entwickelt. Die Entwicklung erfolgt, ebenso wie die Anwendungen in einer SOA selbst, dezentral. Diese vorgenannten Punkte stellen eine konsequente Weiterführung der Ziele des Komponentenbasierten Software-Engineerings dar (vgl. (Heineman, 2001)). Die Dienstgranularität in einer SOA richtet sich grundsätzlich nach funktionalen Anforderungen aus. Das bedeutet, ein Dienst erfüllt genau eine funktionale Anforderung. Dies trägt weiterhin zur allgemeinen Nutzbarkeit und Wiederverwendbarkeit bei.

2.1.1 SOA Dreieck

Als grundsätzliche Akteure in einer SOA sind *Dienstanbieter*, *Dienstnutzer* und *Dienstmakler* zu nennen, deren Beziehungen zueinander mit dem “magischen” SOA Dreieck abgebildet werden (siehe Abbildung 1). Sie stellen Dienste bereit, nutzen Dienste und stellen die Beziehung zwischen Dienstanbieter und Dienstnehmer her. Dabei kann ein Dienstnutzer verschiedene Dienste zu einem höherwertigen Dienst zusammenfügen und diesen neu komponierten Dienst selbst wieder als Dienstanbieter beim Dienstmakler anbieten. Die Dienstsuche und Bindung zwischen Dienst und Dienstnutzer erfolgt dynamisch zur Laufzeit. Der Dienstmakler stellt dazu einen Verzeichnisdienst (Registry) bereit, in dem Dienstbeschreibungen von Dienstanbietern abgelegt werden. Somit entstehen (gewollte) Dienst-Redundanzen in der Registry, aus denen Dienstnutzer zur Laufzeit - anhand von Metainformationen - eine Auswahl treffen können. Diese Rollenverteilung kann entweder rein unternehmensintern erfolgen oder sich über verschiedene Unternehmen erstrecken.

(Melzer, 2007) fasst diese oben genannten Punkte unter den Stichworten *Verteilung*, *Lose Kopplung*, *Verzeichnisdienste* und *Prozessorientierung* zusammen und liefert als SOA Definition “[...] eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachunabhängige Nutzung und Wiederverwendung ermöglicht”.

2.1.2 SOA Vision - Verschiedene Dimensionen einer SOA

Die in (Melzer, 2007) genannte SOA-Definition geht im Kern jedoch nicht weit genug um die ursprüngliche SOA-Vision zu erfassen, nach der der Hauptnutzen in der dezentralen Verwendung der Dienste über Anwendungs- und Unternehmensgrenzen liegt. Demnach wird eine SOA zwar teils unternehmensintern zur reinen Entwicklung von Verteilten An-

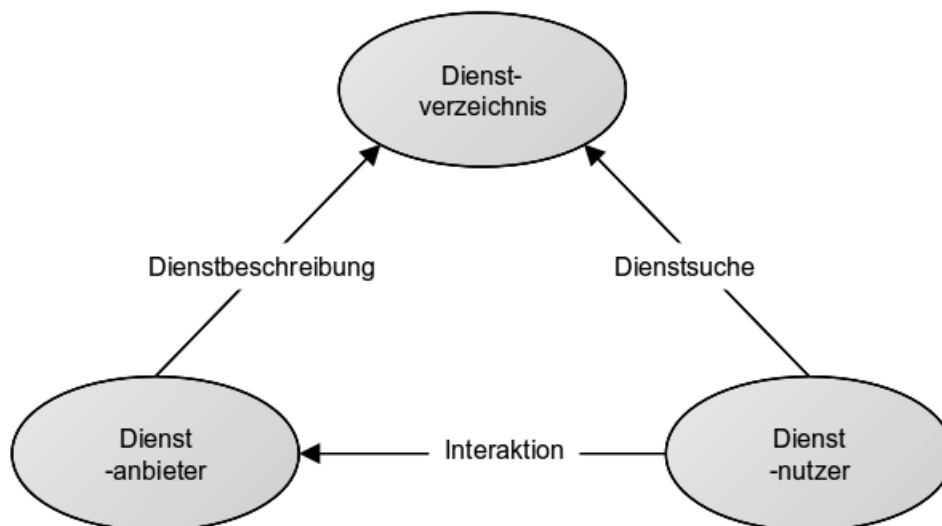


Abbildung 1: SOA Dreieck (Champion u. a., 2002)

wendungen bis hin zu Enterprise Architektur Integration (EAI) eingesetzt (Verteilung über Anwendungsgrenzen), aber ebenso stark auch über Organisationsgrenzen hinweg genutzt. Es entstehen schlussendlich offene und weltweit operierende *Dienstmärkte* in denen Dienste vertrieben werden und sich verschiedene Unternehmen zu virtuellen Organisationen zusammen schließen. Diese verschiedenen Einsatzgebiete einer SOA werden in Abbildung 2 visualisiert.

Diese SOA-Nutzung bereitet, zusammen mit der gesteigerten Agilität gegenüber traditioneller Software-Entwicklung, durch Lose Kopplung in der SOA-Entwicklung (Design time) und im SOA-Betrieb (Runtime) eine Reihe von Herausforderungen (vgl. Niemann u. a. (2009); Schepers u. a. (2008, S. 602)). Schepers u. a. (2008) kategorisiert sie in fünf Bereichen:

- *Einhaltung von Unternehmens-Standards und rechtlichen Rahmenbedingungen*: Die globale Nutzung von Diensten erfordert eine Befolgung von zuvor aufgestellten Richtlinien (Policies) und Standards um einen sachgemäßen Betrieb und allgemeine Nutzbarkeit zu ermöglichen. Ferner müssen verschiedenen (ggf. widersprüchlichen) Rechtsprechungen nachgekommen werden, abhängig davon in welcher Umgebung (Kontext) ein Dienst zu diesem Zeitpunkt genutzt wird.
- *Schwierigkeiten bei der Budgetierung*: Durch die unternehmensweite oder organisationsübergreifende Verwendung von Diensten entstehen Schwierigkeiten bei der Zu-

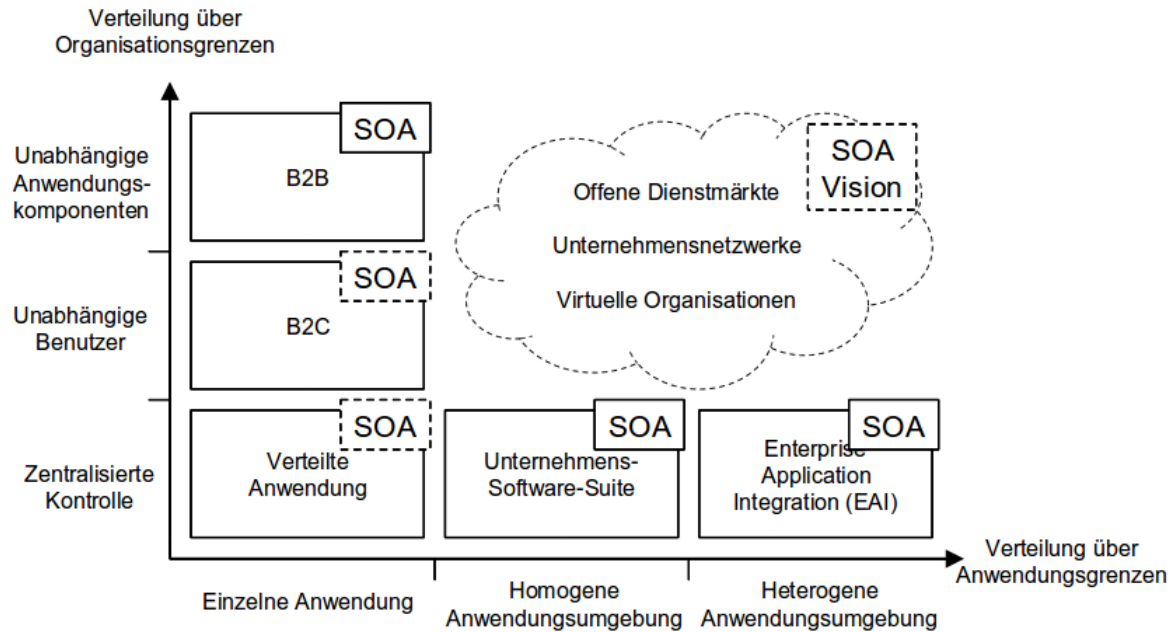


Abbildung 2: Verschiedene Einsatz-Szenarien einer Service-orientierten Architektur nach Braubach u. a. (2009)

weisung von entstehenden Kosten auf die jeweiligen Kostenverursacher. Es müssen deshalb Bezahlmodelle eingeführt werden, die sich im Gegensatz zur traditionellen Software-Entwicklung nicht auf die Erstellung der Dienste berechnen, sondern zur Laufzeit für die Benutzung eines Dienstes anfallen.

- *Änderungen im Lebenszyklus eines Dienstes:* Dienste durchleben einen Lebenszyklus, der von der Entwicklung, über die Einführung und den Betrieb bis hin zur Stilllegung eine Reihe von Phasen umfasst (vgl. Niemann u. a. (2009); Schepers u. a. (2008, S. 602)). Durchläuft ein Dienst einen Wechsel innerhalb dieses Lebenszyklus, sind durch die dynamische Kopplung der Dienste potentiell alle (drei) Rollen betroffen und müssen über den Wechsel informiert werden. Ferner muss eine Risikoabschätzung (nicht nur vom Dienstanbieter) vor jedem Wechsel durchgeführt werden, um mögliche Probleme auszuschließen.
- *Sicherstellung der Dienst-Qualität:* Durch die Abbildung von geschäftskritischen Prozessen auf SOA-Dienste kommt der Verfügbarkeit der Dienste eine entscheidende Bedeutung zu. Ein Ausfall eines Dienste zur Hauptgeschäftszeit führt zwangsweise zu Einnahmeeinbußen. Daher müssen Garantien über die Dienstgüte (Service Level Agreements) zwischen Dienstanbietern und Dienstnutzern geschlossen wer-

2 Grundlagen

den, die beide Seiten zu einer reglementierten Nutzung der Dienste verpflichtet und ggf. Konventionalstrafen verhängt, sofern eine Verletzung der Regel vorliegt. Allgemein muss ein Risiko-Management eingeführt werden, das die Analyse von Risiken als Aufgabe hat.

Neben dieser zur Design time getroffenen Vereinbarung kommt weiterhin die Notwendigkeit zur Überwachung (Monitoring) der Dienste zur Laufzeit, sowie die Umsetzung der eingangs erwähnten Richtlinien, Standards und rechtlichen Rahmenbedingungen.

- *Faktor Mensch*: Die Mitarbeiter eines Unternehmens müssen sich als Dienstleister verstehen und ihre Rollen innerhalb eines Geschäftsprozesses akzeptieren. Dies bedeutet einen Bruch mit den bisher üblichen Zuständigkeiten, die ein Mitarbeiter besaß.

Ergänzend sind das darüber hinausgehende Umdenken in Bezug auf Kompetenzen genannt. Dadurch soll der Wille bei den Mitarbeitern und Entscheidern entstehen, fremde Dienste zu nutzen, die nicht ihrer eigenen Kontrolle unterliegen. Dies ist ersichtlich vor dem Hintergrund, dass die Nutzung fremder Dienste die Einführung von externen Abhängigkeiten bedeutet und somit die Gefahr birgt, eigene Ziele nicht erreichen zu können¹. Hier muss ein Wechsel zu einer holistischen Betrachtungsweise auf allen Ebenen des Unternehmens Einzug halten .

Zusätzlich sehen verschiedene Autoren einige weitere, nicht funktionale Schwierigkeiten in Bezug auf eine SOA:

- *Inkompatibilitäten zwischen der Aufbau- und der Ablauf-Organisationen*: Die traditionelle Funktionsorganisation von Unternehmen (vgl. Abecker u. a. (2002)), in der sich die Geschäftsprozesse an den bestehenden Strukturen orientieren, ist allgemein als unvereinbar mit einer unternehmensweiten SOA angesehen (Melzer, 2007). Daher wird eine Anpassung der Unternehmensstruktur hin zur Prozessorientierung notwendig, in der die Aufbauorganisation an die neuen Anforderung angepasst wird. In dieser neuen Struktur werden klare Verpflichtungen und Zuständigkeiten in Bezug auf die Dienste und sie nutzenden Geschäftsprozesse definiert.

(SOA) Governance wird von Wissenschaft und Wirtschaft als Methode genannt, um diese Herausforderungen systematisch zu adressieren.

¹Wird der fremde Dienste bspw. nicht rechtzeitig geliefert, kann das eigene Projekt unverschuldet verzögert werden.

3 SOA Governance

Die Adressierung bzw. Lösung der im vorherigen Abschnitt illustrierten Probleme werden allgemein als Aufgabe der SOA Governance verstanden. Daher soll in diesem Abschnitt eine Definition für SOA Governance gegeben werden, ihre Aufgaben und Rollen sowie eine Verortung innerhalb der Aufbauorganisation geliefert werden. Inwiefern sich SOA Governance als Konzept von der bereits bekannten IT-Governance unterscheidet und um welchen Belangen sie erweitert werden muss, ist ebenfalls Bestandteil dieses Abschnitts.

Grundsätzlich lässt sich SOA Governance als Meta-Architektur verstehen. Melzer (2007) argumentiert, dass verteilte Systeme zur erfolgreichen Einführung und Betrieb einen zentralen Architekten benötigen. Dies ist vor dem Hintergrund der Dezentralisierung einer SOA in Bezug auf Entwicklung und Laufzeit allerdings nicht umsetzbar. Weder innerhalb einer einzelnen Organisation, noch über Unternehmensgrenzen hinweg. Daher definiert die SOA Governance abstrakte (Architektur-) Vorgaben und implementiert Mechanismen, um deren Einhaltung sicherzustellen. Unter Vorgaben werden Richtlinien, Standards und Leitfäden verstanden. Der Umsetzbarkeit der Vorgaben kommt dabei eine entscheidende Bedeutung zu. Daher werden Messwerte über Metriken erhoben, die die SOA quantifizierbar machen. Windley (2006) ergänzt jedoch, dass SOA Governance zum Erfolg Spielräume für Innovation geben und partizipierenden Rollen (Entwicklung, Fachabteilung, ...) kreative Entfaltungsmöglichkeiten lassen muss. Es ist daher zwingend, definierte Ausnahmen von Vorgaben einzubauen.

Zusätzlich wird auf einer zweiten - organisatorischen - Ebene die Lenkung der SOA als Aufgabe verstanden. Hier müssen Zuständigkeiten reglementiert und die Budgetierung vorgenommen werden. Marks und Bell (2006) ordnen der SOA Governance explizit die Budgethoheit zu, was den "Machtanspruch" der SOA Governance verdeutlicht. Als weitere zentrale Aufgabe der SOA Governance ist die Abstimmung sämtlicher Maßnahmen auf die Geschäftsziele des Unternehmens genannt. SOA Governance ist somit immer ein holistischer Ansatz, der von allen Interessengruppen, insbesondere der Führungsebene, mitgetragen werden muss. Auf die Eingliederung der SOA Governance in die Aufbauorganisation wird in Abschnitt ?? genauer eingegangen).

Zusammenfassend ordnen Marks und Bell (2006) der SOA Governance folgende grundsätzliche Aufgabengebiete zu:

- Überwachung und Verwaltung sämtlicher SOA Aktivitäten auf der organisatorischen Ebene
- Erlassung von mess- und durchsetzbaren Richtlinien, Standards und Leitfäden zur

Entwicklung und Betrieb der SOA

- Zuordnung von Verantwortlichkeiten und Befugnissen zu Diensten, insbesondere die Delegation von Budgethoheiten
- Steuerung des SOA-Lebenszyklus

3.1 SOA Governance Modelle

Unter einem *SOA Governance Modell* wird das Vorgehensmodell verstanden, dass zur SOA Governance eingesetzt wird. (Niemann u. a., 2009) liefern ihr ihrer Arbeit ein Vergleich von elf Vorgehensmodellen anhand von zehn charakterisierenden Merkmalen.

Die Merkmale dieser Modelle sind:

- Adressierung von organisatorischen Herausforderungen (*organizational challenges*) und Veränderungen in Bezug auf die Arbeitsweisen (*impact on behavior*) sowie Vorgabe einer SOA Strategie (*SOA Roadmap*) (vgl. Schepers u. a. (2008) Kategorisierung in Abschnitt 2.1.2)
- Schaffung von spezifischen Rollen sowie Zuweisung von Zuständig- und Verantwortlichkeiten (*Roles/Accountability*) (vgl. Abschnitt 3.1.2)
- Einführung der SOA Governance anhand von Reifegradmodellen (*SOA maturity modell*) unter Nutzung von SOA spezifischen Metriken (*metric models*) (siehe Abschnitt 3.1.3)
- Lieferung von Richtlinien und Leitfäden (*Policy Catalog, Best practices*)^{3.1.4}
- Vorgabe von SOA- und Service-Lebenszyklen (*SOA-/Service-Lifecycle*) (Abschnitt 3.1.1)

Das von (Marks und Bell, 2006) genannte SOA Governance Modell deckt die meisten Merkmale ab und soll daher für die noch ausstehenden Punkte im Folgenden näher erläutert werden. Jedoch referenzieren die Autoren in ihrem Modell keine Reifegradmodelle, weshalb in dieser Arbeit stellvertretend das Modell der Open Group vorgestellt wird (vgl. 3.1.3). Ferner geben (Marks und Bell, 2006) nur grundsätzliche Anmerkungen zur SOA Strategie und Service-Lebenszyklen. Daher wird der durch Schepers u. a. (2008) vorgeschlagene Lebenszyklus dargestellt.

3.1.1 SOA und Service Lifecycle

Wie in Abschnitt 2.1.2 bereits kurz eingeführt, gliedern verschiedene Autoren die Entwicklung und den Betrieb einer SOA in verschiedene Phasen oder Lebenszyklen (Lifecycles). Übereinstimmend unterteilen alle Autoren den SOA Lifecycle in eine Design- und in eine Laufzeit-Phase. Diese grundsätzliche Separierung wird sowohl von Marks und Bell (2006), Schepers u. a. (2008) und Niemann u. a. (2009) als auch von Gu und Lago (2007) getragen. Marks und Bell (2006) erweitern diese Aufteilung um eine Deployment-Phase, die zwischen Design- und Laufzeit angesiedelt ist. Niemann u. a. (2009) sehen im Anschluss an die Laufzeit-Phase grundsätzlich noch eine Änderungsphase. Sie argumentieren aber selber, dass diese Phase durch den iterativen Ansatz des gesamten Lifecycles nicht zwingend notwendig ist.

Je nach Autorengruppe werden unter den beiden Hauptphasen entweder eine technische oder eine umfassende - organisatorische - Sichtweise verstanden. Niemann u. a. (2009) hebt hierzu hervor, dass der SOA Lebenszyklus nicht mit dem reinen Dienst Lebenszyklus gleichzusetzen ist. Diese Ansicht wird von Schepers u. a. (2008) gestützt, die den Dienstzyklus als Teil des SOA Lifecycle sehen. Andere Autoren definieren hingegen lediglich einen Service Lifecycle. Niemann u. a. (2009, S. 601) bezeichnen diese Sichtweise als "the fairly narrow view" und schreiben sie hauptsächlich den kommerziellen Werkzeugherstellern zu, deren SOA Governance Perspektiven auf ihrem Werkzeug-Portfolio basieren. Im Kontext dieser Arbeit wird daher auf den von Schepers u. a. (2008) vorgestellten und wissenschaftlich motivierten SOA Lifecycle zurückgegriffen, der an geeigneten Stellen um Konzepte aus Niemann u. a. (2009) und Gu und Lago (2007) erweitert werden soll. Demnach unterteilt sich ein SOA Lifecycle in sechs aufeinander folgende Phasen:

1. *Aufstellung einer SOA Strategie:* In diese Phase fällt die langfristige strategische Ausrichtung der (unternehmensweiten!) SOA Bestrebung mit den Anforderungen, Zielen und der Vision des Kerngeschäfts. Die hauptsächlichsten Probleme in dieser ersten Phase der SOA Initiative sehen die Autoren in den zusätzlichen Kosten, die nicht sofort durch offensichtliche Vorteile ausgewogen werden. Daher ist eine frühzeitige Einbindung aller Entscheider in die SOA Initiative erforderlich, um eine breite Basis für die Initiative zu sichern. Auf der Gegenseite kann diese breite Basis auch zu Unverträglichkeiten führen und damit die Initiative behindern und verzögern. Hier muss ein ausgewogenes Gleichgewicht herbeigeführt werden. Durch welche Maßnahmen sich diese Ausgewogenheit einstellt, wird von den Autoren nicht erwähnt.
2. *Vorbereitung der Aufbauorganisation auf die SOA Initiative:* Nachdem in der ersten

3 SOA Governance

Phase die strategische Ausrichtung festgeschrieben wurde, müssen in der zweiten Phase Aufgaben abgeleitet und Verantwortlichen zugeordnet werden. Die Aufgaben umfassen sowohl die Schaffung der technischen Grundlagen, als auch den Umbau der Aufbauorganisation des Unternehmens hin zur Prozessorientierung.

3. *Definition Service Portfolio:* Zur Umsetzung der Geschäftsprozesse des Unternehmens wird in der vierten Phase das Service Portfolio definiert. Dazu können in einem Top-Down Ansatz ausgehend von den Geschäftsprozessen die benötigte Dienste abgeleitet werden, die dann zur Implementierung bzw. zum Einkauf vorgesehen werden. Alternativ kann aber auch in einem Bottom-Up Ansatz auf der Basis der bereits existierenden Altanwendungen Dienstkandidaten identifiziert werden und in die zu realisierenden Geschäftsprozesse integriert werden. Die Vorteile eines Top-Down Vorgehensmodell werden allgemein bei der Freiheit in der Modellierung gesehen. Nachteilig ist, dass eventuell existierende Fähigkeiten des Unternehmens nicht wiederverwendet werden.
4. *Verwaltung Service Lifecycle:* Im Anschluss an die Definition des Service Portfolios kann mit der konkreten Umsetzung (Entwicklung und Inbetriebnahme) der Dienste und der darauf basierenden Geschäftsprozesse begonnen werden.
5. *Umsetzung der SOA Vorgaben:* Wie bereits vielfach erwähnt (siehe Abschnitt 3), umfasst SOA Governance Richtlinien, Standards und Leitfäden für die Entwicklung und den Betrieb einer SOA. Schepers u. a. (2008) halten die fünfte Phase des SOA Lebenszyklus als den Zeitpunkt zur Umsetzung dieser Vorgaben. Dabei unterscheiden sie wiederum zwischen einer Umsetzung zur Design-Time und einer Umsetzung zur Laufzeit. An dieser Stelle sei auf Kapitel 4 verwiesen, dass insbesondere die automatische Umsetzung der SOA Vorgaben sowohl zur Lauf- als auch Entwicklungszeit thematisiert.
6. *Verwaltung der Service Levels:* Neben den vorherigen Vorgaben sind ebenfalls Service Level Agreements Bestandteil einer SOA. Die Vereinbarung solcher SLAs zwischen den (abstrakten) Rollen der Dienstanbieter und Dienstanutzer (vgl. 1) sind Gegenstand dieses Lifecycle-Schritts. Weiterhin wird die Erhebung von Messwerten durch Metriken dieser Phase zugeordnet, die die SOA quantifizierbar machen.

Schepers u. a. (2008) nennen im vierten Punkt das Deployment der Services in der Service Registry als auch die Spezifikation der Service Granularität als besonders charakterisierend. Dies sei als Indiz gewertet, dass dieser Punkt eine software-technische Perspektive einnimmt (vgl. Deployment-Phase nach Marks und Bell (2006)). Daher sei stellvertretend

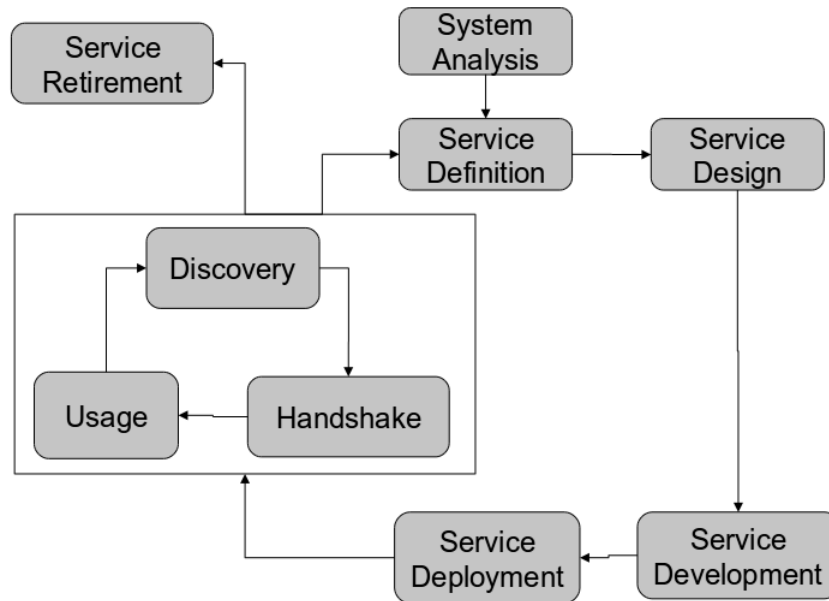


Abbildung 3: SOA Lifecycle nach Niemann u. a. 2009

für die vierte Phase das konkretere Service Lifecycle Modell von Niemann u. a. (2009) vorgestellt (siehe Abbildung 3). Aber auch Teile der fünften und sechsten Phase werden durch dieses Modell abgebildet. Insgesamt lässt sich konstatieren, dass die letzten drei Phasen nicht strikt voneinander abhängig sind und sich zur Parallelisierung eignen.

Niemann u. a. (2009) gliedern ihr Service Lifecycle Modell in sieben Phasen. Die Phasen stellen eine software-technische Sicht auf den Lebenszyklus eines Services dar, lassen sich aber ebenfalls in eine Designphase und eine Laufzeitphase unterteilen. Zur Designphase - alle Arbeitsschritte vor Inbetriebnahme des Dienstes - lassen sich die Phasen *System Analysis*, *Service Definition*, *Service Design* und *Service Development* zusammenfassen. Diese Phasen umfassen alle Arbeiten von der Anforderungsanalyse bis hin zur software-technischen Umsetzung des Dienstes. Mit der *Service Deployment* Phase wird der Dienst in den Regelbetrieb überführt. Der Regelbetrieb setzt sich aus den bereits eingangs erwähnten *Discovery* (Dienstsuche), *Handshake* (Dienstbindung) und *Usage* (Dienstnutzung) zusammen. Da es sich bei diesem Service Lifecycle Modell um einen iterativen Prozess handelt, koppelt die Dienstnutzungs- an die zweite Phase der Designphase (*Service Definition*) an. Die Autoren heben dies als entscheidende Eigenschaft ihres Modells hervor, da dadurch eine gesonderte *Change-Time* Phase, wie zum Beispiel von Marks und Bell (2006) gefordert, entfallen kann. Gibt es Gründe für die Einstellung des Dienstes, wird der Dienst im Anschluss an den Regelbetrieb außer Dienst gestellt (*Service Retirement*).

ment). Ursachen sind hierfür, dass ein Dienst die Anforderungen nicht mehr erfüllt, oder keine Dienstanutzer mehr existieren.

3.1.2 Rollen und Gruppen in der SOA Governance

Verschiedene Autoren äußern sich in ihren Arbeiten zu den nötigen Erweiterungen von bestehenden Rollen bzw. der Neuschaffung SOA-spezifischer Rollen bei der Durchführung einer SOA Initiative. Dabei ist kein einheitliches Modell für Rollen und Ausschüsse erkennbar, das von allen Autoren gefordert wird. Brown u. a. (2008); Marks und Bell (2006); Mitra (2005) nennen jedoch die Einrichtung eines *Center of Excellence (CoE)* als zentrales Instrument bei der organisatorischen Strukturierung der SOA Governance. Dem CoE kommt dabei die Aufgabe zu, als Instanz zwischen den Geschäftsbereichen und der IT zu vermitteln. Die wird durch die paritätische Besetzung durch Mitarbeiter aus beiden Bereichen sichergestellt. Dadurch ist der Dialog das Mittel, mit dem Entscheidungen im CoE getroffen werden. Dies impliziert die strategische Wichtigkeit der IT, die als gleichberechtigter Partner mit dem Geschäftsbereich arbeitet und nicht mehr klassisch als reiner Dienstleister angesehen wird.

Eine weitere Aufgabe des CoE sehen Brown u. a. (2008) bei der grundsätzlichen Vorbereitung des Unternehmens auf die gesamte SOA-Initiative. Dem CoE kommt die Rolle eines verfassungsgebenden Gremiums zu, welches das konkrete SOA Governance Modell auswählt und die dazu passenden Entscheidungsstrukturen etabliert. Außerdem fallen dem CoE die in 1 genannte Einbindung aller Entscheider aus den Geschäftsbereichen (vgl. Prozessverantwortliche und Dienstanutzer) und der IT-Abteilung (vgl. Dienstleister) zu, um frühzeitig eine breite Unterstützung für die SOA-Initiative zu finden. In dieser frühen Phase ist das CoE teilweise mit externen Beratern besetzt, die das nötige SOA-Fachwissen bereitstellen.

Marks und Bell (2006) gliedern in ihrem Modell das CoE in mehrere Gremien und verteilen die Verantwortlichkeiten und Befugnisse dadurch stärker. Abbildung 4 gibt eine grafische Darstellung wieder². Das SOA Core Team, das SOA Architecture Review Board (ARB) und das SOA Process Review Board (PRB) übernehmen hier die Aufgaben des CoE. Dem Core Team fällt dabei in erster Linie die Einführung des SOA Governance Modells und die initiale Leitung des Architecture Review Boards zu. Ist die SOA-Initiative im Regelbetrieb angelangt, löst sich das Core Team auf. Sowohl ARB als auch PRB sind jeweils die für IT-relevante Fragestellungen sowie für fach- bzw. geschäftsbereichsrelevante Probleme zuständige Gremien, die von der jeweilig verantwortlichen Abteilung geleitet

²Die Grafik wurde aus der textuellen Beschreibung rekonstruiert, da die Originalgrafik auf Grund eines Druckfehlers nicht abgedruckt war.

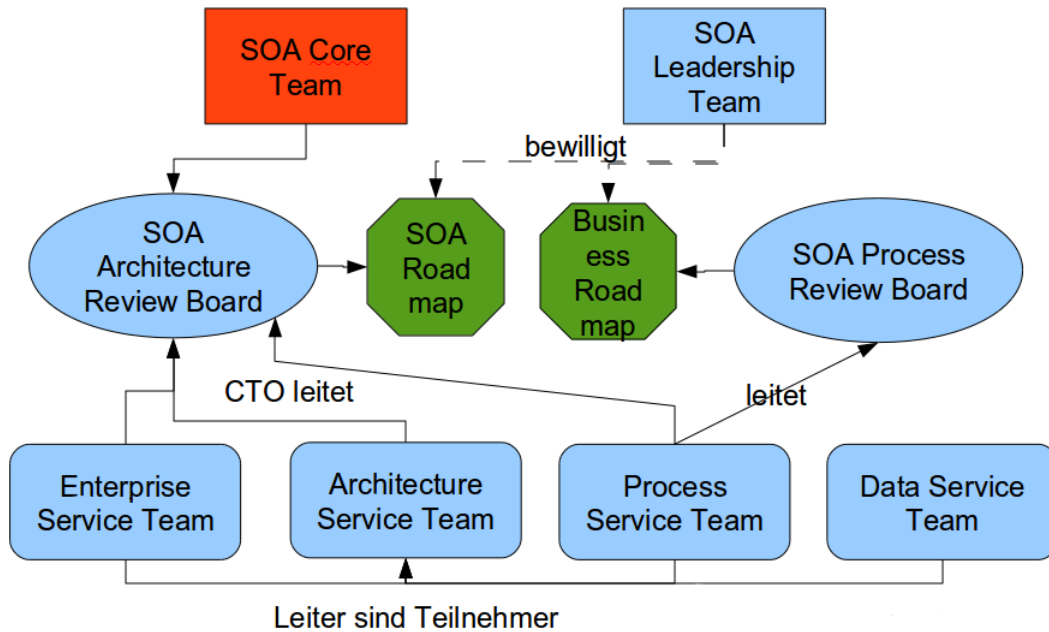


Abbildung 4: Spezifische Rollen und Gremien einer SOA-Initiative (nach Marks und Bell, 2006)

werden. Trotzdem sind beide Gremien miteinander verzahnt, da im ARB Vertreter (Entscheider) aus den Geschäftsbereichen und im PRB Leiter der IT-Abteilung vertreten sind. Als Schiedsinstanz kann das SOA Leadership Team (LB) verstanden werden, wohin Fragestellungen eskaliert werden, die nicht innerhalb der ARB und PRB Ebene beantwortet werden können. Das LB ist mit den Mitgliedern der Geschäftsleitung besetzt. Die unterste Ebene (Enterprise Service Team, Architecture Service Team, Process Service Team und Data Service Team) bildet die operationale Ebene einer SOA-Initiative. Zusammenfassend lässt sich festhalten, dass sämtliche Autoren die Notwendigkeit sehen, die Entscheidungshoheit für übergreifende Aufgaben innerhalb der unternehmensweiten Gremien anzusiedeln. Dies ist im Hinblick auf die Motivationen einer SOA-Initiative konsequent, die die unternehmensweite Wiederverwendung in den Vordergrund stellen (vgl. Abschnitt 2.1.1). Allerdings bekräftigt es auch den (neuen) Stellenwert der IT-Abteilung, die jetzt nicht mehr als reiner Dienstleister für die Geschäftsbereiche fungiert, sondern gleichberechtigt mit den Geschäftsbereichen Entscheidungskompetenzen besitzt.

3.1.3 Reifegradmodelle

Ein weiteres wichtiges Merkmal eines SOA Governance Modells ist die Vorgabe eines Reifegradmodells, anhand dessen die Einführung der SOA Governance im Unternehmen

3 SOA Governance

	Maturity Indicator	Maturity Attributes	Assessment Questions
Level 1 (Silo) Isolated Business Line Driven	Formal definition and documentation of the organization's business drivers and processes.	Low or Nonexistent No formal Enterprise Architecture. Limited to how specific applications must behave, IT specific	What is the business vision and goals and how are these related to what IT is currently doing? Is your current Business Process Architecture formally defined, documented and governed?

Tabelle 1: Reifegrad Indikator für die Geschäfts-Dimension (The Open Group, 2009)

schrittweise vorangetrieben werden kann. Windley (2006) argumentiert hierbei, dass ein "chicken-little" Vorgehen entscheidend ist, da ein schwergewichtiges Governance Modell insbesondere in der Einführungsphase einer SOA Initiative zu Problemen führen kann und die gesamte Initiative gefährdet. Daher sollte das Governance Modell sukzessive erweiterbar sein und sich inkrementell den jeweiligen Anforderungen anpassen lassen.

Verschiedene Reifegradmodelle wurden daher für die Einführung der SOA Governance vorgeschlagen. Schepers u. a. (2008) nennen in diesem Konzept das *Service Integration Maturity Model (SIMM)*, das inzwischen von der OpenGroup (vgl. Wikipedia (2009)) unter der Bezeichnung *Open Group Service Integration Maturity Model (OSIMM)* standardisiert wird und in Version 1.8 als Entwurf vorliegt (The Open Group, 2009). Das OSIMM spannt eine 7x7 Matrix über Reifegrade (*Silo, Integrated, Componentized, Service, Composite Service, Virtualized Service, Dynamically Reconfigurable Service*) und den unternehmensrelevanten Dimensionen (*Business, Organization & Governance, Methods, Applications, Architecture, Information, Infrastructure*) auf. Die in den Zellen erfassten Charakteristika dienen zur (periodischen) Selbsteinschätzung (Assessment) des Ist-Zustands der eigenen SOA Governance Einführung und beschreiben Aufgabengebiete zur weiteren Verbesserung hin zu einem Soll-Zustand. Tabelle 1 stellt die Reifegrad Indikatoren für die erste Ebene (*Silo*) in Bezug zur Dimension *Business* beispielhaft dar.

3.1.4 Richtlinien und Leitfäden

Insgesamt sehen Marks und Bell (2006) sechs verschiedene Kategorien für Governance-relevante Richtlinien und Leitfäden, die sich von abstrakten und organisatorischen Richtlinien zu stark präzisierten technischen Vorgaben erstrecken:

- Unternehmensweite Richtlinien: Organisatorische Richtlinien, die sich auf alle Geschäftsbereiche und Abteilungen beziehen. Als Beispiele sind die Festlegung auf ISO-Standards aber auch ein SOA Governance Modell zu nennen.
- Geschäftsprozess-relevante Richtlinien: Vorgaben, die sich auf einzelne Geschäftsprozesse beziehen, die hauptsächlich in Form von Service Level Agreements zwischen Dienstanbieter und Dienstanutzer ausformuliert sind.
- Vorgehensrichtlinien: Festlegung von wiederkehrenden Arbeitsabläufen innerhalb einer SOA-Initiative wie z.B. Entwicklungsrichtlinien zur Designtime eines Services.
- Gesetzliche Vorgaben: Extern vorgegebene Gesetze, die bei international agierenden Konzernen widersprüchlich sein können.
- Technische Richtlinien: Die Festlegung auf Standards und Werkzeuge auf der technischen Ebene (siehe Standards 2.1.2).
- Sicherheitsrichtlinien: Alle Vorgaben, die die Sicherheit der eingesetzten IT-Mittel sicher stellen, gleichzeitig aber auch die Umsetzung durch gesetzliche Vorgaben garantieren.

Eine Möglichkeit zur automatisierten Umsetzung der Richtlinien ist mit dem Stand der heutigen Mittel nur bei den beiden letztgenannten Kategorien möglich. Dies liegt nicht nur an der niedrigen Abstrahierungsstufe, sondern auch an der vollständig IT-gestützten Durchführung dieser Abläufe.

4 SOA Governance Architektur

Nachdem in den vorigen Abschnitten die organisatorischen Maßnahmen der SOA Governance diskutiert und im letzten Abschnitt die verschiedenen Richtlinien-Kategorien genannt wurden, wird im Folgenden die technische Umsetzbarkeit und automatisierte Durchsetzung von Richtlinien thematisiert.

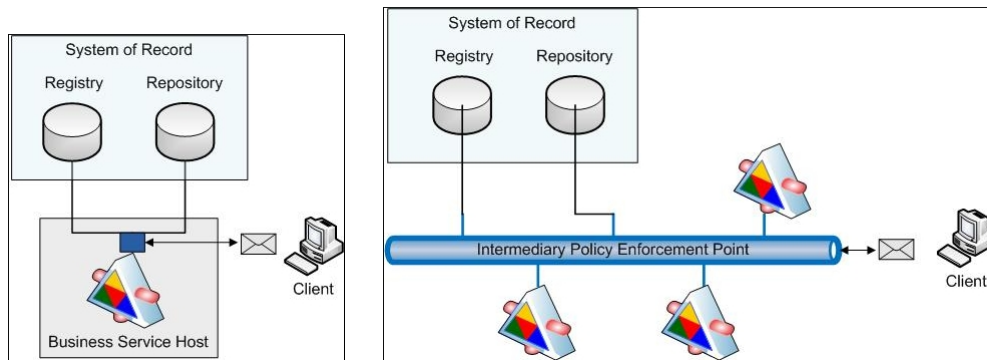


Abbildung 5: Interceptor und Intermediary Policy Enforcement Points (Pijanowski, 2008)

4.1 Policy Framework

Die Grundlage zur automatisierten Richtlinien-Durchsetzung (Policy Enforcement) ist eine maschinenlesbare Definition der Policies - eine Richtlinien-Sprache (Policy-Language). Dazu existieren eine ganze Reihe von (Beschreibungs)-Sprachen, wovon das *WS-Policy Framework* (W3C, 2006) ein weit verbreiteter Vertreter ist. Richtlinien werden darin auf XML Schema (siehe W3C, 2001) basierenden XML Dokumenten abgelegt. Innerhalb eines Dokuments können unterschiedliche Anforderungen anhand von logischen Operatoren miteinander in Verbindung gesetzt werden. So ist es beispielsweise möglich, die Verschlüsselung einer SOAP Nachricht zu erzwingen, dabei allerdings eine Auswahl aus mehreren Verschlüsselungs-Algorithmen zuzulassen. Grundsätzlich ist das WS-Policy Framework jedoch nicht nur auf Sicherheits-Richtlinien ausgerichtet, sondern kann allgemein domänenspezifisch erweitert werden. Bis dato wurden Erweiterungen für Nachrichtensicherheit (WS-Security Policy), verlässliche Kommunikation (WS-Reliable Messaging Policy) oder zur Optimierung der Nachrichtenübertragung von SOAP Nachrichten (SOAP Message Transmission Optimization Mechanism) umgesetzt.

Neben der Policy-Language setzt sich ein Policy Framework aus einer zweiten Komponente zusammen, die die Durchsetzung einer Richtlinie zur Laufzeit regelt. Dazu werden Policy Enforcement Points (PEP) in das Nachrichten-Routing zwischengeschaltet, so dass sie Zugriff auf sämtlich Nachrichten bekommen. Je nach technischer SOA Architektur werden die Nachrichten entweder vor der Zustellung am Service abgefangen und bei einer Richtlinienverletzung verworfen (Interceptor) oder bereits durch Vermittler (Intermediary) auf dem Enterprise Service Bus (ESB) behandelt (siehe Abbildung 5).

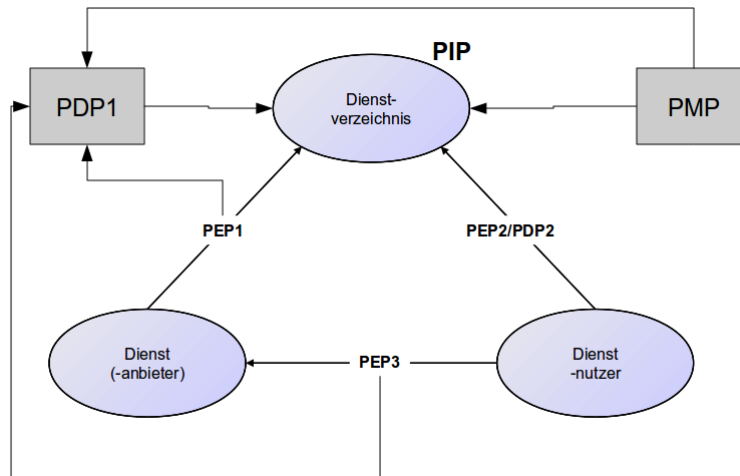


Abbildung 6: IETF Policy Architecture Meta Model (Snir u. a., 2003)

4.1.1 IETF Policy Architecture Meta Model

Die im vorherigen Abschnitt beschriebene Architektur hat die Internet Engineering Task Force (IETF) in einer Hersteller-unabhängigen Meta-Architektur zusammengefasst (Snir u. a., 2003). Neben den *Policy Enforcement Points* (PEP) zur Durchsetzung der Richtlinien werden weitere Komponenten definiert. Der *Policy Management Point* (PMP) dient der Eingabe und Administration der Richtlinien. Weiterhin werden widersprüchliche Richtlinien am PMP durch den Administrator aufgelöst. Der *Policy Information Point* (PIP) stellt die (zentrale) Richtlinien-Datenbank samt Suchfunktionalität dar. Hier bietet sich eine Verbindung mit der SOA Registry an, was sich in verschiedenen Produkten wieder spiegelt und in Abbildung 5 bereits mit *System of Record* bezeichnet wird. Das PIP dient dem *Policy Decision Point* (PDP) als Backend, um Anfragen eines PEP zu beantworten. Dazu ruft das PDP die aktuell gültigen Richtlinien beim PIP ab und trifft auf dieser Grundlage eine Entscheidung, ob der PEP die Nachricht verwerfen oder weiterleiten darf. Kann diese Entscheidung wegen eines Widerspruchs nicht getroffen werden, wird die Anfrage an den PMP delegiert. In dieser Architektur stellt das PIP (und PMP) jedoch einen Single-Point-of-Failure dar. Daher können PDPs Richtlinien zwischenspeichern, um bei einem Ausfall des PIP, weiterhin den PEPs antworten zu können.

Die Integration in das eingangs vorgestellte SOA Dreieck ist in Abbildung 6 wiedergegeben.

4.2 Anforderungen an Policy Frameworks

Die Autorengruppe Phan u. a. (2008) argumentieren, dass insbesondere die Policy-Language des WS-Policy Frameworks für eine umfassende SOA Governance unzulänglich ist. „*WS-Policy is a low-level policy language that is specific to Web Services implementation and is not suitable for managing an overall SOA system*“ (Phan u. a., 2008, S. 9). Davon ausgehend formulieren die Autoren einen Anforderungskatalog mit acht Merkmalen an Policy Frameworks:

- Ein vollständiges Policy Framework unterstützt die *Spezifikation* von Richtlinien mit Policy Languages, hilft bei der *Analyse* von Richtlinien und auftretenden Widersprüchen und bietet außerdem eine *Architektur* zur Durchsetzung der Richtlinien.
- Das dem Policy Framework zugrunde liegende Sprachgerüst soll eine formale Definition besitzen und Erweiterungen in Bezug auf die Sprachkonstrukte zulassen. Insbesondere muss die Sprache sowohl eine Verifikation als auch eine Konflikt-Erkennung/Auflösung bieten. Als Beispiele für eine *formale Sprache* nennen die Autoren die Beschreibungslogik (siehe Baader u. a., 2003).
- Zur vereinfachten Verwaltung soll zumindest eine *Gruppierung* von Richtlinien möglich sein, wenn nicht sogar eine Unterstützung von *Domänen*.
- Mit Hilfe von *Meta-Policies* muss eine Verwaltung von Richtlinien vorhanden sein, die insbesondere durch Prioritäten eine einfache Konflikt-Auflösung zulässt.
- Um Skalierbarkeit, Fehlertoleranz und Flexibilität sicher zu stellen, müssen sämtliche Komponenten der Architektur eine *Verteilung* erlauben.
- Neben den konkreten, technischen (“low-level”) Richtlinien soll das Framework das *Abstraktionsvermögen* besitzen, auch organisatorische (“high-level”) Policies abzubilden. Aus den abstrakten Policies können implementierungs-spezifische Policies generiert werden.
- Das Framework muss eine *Änderungsunterstützung* besitzen, um zum einen den Lebenszyklus eines Services abzubilden, zum anderen auch dynamische Updates zu zulassen.
- Dienst-Kompositionen müssen unterstützt werden und auf dieser Ebene ein logisches Schließen (Reasoning) in Bezug auf Widersprüche durchführen können.

Den Großteil dieser Anforderungen erfüllt KAOs. Dabei handelt es sich um ein generisches Policy Framework, welches jedoch bereits auf eine SOA adaptiert wurde (siehe Uszok u. a., 2004). Als Policy-Language kommt OWL-DL (vgl. W3C, 2004) zum Einsatz. Es definiert eine dem IETF ähnliche Referenzarchitektur und realisiert mit Hilfe von Agenten-Technologie die Verteilung. Phan u. a. (2008) vergleichen KAOs in ihrer Arbeit mit vier weiteren Policy Frameworks und stellen fest, dass KAOs die vollständigste Implementierung eines Policy Frameworks zur SOA Governance ist.

5 Zusammenfassung und Ausblick

In dieser Seminararbeit wurden die Kernkonzepte einer SOA vorgestellt, um den wirtschaftlichen wie technischen Nutzen der SOA Governance zu erläutern und eine Definition von SOA Governance Modellen zu liefern. Auf dieser Basis wurde im Anschluss ein konkretes SOA Governance Modell diskutiert und eingeführt. Die beschriebenen Charakteristika zeigen, dass die vermeintlich technisch gesehene SOA Governance in erster Linie die Mitarbeiter bzw. die organisatorischen Strukturen eines Unternehmens behandelt und erst in zweiter Linie technische Belange adressiert. Trotzdem ist die automatische Durchsetzung von SOA Governance Richtlinien eine Notwendigkeit, die mit einem Anforderungskatalog eingegrenzt und anhand verschiedener Vertreter in dieser Arbeit vorgestellt wurde.

Neben der Einführung der SOA Governance wurde eine Abgrenzung zu anderen Governance Bereichen, insbesondere der IT Governance geliefert. Daraus lässt sich die weitergehende Fragestellung ableiten, inwiefern die SOA Governance, IT Governance oder auch Corporate Governance zur weiteren Optimierung zusammengefasst werden können, um in einer Total Governance alle Aspekte eines Unternehmens zu berücksichtigen.

Literatur

- [Abecker u. a. 2002] ABECKER, Andreas ; HINKELMANN, Knut ; MAUS, Heiko ; MUELLER, Heinz-Juergen: *Geschäftsprozessorientiertes Wissensmanagement - Effektive Wissensnutzung bei der Planung und Umsetzung von Geschäftsprozessen*. Springer, 2002. – 472 S
- [Baader u. a. 2003] BAADER, Franz (Hrsg.) ; CALVANESE, Diego (Hrsg.) ; MCGUINNESS, Deborah L. (Hrsg.) ; NARDI, Daniele (Hrsg.) ; PATEL-SCHNEIDER, Peter F. (Hrsg.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. – ISBN 0-521-78176-0
- [Braubach u. a. 2009] BRAUBACH, Lars ; POKAHR, Alexander ; LAMERSDORF, Winfried: *Service Oriented Architecture (SOA): Grundlagen*. 2009. – URL http://vsis-www.informatik.uni-hamburg.de/getDoc.php/lectures/2222/vl_08_soa_einfuehrung.pdf. – [Online; Stand 17. Dezember 2009]
- [Brown u. a. 2008] BROWN, William A. ; LAIRD, Robert G. ; GEE, Clive ; MITRA, Tilak: *SOA Governance: Achieving and Sustaining Business and IT Agility*. IBM Press, 2008. – ISBN 0137147465, 9780137147465
- [Champion u. a. 2002] CHAMPION, Miachel ; FERRIS, Chris ; NEWCOMER, Eric ; ORCHARD, David: *Web Services Architecture*. 2002. – URL <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>. – [Online; Stand 20. Dezember 2009]
- [Gu und Lago 2007] GU, Qing ; LAGO, Patricia: A stakeholder-driven service life cycle model for SOA. In: *IW-SOSWE '07: 2nd international workshop on Service oriented software engineering*. New York, NY, USA : ACM, 2007, S. 1–7. – ISBN 978-1-59593-723-0
- [Heineman 2001] HEINEMAN, George: *Component-based Software Engineering. Putting the Pieces Together*. Addison-Wesley Longman, 2001. – 416 S
- [Marks und Bell 2006] MARKS, Eric A. ; BELL, Michael: *Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology*. New York, NY, USA : John Wiley & Sons, Inc., 2006. – ISBN 0471768944
- [Melzer 2007] MELZER, Ingo: *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*. 3. Aufl. Spektrum Akademischer Verlag, 2007. – URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/382741993X>. – ISBN 382741993X

- [Mitra 2005] MITRA, Tilak: *A Case for SOA Governance*. IBM developerworks. 2005. – URL <http://www.ibm.com/developerworks/webservices/library/ws-soa-govern/>. – [Online; Stand 17. Dezember 2009]
- [Niemann u. a. 2009] NIEMANN, M. ; JANIESCH, C. ; REPP, N. ; STEINMETZ, R.: Challenges of governance approaches for service-oriented architectures. In: *Digital Ecosystems and Technologies, 2009. DEST '09. 3rd IEEE International Conference on*, June 2009, S. 600–605
- [Phan u. a. 2008] PHAN, Tan ; HAN, Jun ; SCHNEIDER, Jean-Guy ; EBRINGER, Tim ; ROGERS, Tony: A Survey of Policy-Based Management Approaches for Service Oriented Systems. In: *ASWEC '08: Proceedings of the 19th Australian Conference on Software Engineering*. Washington, DC, USA : IEEE Computer Society, 2008, S. 392–401. – ISBN 978-0-7695-3100-7
- [Pijanowski 2008] PIJANOWSKI, Keith: *Automating SOA Governance*. Februar 2008. – URL <http://www.keithpij.com/Home/tabid/36/EntryID/8/Default.aspx>. – [Online; Stand 17. Dezember 2009]
- [Schepers u. a. 2008] SCHEPERS, T. G. J. ; IACOB, M. E. ; VAN ECK, P. A. T.: A lifecycle approach to SOA governance. In: *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*. New York, NY, USA : ACM, 2008, S. 1055–1061. – ISBN 978-1-59593-753-7
- [Snir u. a. 2003] SNIR, Y. ; RAMBERG, Y. ; STRASSNER, J. ; COHEN, R. ; MOORE, B.: *Policy Quality of Service (QoS) Information Model*. 2003
- [The Open Group 2009] THE OPEN GROUP: *OSIMM: The Open Group Service Integration Maturity Model*. November 2009. – URL <http://www.opengroup.org/projects/osimm/>. – [Online; accessed 27-December-2009]
- [Uszok u. a. 2004] USZOK, Andrzej ; BRADSHAW, Jeffrey M. ; JOHNSON, Matthew ; JEFFERS, Renia ; TATE, Austin ; DALTON, Jeff ; AITKEN, Stuart: KAoS Policy Management for Semantic Web Services. In: *IEEE Intelligent Systems* 19 (2004), Nr. 4, S. 32–41. – ISSN 1541-1672
- [W3C 2006] W3C: *Web Services Policy Framework (WSPolicy) 1.2*. March 2006
- [Weill und Ross 2004] WEILL, P ; ROSS, J W.: IT governance on one page. In: *Rep. 4237-04, CIS Research Working Paper No. 349* MIT Sloan School of Management (Veranst.), 2004

5 Zusammenfassung und Ausblick

[Wikipedia 2009] WIKIPEDIA: *The Open Group* — *Wikipedia, The Free Encyclopedia*. 2009. – URL http://en.wikipedia.org/w/index.php?title=The_Open_Group&oldid=327664104. – [Online; accessed 27-December-2009]

[Windley 2006] WINDLEY, Phillip J.: *SOA Governance: Rules of the Game*. August 2006. – URL http://akamai.infoworld.com/pdf/special_report/2006/04SRsoagov.pdf. – [Online; Stand 17. Dezember 2009]

Verweise

[W3C 2001] W3C: *XML Schema*. May 2001. – URL <http://www.w3.org/XML/Schema>. – [Online; accessed 27-December-2009]

[W3C 2004] W3C: *OWL Web Ontology Language Guide*. February 2004. – URL <http://www.w3.org/TR/owl-guide/>. – [Online; accessed 26-January-2010]