UNIVERSITÄT
BERN

Institut für Wirtschaftsinformatik
der Universität Bern

Arbeitsbericht Nr. 212

Compliance-Nachweise bei Anwendung dynamischer Bindung in serviceorientierten Architekturen

Gabriela Loosli

2008-07

Institutsadresse: Engehaldenstrasse 8, CH-3012 Bern, Schweiz
Tel.: +41 (0)31 631 39 12
E-Mail: gabriela.loosli@iwi.unibe.ch

COMPLIANCE-NACHWEISE BEI ANWENDUNG DYNAMISCHER BINDUNG IN SERVICEORIENTIERTEN ARCHITEKTUREN

ABSTRACT

Eine wesentliche Eigenschaft der serviceorientierten Architektur (SOA) ist die mögliche Bindung zur Laufzeit. Diese dynamische Bindung von Services kann verschiedene Ausprägungsformen aufweisen. Bei Anwendung der weitreichendsten Form der dynamischen Bindung besteht die Gefahr, dass aufgrund von Änderungen des Service-Bestands im Repository zur Laufzeit automatisch nicht konforme Services gewählt werden. Nicht konform bedeutet in diesem Zusammenhang, dass nicht alle erforderlichen Regulierungen erkannt und demnach nicht alle erfüllt werden. Bei Wiederverwendung von Services in einem anderen Kontext müssen u. U. zusätzliche Regulierungen erfüllt werden. Bisherige Change-Management-Ansätze, welche auf der Überprüfung aller Services und Applikationen vor Produktivsetzung basieren, lösen dieses Problem nicht. Demnach kann Compliance nicht immer gewährleistet werden. Dieser Beitrag stellt einen Ansatz vor, der mithilfe von semantischen Konzepten die Auswahl nicht konformer Services verhindern soll.

KEYWORDS

Serviceorientierte Architektur (SOA);Compliance; dynamische Bindung; semantische Konzepte.

1 EINLEITUNG

Ein wichtiges Ziel einer serviceorientierten Architektur (SOA) ist die Unterstützung der Agilität eines Unternehmens im Sinne der Flexibilität, Änderungen in den Geschäftsprozessen zeitnah in IT-Systemen umsetzen zu können [Er04, 297; KBS04, 1ff; Ab07, 1; JG07, 189; WS07, 44ff; ZTZ07]. Eine SOA ist eine Software-Architektur, in der Services die fundamentalen Elemente darstellen. Services sind Software-Einheiten verschiedener Granularität, welche zu komplexeren Services bis hin zu Prozessen bzw. Applikationen aggregiert werden können (vgl. z. B. [PG03; Er04, 33f; KBS04, 58ff; Do05, 7]). Die Flexibilität wird erreicht durch die lose Kopplung der Services. Indem eine Beziehung zwischen ihnen erst im Prozessablauf hergestellt wird, können sie innerhalb eines Prozesses besser ausgetauscht oder erst bei Bedarf eingebunden werden. Im SOA-Konzept ist dies sogar zur Laufzeit möglich, da die Services nicht im Quellcode fest verlinkt, sondern über ihre Angaben im Repository gebunden werden. Die Bindung zur Laufzeit wird auch als dynamische Bindung bezeichnet und trägt wesentlich zur losen Kopplung bei [Er04, 297; KBS04, 46ff; Do05, 9; JG07, 190f].

Neben diesem Vorteil ist jedoch zu beachten, dass bei einer dynamischen Auswahl von Services ein besonderes Augenmerk auf die rechtlichen Aspekte gelegt werden muss. Beispielsweise bedeutet eine "zufällige Einbindung eines Dienstes" eine "nicht gesteuerte Veränderung des Gesamtsystems. Somit wäre das System in einem undefinierten Zustand und damit nicht mehr betriebsbereit" [Do05, 261].

In diesem Beitrag wird gezeigt, welche Auswirkungen die verschiedenen Ausprägungsformen der dynamischen Bindung von Services auf die Compliance zu Vorschriften haben und ein Ansatz vorgestellt, wie mithilfe von semantischen Konzepten der Kontext der Servicenutzung berücksichtigt und daraus folgend die anzuwendenden Regulierungen bestimmt werden können. Damit soll die Auswahl nicht konformer Services verhindert werden.

Der Rest des Beitrags ist wie folgt gegliedert: In Abschnitt 2 wird auf verwandte Arbeiten sowie die Abgrenzung zu ihnen eingegangen. Abschnitt 3 definiert den Begriff Compliance. Danach werden in Abschnitt 4 anhand einzelner Beispiele die Problematik der Erfüllung der Compliance in einer SOA bei Anwendung von verschiedenen Ausprägungsformen der dynamischen Bindung erläutert und diesbezüglich Lösungsansätze vorgestellt. Ebenfalls in diesem Abschnitt wird die dynamische Bindung mit ihren Ausprägungsformen definiert. Abschließend gibt Abschnitt 5 eine Zusammenfassung und einen Ausblick.

2 VERWANDTE ARBEITEN

Obwohl Compliance ein bedeutendes Thema ist, besitzt es in der SOA-Literatur keine große Bedeutung. Beiträge dazu behandeln hauptsächlich den Sicherheitsbereich. Dostal et al. widmen den rechtlichen Rahmenbedingungen ein eigenes Teilkapitel [Do05, 252ff] und erwähnen auch die Problematik der dynamischen Bindung diesbezüglich. Jedoch geben sie an, das Thema nicht erschöpfend zu behandeln. So sollen zwar "alle zur Auswahl stehenden Dienste den rechtlichen Anforderungen genügen und ein Wechsel der Dienste im Voraus einkalkuliert und dokumentiert" werden [Do05, 261]; die Prüfung der Prozesse und die Wiederverwendung in einem anderen Kontext werden nicht betrachtet. Ebenso nicht die verschiedenen Formen der dynamischen Bindung. Letzteren wird in der SOA-Literatur allgemein sehr wenig Beachtung geschenkt.

In neueren Beiträgen wird der Begriff *SOA-Governance* erwähnt. Jedoch ist zu beachten, dass dieser Begriff oftmals umfassender betrachtet wird als nur im Hinblick auf die Erbringung der Compliance-Nachweise. Aspekte des klassischen IT-Managements sind darin enthalten. So werden Aufgaben wie beispielsweise "Ausrichten der SOA an den Geschäftszielen" [Ke07, 292; SS07, 69], "Optimieren der IT" [Ke07, 300], "Erfolgreiches Umsetzen der SOA" [Ka07, 330; Ke07, 304; Si07, S111; SS07, 69; Jo08, 325] oder "Treffen von Gestaltungsentscheidungen" [Ab07, 3; Fa07, 314; Ka07, 330; Jo08, 326] ebenfalls miteinbezogen. In diesem Beitrag werden lediglich Compliance-Aufgaben wie "IT-Unterstützung zur Ermittlung korrekter Zahlen und für automatisierte Kontrollen", "Definieren der Vorgehensweisen bei Systemänderungen", "Vermeidung von unkontrolliertem Service-Wildwuchs" oder "Festlegen von Zugriffsregelungen" [KL06, 452; Fa07, 313; Ka07, 330] berücksichtigt. Zur Abgrenzung der Begriffe vgl. [KL06, 451f, 454].

Die dynamische Bindung wird in diesen Beiträgen nicht berücksichtigt. In [JG07, 191] wird die Wiederverwendung in anderen Kontexten erwähnt, mögliche Probleme diesbezüglich werden jedoch nicht betrachtet. Dies im Gegensatz zu [Fo06], welcher die zusätzlich zu erfüllenden Regulierungen berücksichtigt und demnach die Auswirkungen der Wiederverwendung von *bestehenden Services* in *neuen Prozessen*. Was fehlt, ist die umgekehrte Betrachtungsweise, die Auswirkungen von *neu* im Repository verfügbaren *Services* auf *bestehende Prozesse* (vgl. Abschnitt 4.1).

Für die Erbringung der Compliance-Nachweise existieren interessante *Lösungsansätze*, welche die Prozesse, teilweise sogar zur Laufzeit, überprüfen. Einige sind nicht spezifisch auf eine SOA ausgerichtet [Gi05; Ag06; NS07]. Während [LMX07; NS07] Prozessänderungen zur Laufzeit nicht prüfen, konzentrieren sich [Ag06] auf eine einzige Regulierung. In [Og04] werden generische, allen Regulierungen zugrunde liegende, Compliance-Services (z. B. Zugriffskontrollen) vorgeschlagen. Jedoch ist

dabei zu Beachten, dass gerade die teilweise unterschiedlichen Details der verschiedenen Regulierungen für ihre Erfüllung von Bedeutung sind. In [He05] wird mit semantischen Beschreibungen und darauf aufbauend Abfragen wie "List all business processes that depend on system x" u.a. Compliance-Unterstützung geboten. Jedoch genügen Abfragen alleine nicht und jeder (mögliche) Kontextwechsel muss bewusst sein, damit er explizit abgefragt werden kann (vgl. Abschnitt 4.3).

3 COMPLIANCE

Es ist nicht neu, dass Unternehmen Vorschriften erfüllen müssen. Beispielsweise müssen seit langem Rechnungslegungsvorschriften eingehalten werden. Neu ist die starke Zunahme, die weiter reichenden Anforderungen sowie die Internationalität der Vorschriften (vgl. z. B. [Do05, 252f; Kn07, S98f]). Das meist erwähnte Beispiel in diesem Zusammenhang ist der Sarbanes-Oxley Act (SOX). Einen Überblick über die Auswirkungen dieser Regulierung auf Informationssysteme geben [KW06]. Neben den Vorschriften zur Rechnungslegung existieren weitere gesetzliche Vorgaben wie Datenschutzgesetze, branchenspezifische Vorschriften (etwa Basel II, Solvency II, REACH, MiFID), oder interne Regeln, welche beachtet werden müssen. Zu all diesen unterschiedlichen Regulierungen, welche oft wenig präzis formulierte Bestimmungen enthalten und sich häufig ändern, müssen Unternehmen Compliance nachweisen.

Compliance kann definiert werden als Einhalten aller für das entsprechende Unternehmen relevanten gesetzlichen, behördlichen oder aufsichtsrechtlichen und internen Vorschriften sowie die Beachtung von marktüblichen Standards und Standesregeln zur Schaffung höherer Transparenz und Kontrollierbarkeit der Verhaltensweisen eines Unternehmens und der Mitarbeiter (vgl. [Th01, 447; Eb06, 10; JG07, 15]).

Die IT ist davon (neben den Regelungen, die sie direkt erfüllen muss) einerseits durch die Abbildung der Geschäftsprozesse in ihren Systemen und andererseits durch die automatisierte Unterstützung der Prüfung mittels Kontrollen und Berichten (vgl. u.a. [KW06; Kn07]) betroffen.

Compliance wird oft in Verbindung mit Corporate Governance erwähnt. Aus Corporate Governance leitet sich die IT-Governance für den IT-Bereich [KL06, 451; Kn07, S98; SS07, 69] und daraus wiederum die SOA-Governance für serviceorientierte Architekturen ab [Ke07, 289; SS07, 69].

4 COMPLIANCE-NACHWEISE IN EINER SOA

4.1 Mögliche Problemfälle

In einer SOA werden Services in Repositories verzeichnet und gesucht. Es ist demnach wichtig, dass die Services im Repository compliant sind. Ein Service kann von mehreren Prozessen in jeweils verschiedenen Kontexten genutzt werden, indem er *wieder verwendet* wird. Dabei können auch Services von *externen* Anbietern eingesetzt werden. Zunächst stellt sich die Frage, was in einer SOA unter einem compliant Service zu verstehen ist. Dies verdeutlicht folgendes Beispiel eines Gefahrenpotenzials für den SOX:

"The company has a standard sales contract, but sales personnel frequently modify the terms of the contract. Sales personnel frequently grant unauthorized and unrecorded sales discounts to customers without the knowledge of the accounting department. These amounts are deducted by customers in paying their invoices and are recorded as outstanding balances on the accounts receivable aging. Although these amounts are individually insignificant, they are material in the aggregate and have occurred consistently over the past few years" [Pc04, 261].

Bezogen auf eine SOA bedeutet dies beispielsweise, dass es einen separaten "Rabatt-Service" gibt, der nur von autorisierten Mitarbeitern aufgerufen werden kann. Dieser nimmt eine korrekte Verbuchung des Rabatts im System vor und reduziert damit den Rechnungsbetrag entsprechend. Der nachfolgende "Rechnungsstellungs-Service" stellt die Rechnung auf genau den (Rest-) Betrag, der im System vorhanden ist, aus. Die Compliance beim "Rabatt-Service" wird sichergestellt, indem die Zugriffsautorisierungen, die Übereinstimmung mit dem Standard-Verkaufsvertrag sowie die korrekte Verbuchung im System überprüft werden. Beim "Rechnungsstellungs-Service" geschieht dies, indem sichergestellt wird, dass der Rechnungsbetrag aus dem System geholt und nicht manuell eingegeben oder verändert werden kann. Diese Prüfungen können zur Entwicklungszeit stattfinden, sodass die Services erst nach erfolgreicher Prüfung im Repository registriert werden. Das bedeutet, dass dem mit einer organisatorischen Maßnahme, einem Genehmigungsprozess für die Produktivsetzung der Services, begegnet werden kann.

Ein Sonderfall stellt der Bezug von *externen* Services dar. Wird beispielsweise zur Laufzeit der Service-Anbieter gewechselt, muss dieser ebenfalls den Anforderungen der Regulierungen genügen und der Service-Nutzer muss dies sicherstellen. Er muss sich also für Compliance-relevante Aufgaben darum kümmern, wie die Leistung erbracht wird [Do05, 261; Kn07, S101]. Das widerspricht einem wichtigen Merkmal der SOA: Die Trennung der Schnittstelle (*was*, für den Service-Nutzer relevant) von der Implementierung (*wie*, erledigt durch den Service-Anbieter) (vgl. z. B. [Er04, 37]). Auch dieses Problem kann mit einer organisatorischen Maßnahme gelöst werden: Die Einschränkung der zur Auswahl stehenden Services. Beispielsweise dürfen nur Services aus dem unternehmenseigenen Repository verwendet werden. Externe Services werden wie die internen geprüft, bevor sie in dieses Repository aufgenommen werden.

Selbst wenn alle Services im Repository compliant sind, gilt das nicht automatisch für alle Prozesse, welche diese Services verwenden. Nämlich dann, wenn Services in einem zuvor nicht berücksichtigten *Kontext* in einem anderen Prozess *wieder verwendet* werden. Auch diese Situation wird nachfolgend mit einem Beispiel verdeutlicht:

Ein Service, welcher Log-Daten speichert, wird auf Compliance geprüft, bevor er in Produktion geht. Später wird der Service in einem Bestellprozess wieder verwendet, in dem er neben den allgemeinen Bestellinformationen auch Kreditkartendaten speichert. Nun unterliegt derselbe Service beispielsweise zusätzlich den Regeln des Payment Card Industry Data Security Standard (PCI DSS), ohne dass er selber verändert und neu in Produktion gebracht wurde. Das Einzige, was geändert hat, ist die Art der Verwendung, d. h. der Kontext [Fo06]. Die PCI-DSS-Regeln untersagen beispielsweise das Speichern bzw. Aufbewahren von Daten wie volle Magnetstreifen-Information oder PIN [Pc08].

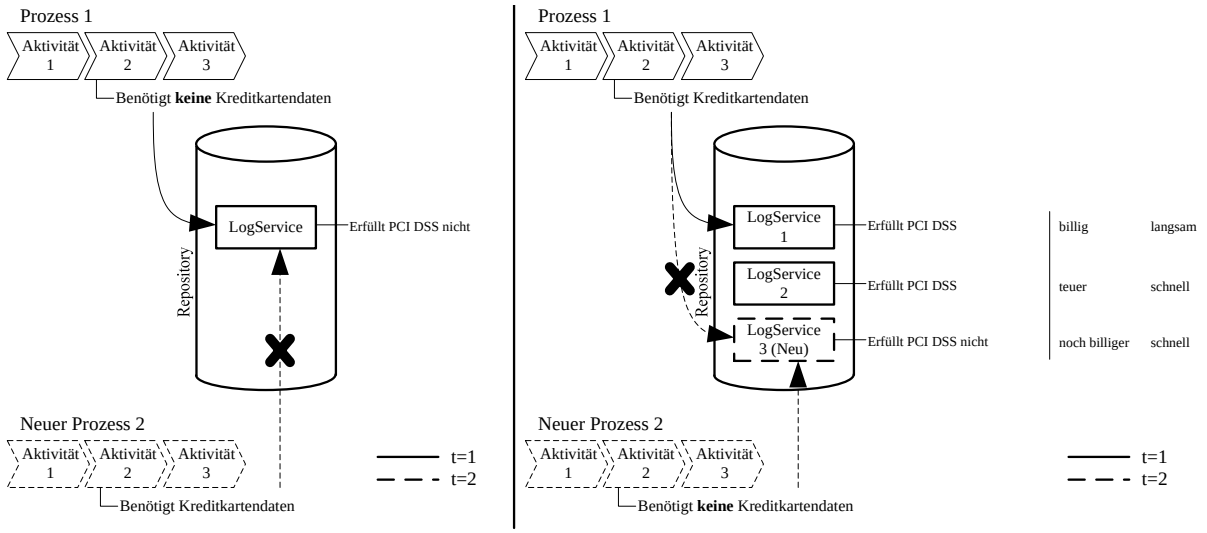

Abbildung 1: Auswirkungen von bestehenden Services auf neue Prozesse und umgekehrt

Die linke Seite der Abbildung 1 zeigt, dass es nicht genügt, das Angebot (die Services bzw. die Applikationen, welche die Services implementiert haben) alleine zu überprüfen, sondern dass zusätzlich der Kontext (Prozesse/Applikationen), in dem die Services genutzt werden, zu berücksichtigen ist. Eine logische Folgerung wäre, analog zu den Services die (neu erstellten oder geänderten) Prozesse zu überprüfen, bevor sie in das produktive System gelangen. Also wiederum eine organisatorische Maßnahme, ein Genehmigungsprozess für die Produktivsetzung der Prozesse. Jedoch genügt dies alleine nicht, wie der umgekehrte Fall (grafisch dargestellt in der rechten Seite der Abbildung 1) zeigt: Ein bestehender Prozess 1, welcher Kreditkartendaten benötigt, hat zwei bestehende Log-Services, einen billigeren, langsameren und einen teureren, schnelleren zur Auswahl. Beide erfüllen die benötigte PCI-DSS-Regulierung. Der Prozess sucht den billigsten Service[1], die Wahl fällt auf den Log-Service 1. Später wird ein weiterer Prozess 2 erstellt, welcher einen eigenen Log-Service 3 einbringt. Da dieser unterschiedliche nicht-funktionale Eigenschaften (Preis, Geschwindigkeit: Er ist noch billiger und trotzdem schnell, da er die Aussortierung kritischer Daten unterlässt und alles archiviert) hat, wird er zusätzlich bewilligt. Er erfüllt die PCI-DSS-Regulierung nicht, was für den Prozess 2 kein Problem darstellt, da er keine Kreditkartendaten benötigt. Jedoch greift der Prozess 1 beim nächsten Aufruf automatisch auf den nun billigsten, jedoch für ihn nicht konformen, Log-Service 3 zu. Das bedeutet, durch die Änderung des Service-Bestands im Repository wird zur Laufzeit automatisch ein nicht konformer Service gewählt.

4.2 Auswirkungen der dynamischen Bindung auf die Compliance

Wie nachfolgend ausgeführt wird, kann die dynamische Bindung verschiedene Ausprägungsformen aufweisen. Je nach eingesetzter Form, entsteht das Problem der Wiederverwendung in einem anderen Kontext gar nicht bzw. kann mit organisatorischen Maßnahmen gelöst werden. Bevor auf die Formen eingegangen wird, wird der Begriff der dynamischen Bindung definiert.

In einer SOA wird bei einer *Bindung* von Services zu Prozessen bzw. höherwertigen Services von den Implementierungsdetails (wie beispielsweise der physischen Adresse) abstrahiert, indem diese nicht fest codiert im Quellcode des implementierten Prozesses, sondern in einem separaten Dokument, der Service-Beschreibung, im Repository enthalten sind. Im Quellcode wird nur angegeben, auf welche Service-Beschreibung zugegriffen werden soll [PA05, 152ff; TS07, 18]. Diese Abstrahierung ermöglicht erst die in einer SOA verwendete Bindung zur Laufzeit (*dynamische Bindung*).

Analog zum oben stehenden Beispiel, der Abstrahierung von der physischen Adresse, liegt für [TS07, 18f] eine dynamische Bindung dann vor, wenn ein Service-Consumer erst zur Laufzeit die Adresse seines Service-Providers ermittelt. [Do05, 9] gehen einen Schritt weiter und verstehen unter diesem Begriff, dass zum Zeitpunkt der Code-Generierung meist nicht bekannt ist, welche Services überhaupt, beispielsweise aufgrund externer Einflüsse oder Benutzerpräferenzen, zur Laufzeit aufgerufen werden.

[PA05, 158f] untergliedern statische (Bindung zur Entwicklungszeit) und dynamische Bindung detaillierter: Von der Bindung zur "Registration time" (die Art, wie ein Service resp. seine Beschreibung im Repository erfasst wird, beeinflusst seine Auffindbarkeit) bis zur Bindung zur "Invocation time" (die Entscheidung, welcher Service benutzt wird, wird zum letztmöglichen Zeitpunkt, beim Serviceaufruf, gefällt). Die Autoren vermerken jedoch, dass in den meisten Fällen die Bindung zur "Invocation time" als dynamische Bindung bezeichnet wird. In diesem Beitrag wird unter dem Begriff der dynamischen Bindung ebenfalls die Bindung zum spätestmöglichen Zeitpunkt verstanden.

Neben dem *Zeitpunkt* der Servicebindung (dynamische Bindung) kann weiter untergliedert werden, wie viel Information im Quellcode enthalten ist, um den entsprechenden Service bzw. dessen Beschreibung im Repository zu identifizieren. Das bedeutet, dass verschiedene *Formen* der dynamischen Bindung unterschieden werden können.

---

[1] Eine detaillierte Betrachtung der Wirtschaftlichkeitsbewertung alternativer Geschäftsprozesskonfigurationen liefert [Br08].

4.2.1 Bindung anhand des Namens

Im einfachen und häufig angewandten Fall wird genau *ein* Service eindeutig anhand seines *Namens* im Repository identifiziert. Dies wird als "Binding by reference" [PA05, 155] bzw. "Runtime service lookup by name" [KBS04, 63] bezeichnet. Gemäß [KBS04, 63] ist die Service-Definition (wie der Service anzusprechen ist und was er als Resultat liefert) zur Entwicklungszeit dem Service-Nutzer bekannt und wird entsprechend berücksichtigt. In diesem Fall besteht das dynamische Element lediglich darin, dass der physische Ausführungsort (die Adresse) des Services nicht im vornherein bestimmt ist. So kann sich dieser ändern, ohne dass der Quellcode des implementierten Prozesses geändert werden muss. Ebenso besteht die Möglichkeit, denselben Service auf verschiedenen Maschinen bereitzustellen (in der Service-Beschreibung können mehrere physische Adressen, sogenannte "Endpoints", angegeben werden) und entsprechend der Auslastung den einen oder anderen Ausführungsort zu wählen. Eine solche Lastenverteilung kann von der Infrastruktur, beispielsweise dem Enterprise Service Bus, übernommen werden [Si07, S113]. Des Weiteren kann der Anbieter gewechselt werden, indem das Angebot im Repository entsprechend angepasst wird. Wird der Service bzw. dessen Beschreibung des neuen Anbieters unter demselben Namen abgelegt (und der "alte" Service entfernt), so greift der Prozess beim nächsten Aufruf automatisch auf den neuen Service zu.

Compliance-Nachweise: Wird als organisatorische Maßnahme die Compliance der Services geprüft, bevor diese im Repository registriert werden, ist diese Form der dynamischen Bindung eher unproblematisch: Der Service ist eindeutig bestimmt. Die dynamische Auswahl betrifft lediglich unterschiedliche Ausführungsorte. Das Problem der Wiederverwendung in einem anderen Kontext kann nicht entstehen, da zur Laufzeit kein anderer Service gewählt werden kann. Im Beispiel mit den Log-Services ist der Log-Service 1 anhand des Namens eingebunden, egal mit welchen Services das Repository erweitert wird, es wird immer auf diesen zugegriffen (rechte Seite der Abbildung 1). Da davon ausgegangen werden kann, dass nicht nur die Services, sondern auch die Prozesse selbst vor der produktiven Einführung auf die Erfüllung der vorgeschriebenen Regulierungen geprüft werden, kann im umgekehrten Fall (linke Seite der Abbildung 1) das Problem mit dieser organisatorischen Maßnahme gelöst werden.

4.2.2 Bindung anhand Bedingungen oder Eigenschaften

Im komplexen Fall werden die Services nach *Bedingungen* oder *Eigenschaften* im Repository gesucht. Dementsprechend wird diese Art als "Binding by constraint" [PA05, 156] bzw. als "Runtime service discovery based on reflection" [KBS04, 63f] bezeichnet. Krafzig et al. fügen eine weitere Zwischenform ein, die sie als "Runtime service lookup by properties" bezeichnen. Dabei wird zwar bereits nach Eigenschaften gesucht, aber nur innerhalb einer im Quellcode festgelegten Vorauswahl von Services und nicht innerhalb des gesamten aktuellen Bestands des Repositorys. Da bereits eine Vorauswahl getroffen wurde, ist es, im Gegensatz zur offenen Suche, nicht mehr zwingend nötig, die Semantik bei der Suche zu berücksichtigen [KBS04, 63f; PA05, 156]. Ein Beispiel für diesen Typ liefern [Sp07 und Br08, 104ff]: Um eine Aktivität in einem Geschäftsprozess abzuwickeln, stehen mehrere Services mit identischer Funktionalität, jedoch unterschiedlichen nicht-funktionalen Eigenschaften, wie Ausführungszeit oder Kosten, zur Verfügung (Vorauswahl). Wie im Beispiel mit den Log-Services besteht die Möglichkeit, zwischen einem langsameren, billigeren oder einem schnelleren, teureren Service zu wählen. Anhand der Präferenzen sowie Restriktionen wird die passende Service-Komposition automatisch ausgewählt. Die Festlegung, welche Services von der Funktionalität her identisch sind, wird manuell hinterlegt [Sp06, 5; Sp07, 316f; Br08, 110f]. Werden *mehrere* möglich anzuwendende Services als Resultat der Suche zurückgegeben, müssen diese evaluiert und so ein Service ausgewählt werden. Beispielsweise kann diese Auswahl auf die Dienstgüte der Services, die Reputation der Anbieter oder die Kosten Bezug nehmen [Be05, 269ff; PA05, 156f; Br08, 104f].

Compliance-Nachweise: Bei der Zwischenform ist die Situation ähnlich wie bei der Bindung anhand des Namens. Anstelle eines einzelnen Services ist hier eine Menge von Services vorgegeben. Mithilfe

von Dokumentation muss die Auswahl begründbar sein (Angabe der Kriterien oder Regeln). Bei der weitreichendsten Form jedoch, der offenen Suche, kann das Problem der Wiederverwendung in einem anderen Kontext nicht alleine mit der Überprüfung des Prozesses vor Produktivsetzung gelöst werden, wie das Beispiel der Log-Services zeigt: Da anhand der Eigenschaft Kosten bei jedem Aufruf von neuem der billigste Service innerhalb des gesamten aktuellen Bestands des Repositorys gesucht wird, kann aufgrund einer Änderung des Bestands beim nächsten Aufruf ein nicht konformer Service gewählt werden, wie im Beispiel der Log-Service 3 für den Prozess 1 (vgl. Abbildung 1, rechte Seite).

Zwar wird diese Form der dynamischen Bindung in der Praxis kaum angewandt [KBS04, 64], u.a. weil semantische Konzepte benötigt werden, von denen ebenfalls noch nicht zahlreiche Implementierungen existieren (vgl. z. B. [He05; Be07, 70; HKZ08]). Jedoch ist wünschenswert, dass zukünftig anhand von Zielvorgaben möglichst automatisiert Prozesse erstellt, geändert oder gelöscht werden können [He05]. Dafür wird eine Suche (und Einbindung) von Services anhand Bedingungen oder Eigenschaften benötigt. Aus diesem Grund sollten die daraus entstehende Compliance-Problematik frühzeitig erkannt und diesbezügliche Lösungsansätze erstellt werden.

4.3 Lösungsansätze

4.3.1 Bisherige Ansätze und ihre Limitationen

Eine Lösungsmöglichkeit sind die bereits in Abschnitt 4.1 erwähnten *organisatorischen* Maßnahmen, Genehmigungsprozesse für Services und Prozesse sowie Einschränkung der zur Auswahl stehenden Services. Jedoch genügen diese je nach angewandter Form der dynamischen Bindung nicht, wie im vorhergehenden Abschnitt erläutert wurde. Zudem sind sie aus Gründen der Wirtschaftlichkeit und der Fehleranfälligkeit, falls möglich, um *technische* Hilfsmittel zu ergänzen oder gar zu ersetzen.

Es gibt einige *Tools* auf dem Markt, welche die Compliance unterstützen sollen. Einen Überblick über SOX-Software-Produkte geben [Ag06]. Jedoch ist zu beachten, dass nicht alle Produkte Funktionalitäten wie die Überprüfung eines Geschäftsprozesses zur Laufzeit beinhalten [Fo06]. Zudem müssen Kontrollaktivitäten manuell implementiert werden [Ag06]. Ob es möglich ist, *alle* Regulierungen in *allen* Aspekten detailliert zu modellieren und zur Laufzeit zu überprüfen ist fraglich.

In Abschnitt 2 wurden bereits Lösungsansätze und ihre Limitationen für dieses Problem betrachtet. An dieser Stelle werden die Limitationen mit dem Beispiel der Log-Services anhand des Ansatzes von [He05] verdeutlicht: In dem Augenblick, indem der Prozess 1 (mit den Kreditkartendaten) produktivgesetzt wird, müsste bereits bedacht werden, dass u. U. zu einem späteren Zeitpunkt das Repository mit einem für diesen Prozess nicht konformen Service (Log-Service 3) erweitert werden könnte und vorsorglich eine Abfrage generiert werden, welche alle Prozesse auflistet, die Kreditkartendaten benötigen. Bei der Einfügung des Log-Services 3 ins Repository müsste erneut daran gedacht werden, die Abfrage auszuführen und die Bedingungen, nach denen die Prozesse suchen, allenfalls anzupassen. Dafür wären viele (fehleranfällige) manuelle Eingriffe nötig.

Zudem müssen bei allen Ansätzen die zu unterstützenden Regulierungen manuell bestimmt werden. Wird zur Laufzeit ein Service ausgewählt und soll gleichzeitig die Auswahl eines nicht konformen Service verhindert werden, wird eine technische Unterstützung für dieses Problem benötigt, denn wie das Beispiel der Log-Services zeigt, sind die zu erfüllenden Regulierungen kontextabhängig. Das bedeutet, dass zuerst der Kontext bestimmt werden muss, um in einem weiteren Schritt daraus die zu erfüllende(n) Regulierung(en) abzuleiten, welche dann durch den Service erfüllt werden muss/müssen.

4.3.2 Eigener Lösungsansatz

4.3.2.1 Anfrage und Kontext

Da es sich um die Nutzung des Services handelt, hängt der *Kontext* von den Eingabedaten, also dem Wert der Inputparameter, ab. Eine Möglichkeit wäre, ihn anhand des Datentyps der zu liefernden Daten zu bestimmen; dies setzt jedoch voraus, dass die Datentypen entsprechend genau festgelegt werden. Eine weitere Möglichkeit wäre, die Daten selbst auf Muster zu überprüfen, jedoch ist fraglich, ob daraus ein Kontext eindeutig bestimmbar ist. Die wohl beste Variante ist, nicht nur das Angebot, sondern auch die *Anfrage*, den aufzurufenden Prozess, der wiederum als Service betrachtet werden kann, semantisch zu beschreiben. Dabei muss eine Beschreibungssprache gewählt werden, welche die Anfragen separat behandelt.

Semantik und Semantic Web Services

Semantik ist der Begriff für die Bedeutung von Symbolen bzw. Zeichen. Um vermehrt automatisiertes Handeln von Maschinen zu ermöglichen, müssen diese die Sachverhalte verstehen können. Die Semantik muss somit in maschinenverarbeitbarer Form dargestellt werden. Dazu werden die Dokumente mit Zusatzinformationen in Form von Metadaten erweitert („annotiert“) [He02; Do05, 287; Kl06, 25].

Web Services, die zurzeit aktuelle Implementierungsform von Services in einer SOA, werden größtenteils in der Sprache WSDL beschrieben. Aus einer WSDL-Datei können jedoch nur syntaktische Informationen (*wie* wird der Service aufgerufen), beispielsweise der Name und der Datentyp eines Parameters, gelesen werden, nicht jedoch die Bedeutung (Semantik; *was* leistet der Service und was bedeutet der Parameter). Diese ist unstrukturiert in einem Textfeld oder teilweise in externen Dokumenten enthalten. Mit der Anfügung von zusätzlichen Metadaten, weitere Datenfelder sowie die Möglichkeiten von Referenzen, wird die Text-Beschreibung strukturiert aufgeteilt. Web Services, welche in dieser Art semantisch beschrieben werden, werden als *Semantic Web Services* bezeichnet [Do05, 283; PLL06, 509; Be07, 69 f, 77 f].

Ein weiterer wichtiger Punkt für eine automatisierte Verarbeitung ist eine einheitliche Verwendung (u. a. der Bezeichnungen) der Metadaten. Im Bereich Semantic Web Services gibt es entsprechende standardisierte Beschreibungssprachen. Die wichtigsten sind OWL-S, WSMO und WSDL-S. In [Kl06] sowie in [PLL06] werden sie erläutert und verglichen. Ein neuerer Ansatz stellt SAWSDL dar (vgl. z. B. [VS07]).

Ontologien

Auch wenn die Services nun detailliert und standardisiert beschrieben sind, kann es weiterhin Missverständnisse geben bezüglich des *Inhalts* der Metadaten, beispielsweise durch unterschiedlichen Sprachgebrauch von Menschen. Denn der Mensch kommuniziert zwar mit Worten, meint jedoch implizit nicht nur das Wort, sondern das abstrakte Konzept dahinter. So kann unter einem Jaguar ein Auto oder ein Tier verstanden werden. Weiter verschärft wird dies, wenn Services aus unterschiedlichen Quellen bezogen werden [MSS01, 394]. Diese zu wenig standardisierte Verwendung der Sprache bedingt ein einheitliches Vokabular, eine *Ontologie*. Eine Ontologie ist eine formale Beschreibungen von Begriffen (Konzepten) und ihren Beziehungen (Relationen), welche für eine Gruppe von Personen (Domäne) Gültigkeit hat (vgl. dazu [Gr93; MSS01; He02]). In den Semantic-Web-Services-Beschreibungssprachen können Referenzen zu solchen Domänen-Ontologien hergestellt werden.

4.3.2.2 Regulierung(en)

Als nächster Schritt muss die Beziehung zur/zu den zu erfüllenden *Regulierung(en)* hergestellt werden. Um Beziehungen zwischen Begriffen bzw. Konzepten darzustellen, sowie zur Sicherstellung eines gemeinsamen Vokabulars, eignet sich eine entsprechende Ontologie. In dieser können auch Synonyme berücksichtigt werden: So können Benutzer anstelle des allgemeinen "Kreditkartendaten"-Begriffs auch Kreditkartenbezeichnungen wie "Visa" oder "MasterCard" angeben. Für dieses Anwendungsgebiet (Domäne) sind die beiden Begriffe gleichbedeutend. Vorzugsweise sind bestehende Ontologien anzuwenden oder als Basis zu nehmen, um den Entwicklungs- und Wartungsaufwand zu verringern. In diesem Fall bieten sich zunächst Rechtsontologien an (vgl. z. B. [Ga05]); diese sind jedoch oft sehr umfassend und berücksichtigen juristische Fälle oder spezifische Situationen (wie z. B. das Handeln im Affekt). Demnach ist eine Ontologie vorzuziehen, welche sich auf Regulierungen konzentriert, zu denen Unternehmen ihre Compliance nachweisen müssen. Zu prüfen ist beispielsweise die Eignung der in [Gi05] beschriebenen Ontologie.

4.3.2.3 Angebot

Eine der vorgeschlagenen organisatorischen Maßnahmen lautete, dass alle Services im Repository compliant sein müssen. Jedoch ist mit einer solchen Regelung weiterhin unklar, zu welchen Regulierungen ein Service compliant ist. Um diese Situation zu klären, müsste die Beschreibung der *angebotenen* Services entsprechend erweitert werden.

Damit ein automatisierter Abgleich zwischen Anfrage und Angebot gemacht werden kann, sollte dieselbe Beschreibungssprache verwendet werden. Ebenso sollte auf dieselbe Ontologie Bezug genommen werden. Für extern bezogene Services könnte die Angabe des Partnerunternehmens, zu welchen Regulierungen ihr Service compliant ist, durch eine Prüfungs- oder Zertifizierungsgesellschaft bestätigt werden, sodass auch Services außerhalb des unternehmenseigenen Repositorys verwendet werden könnten.

In Abbildung 2 ist dies grafisch dargestellt: Sowohl der anfragende Prozess (Eingabedaten in einem spezifischen Kontext) wie auch der angebotene Web Service (zu welcher/n Regulierung(en) compliant) werden semantisch beschrieben und beziehen sich auf dieselbe Ontologie, in der die Zuordnung vom Kontext zur/zu den benötigten Regulierung(en) vorgenommen wird.

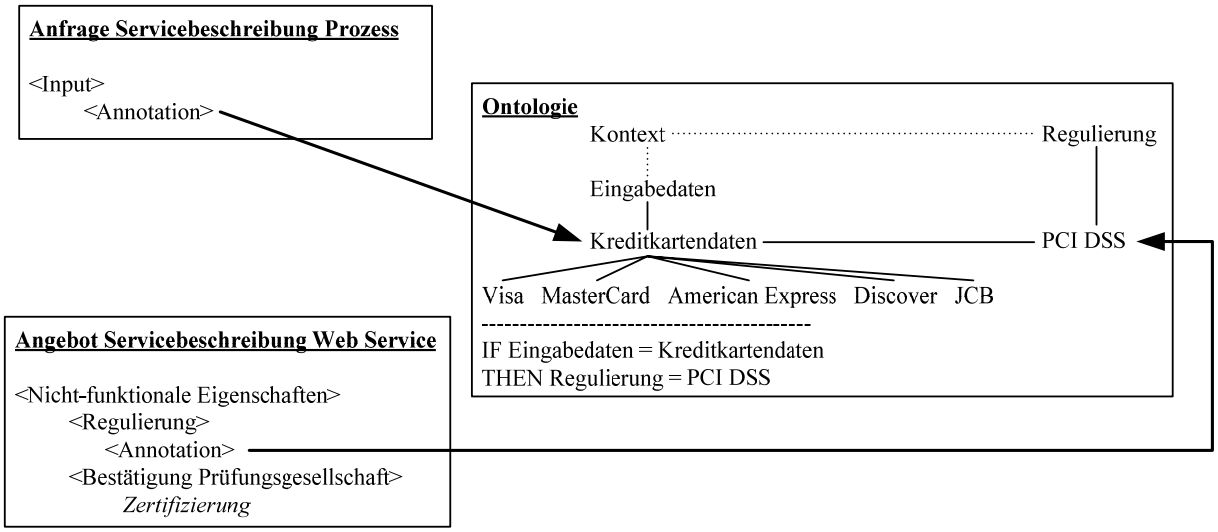

Abbildung 2: Lösungsskizze

4.3.2.4 Detaillierte Ontologie

Eine Ontologie sollte möglichst breit einsetzbar sein (abgebildete Regulierungen, nutzende Unternehmen). Da es Regulierungen gibt, welche abhängig von der Branche (z. B. REACH) oder anderen Kriterien erfüllt werden müssen, wird in einem weiteren Schritt der Kontext um die Unternehmensart erweitert. Das bedeutet, dass je nachdem in *welchem Unternehmen*, mit *welchen Daten* der Service eingesetzt wird, u. U. andere Regulierungen relevant sind. Die Unternehmensangaben können zentral gespeichert werden. Zur Prüfung einer Regulierung (und allfälliger damit einhergehender Zertifizierung) werden oftmals Prüfstandards angewandt, in denen genaue Testprozeduren angegeben werden. Prüfstandards können sich auf Frameworks beziehen.

In Abbildung 3 wird dies veranschaulicht, wobei (in blau) das Beispiel mit den Kreditkartendaten abgebildet ist. PCI DSS muss von allen Unternehmen (z. B. Händler) erfüllt werden, welche Kreditkartendaten einsetzen. Die Anforderungen basieren auf dem ISO/IEC-17799-Standard (welcher in der Zwischenzeit weiterentwickelt und unter der Referenznummer ISO/IEC 27002 veröffentlicht wurde). Die Applikationsanbieter (also auch Service-Anbieter) werden nach dem Payment Application Data Security Standard (PA-DSS) geprüft und zertifiziert, sofern die Software extern angeboten wird [Pc08]. Auch bei Eigenentwicklungen muss der PCI DSS eingehalten werden, jedoch wird für den unternehmensinternen Gebrauch keine Service-Zertifizierung (Bestätigung) benötigt.

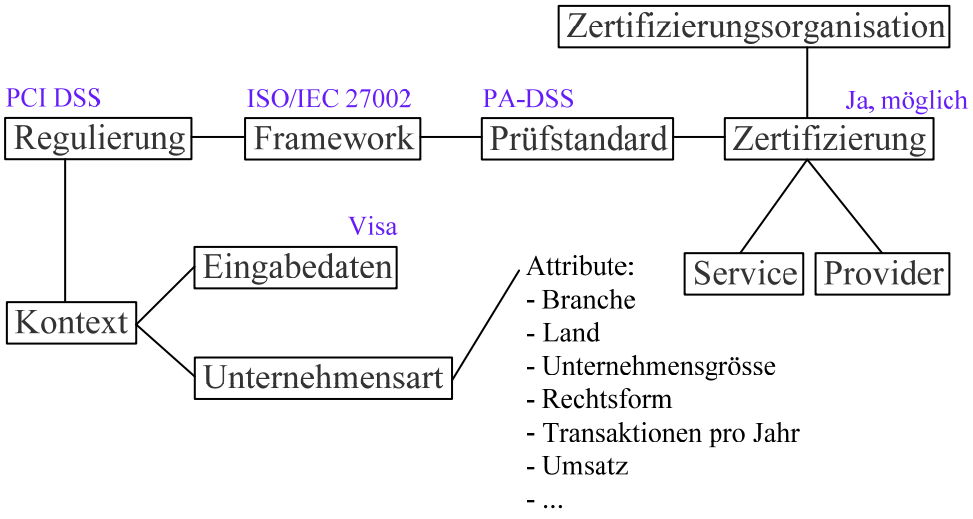

Abbildung 3: Ontologieskizze

5 ZUSAMMENFASSUNG UND AUSBLICK

Die dynamische Bindung, die semantischen Konzepte, und die Governance [Ab07, 7f] gehören zu den "Grand Challenges" der Forschung in Bezug auf serviceorientierte Architekturen [Pa06]. Dementsprechend sind auch Anwendungen der Semantik in der Praxis zurzeit rar. Jedoch sollte mit einer Ausweitung gerechnet und mögliche Probleme schon heute beachtet werden.

In diesem Beitrag wurde gezeigt, dass die verschiedenen Ausprägungsformen der dynamischen Bindung unterschiedliche Auswirkungen auf Compliance-Nachweise haben. Bei Anwendung der weitreichendsten Form der dynamischen Bindung besteht die Gefahr, dass bei Wiederverwendung von Services in einem anderen Kontext nicht konforme Services gewählt und demnach nicht alle erforderlichen Regulierungen erfüllt werden. Da bisherige Change-Management-Ansätze dieses Problem nicht lösen, wurde ein Ansatz vorgestellt, wie mithilfe von semantischen Konzepten der Kontext der Servicenutzung und, daraus folgend, die anzuwendende(n) Regulierung(en) bestimmt werden können. Dabei zeigt sich, dass semantische Konzepte nicht nur bei der Auswahl der Services, sondern auch für Compliance-Nachweise hilfreich sein können.

Als nächster wichtiger Schritt muss die Ontologie, welche in Abbildung 3 grob skizziert wurde, erstellt oder auf Basis einer anderen aufgebaut werden. Eine erste Analyse zeigte, dass Rechtsontologien als Basis wenig geeignet sind; zu prüfen sind Regulierungsontologien. Dabei müssen auch die Ontologiebeschreibungssprache sowie ihre Ausdrucksmächtigkeit berücksichtigt werden. Weiter muss eine geeignete semantische Service-Beschreibungssprache sowie Elemente derjenigen für die benötigten Angaben gewählt werden. Nach einem "proof of concept" könnte in einem zusätzlichen Schritt geprüft werden, ob die Ontologie (semi-)automatisch aus Regulierungs-Texten extrahiert werden kann.

LITERATURVERZEICHNIS

[Ab07] Aberdeen Group: Management and Governance: Planning for an Optimized SOA Application Lifecycle, Report, 2007.

[Ag06] Agrawal, R.; Johnson, C.; Kiernan, J.; Leymann, F.: Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In (Liu, L.; Reuter, A.; Whang, K.-Y.; Zhang, J. Hrsg.): Proc. 22nd International Conference on Data Engineering (ICDE 2006), Atlanta 2006. IEEE Computer Society, 2006; S. 92.

[Be05] Berbner, R.; Heckmann, O.; Mauthe, A.; Steinmetz, R.: Eine Dienstgüte unterstützende Webservice-Architektur für flexible Geschäftsprozesse. In: WIRTSCHAFTSINFORMATIK 47 (2005) 4; S. 268-277.

[Be07] Bell, D.; De Cesare, S.; Iacovelli, N.; Lycett, M.; Merico, A.: A framework for deriving semantic web services. In: Information Systems Frontiers 9 (2007) 1; S. 69-84.

[Br08] vom Brocke, J.: Serviceorientierte Architekturen – SOA – Management und Controlling von Geschäftsprozessen, Vahlen, München, 2008.

[Do05] Dostal, W.; Jeckle, M.; Melzer, I.; Zengler, B.: Service-orientierte Architekturen mit Web Services – Konzepte – Standards – Praxis, Spektrum-Akademischer Verlag, München, 2005.

[Eb06] Eidg. Bankenkommission (EBK): Rundschreiben 06/6 – Überwachung und interne Kontrolle, http://www.ebk.admin.ch/d/regulier/rundsch/2006/rs_0606_d.pdf, 2006.

[Er04] Erl, T.: Service-Oriented Architecture – Concepts, Technology, and Design, Prentice Hall PTR, Upper Saddle River et al., 2004.

[Fa07] Fabini, M.: Governance für komplexe SOA-Unternehmungen – Eine Vision für das Schweizer Gesundheitswesen. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 309-323.

[Fo06] Foody, D.: The Challenges of SOA – Which rules are necessary and which are just nice to have. In: SOA Web Services Journal 6 (2006) 9; http://webservices.sys-con.com/read/284550.htm.

[Ga05] Gangemi, A.; Sagri, M.; Tiscornia, D.: A constructive framework for legal ontologies. In (Benjamins, V.; Casanovas, P.; Breuker, J.; Gangemi, A. Hrsg.): Law and the Semantic Web. Springer, Berlin Heidelberg, 2005; S. 97-124.

[Gi05] Giblin, C.; Liu, A. Y.; Müller, S.; Pfitzmann, B.; Zhou, X.: Regulations Expressed As Logical Models (REALM), Research Report, IBM Research, http://www.zurich.ibm.com/security/publications/2005/GiLiMuPfZh05REALM-Report.pdf, 2005.

[Gr93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition 5 (1993) 2; S. 199-220.

[He02] Hesse, W.: Ontologie(n) – Aktuelles Schlagwort. In: Informatik Spektrum 25 (2002) 6; S. 477-480.

[He05] Hepp, M.; Leymann, F.; Bussler, C.; Domingue, J.; Wahler, A.; Fensel, D.: Semantic Business Process Management: Using Semantic Web Services for Business Process Management. In: Proc. IEEE ICEBE 2005, Beijing 2005. IEEE Computer Society Press, Los Alamitos, 2005.

[HKZ08] Haniewicz, K.; Kaczmarek, M.; Zyskowski, D.: Semantic Web Services Applications – a Reality Check. In: WIRTSCHAFTSINFORMATIK 50 (2008) 1; S. 39-46.

[JG07] Johannsen, W.; Goeken, M.: Referenzmodelle für IT-Governance – Strategische Effektivität und Effizienz mit COBIT, ITIL & Co, dpunkt.verlag, Heidelberg, 2007.

[Jo08] Josuttis, N.: SOA in der Praxis – System-Design für verteilte Geschäftsprozesse, dpunkt.verlag, Heidelberg, 2008.

[Ka07] Kalex, U.: Von der Geschäftsarchitektur zur SOA-Governance. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 325-340.

[KBS04] Krafzig, D.; Banke, K.; Slama, D.: Enterprise SOA – Service-Oriented Achitecture Best Practices, Prentice Hall PTR, Upper Saddle River et al., 2004.

[Ke07] Keller, W.: SOA-Governance – SOA langfristig durchsetzen und managen. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 289-307.

[Kl06] Klein, M.: Automatisierung dienstorientierten Rechnens durch semantische Dienstbeschreibungen, Dissertation, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Universitätsverlag Karlsruhe, Karlsruhe, 2006.

[KL06] Knolmayer, G.; Loosli, G.: IT Governance. In (Zaugg, R. J. Hrsg.): Handbuch Kompetenzmanagement. Durch Kompetenz nachhaltig Werte schaffen. Haupt Verlag, Bern Stuttgart Wien, 2006; S. 449-457.

[Kn07] Knolmayer, G.: Compliance-Nachweise bei Outsourcing von IT-Aufgaben. In: WIRTSCHAFTSINFORMATIK 49 (2007) Sonderheft; S. S98-S106.

[KW06] Knolmayer, G.; Wermelinger, T.: Der Sarbanes-Oxley Act und seine Auswirkungen auf die Gestaltung von Informationssystemen. In (Siegel, T.; Klein, A.; Schneider, D.; Schwintowski, H.-P. Hrsg.): Unternehmungen, Versicherungen und Rechnungswesen. Duncker&Humblot, Berlin, 2006; S. 513–536.

[LMX07] Liu, Y.; Müller, S.; Xu, K.: A static compliance-checking framework for business process models. In: IBM SYSTEMS JOURNAL 46 (2007) 2; S. 335-361.

[MSS01] Mädche, A.; Staab, S.; Studer, R.: Ontologien. In: WIRTSCHAFTSINFORMATIK 43 (2001) 4; S. 393-395.

[NS07] Namiri, K.; Stojanovic, N.: A Model-driven Approach for Internal Controls Compliance in Business Processes. In (Hepp, M.; Hinkelmann, K.; Karagiannis, D.; Klein, R.; Stojanovic, N. Hrsg.): Proc. of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), Innsbruck 2007. CEUR Workshop Proceedings Vol. 251, ISSN 1613-0073, 2007; S. 40-43.

[Og04] O´Grady, S.: SOA Meets Compliance: Compliance Oriented Architecture, Study, RedMonk, http://redmonk.com/public/COA_Final.pdf, 2004.

[PA05] Pautasso, C.; Alonso, G.: Flexible Binding for Reusable Composition of Web Services. In (Gschwind, T.; Assmann, U.; Nierstrasz, O. Hrsg.): Proc. 4th International Workshop on Software Composition (SC 2005), Edinburgh 2005. Springer, Berlin Heidelberg, 2005; S. 151-166.

[Pa06] Papazoglou, M. P.; Traverso, P.; Dustdar, S.; Leymann, F.; Krämer, B. J.: Service-Oriented Computing: A Research Roadmap. In (Cubera, F.; Krämer, B. J.; Papazoglou, M. P. Hrsg.): Proc. Dagstuhl Seminar 05462 - Service Oriented Computing (SOC), Dagstuhl 2006. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Dagstuhl, 2006.

[Pc04] Public Company Accounting Oversight Board: AUDITING STANDARD No. 2 – An Audit of Internal Control Over Financial Reporting Performed in Conjunction with An Audit of Financial Statements, http://www.pcaobus.org/Rules/Rules_of_the_Board/Auditing_Standard_2.pdf, 2004.

[Pc08] Payment Card Industry Security Standards Council: Payment Application Data Security Standard (PA-DSS), http://www.pcisecuritystandards.org/security_standards/pa_dss.shtml, 2008.

[PG03] Papazoglou, M. P.; Georgakopoulos, D.: Service-Oriented Computing. In: Communications of the ACM 46 (2003) 10; S. 25-28.

[PLL06] Polleres, A.; Lausen, H.; Lara, R.: Semantische Beschreibung von Web Services. In (Pellegrini, T.; Blumauer, A. Hrsg.): Semantic Web – Wege zur vernetzten Wissensgesellschaft. Springer, Berlin Heidelberg, 2006; S. 505-524 .

[Si07] Siedersleben, J.: SOA revisited: Komponentenorientierung bei Systemlandschaften. In: WIRTSCHAFTSINFORMATIK 49 (2007) Sonderheft; S. S110-S117.

[Sp06] Spahn, M.; Berbner, R.; Heckmann, O.; Steinmetz, R.: Ein heuristisches Optimierungsverfahren zur dienstgütebasierten Komposition von Web-Service-Workflows, Technical Report, Technische Universität Darmstadt, ftp://ftp.kom.e-technik.tu-darmstadt.de/pub/papers/SBHS06-1-paper.pdf, 2006.

[Sp07] Spahn, M.: Ein heuristisches Verfahren zur dienstgütebasierten Optimierung flexibler Geschäftsprozesse. In (Braun, T.; Carle, G.; Stiller, B. Hrsg.): Proc. 15. Fachtagung Kommunikation in Verteilten Systemen (KiVS 2007), Bern 2007. Springer, Berlin Heidelberg, 2007; S. 315-322.

[SS07] Schelp, J.; Stutz, M.: SOA Governance. In: HMD - Praxis der Wirtschaftsinformatik 43 (2007) 253; S. 66-73.

[Th01] Thelesklaf, D.: Outsourcing von Compliance-Dienstleistungen – Compliance als Teil des Risk Managements. In: Der Schweizer Treuhänder 75 (2001) 5; S. 447-452.

[TS07] Tilkov, S.; Starke, G.: Einmaleins der serviceorientierten Architekturen. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 9-36.

[VS07] Verma, K.; Sheth, A.: Semantically Annotating a Web Service. In: IEEE Internet Computing 11 (2007) 2, S. 83-85.

[WS07] Winter, R.; Schelp, J.: Agilität und SOA. In (Starke, G.; Tilkov, S. Hrsg.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt.verlag, Heidelberg, 2007; S. 41-47.

[ZTZ07] Zhao, J. L.; Tanniru, M.; Zhang, L.-J.: Services computing as the foundation of enterprise agility: Overview of recent advances and introduction to the special issue. In: Information Systems Frontiers 9 (2007) 1; S. 1-8.