Eine Dienstgüte unterstützende Web-Service-Architektur für flexible Geschäftsprozesse

Die Autoren

Rainer Berbner
Oliver Heckmann
Andreas Mauthe
Ralf Steinmetz

Dipl.-Wirtsch.-Inf. Rainer Berbner
Dr.-Ing. Oliver Heckmann
Dr. Andreas Mauthe
Prof. Dr.-Ing. Ralf Steinmetz
Multimedia Communications Lab KOM
Technische Universität Darmstadt
Merckstraße 25
64283 Darmstadt
{berbner | heckmann | mauthe | steinmetz}@kom.tu-darmstadt.de

1 Einleitung

Globalisierte und deregulierte Märkte sowie anspruchvoll gewordene Kunden haben zu einem hohen Kostendruck und harten Wettbewerb geführt. Um in diesem Wettbewerb als Unternehmen bestehen zu können, sind innovative und dynamisch adaptierbare Geschäftsprozesse ein wichtiger Erfolgsfaktor [BeKa03; MCLP03]. Grundvoraussetzung für dynamisch adaptierbare Geschäftsprozesse ist eine flexible IT-Architektur, da Geschäftsprozesse eine Vielzahl von heterogenen Legacy-Anwendungen, Plattformen, Betriebssystemen und Kommunikationsmechanismen integrieren müssen [Kell02, 25]. Diese Heterogenität hat zu einer Inflexibilität und Komplexität geführt, die kaum noch beherrschbar scheint [Kell02, 23ff.]. So hat eine Forrester-Studie [KRLB01] ergeben, dass bei 3500 befragten Unternehmen weniger als 35 % der durchgeführten Integrationsprojekte zeitlich und finanziell im vorgegebenen Rahmen durchgeführt werden konnten. Während diesbezüglich in der als international kompetitiv anerkannten deutschen Automobilindustrie durch eine einheitliche Plattformstrategie schon Erfolge erzielt worden sind [BuKu03, 505], besteht in anderen Bereichen, wie z. B. dem Finanzsektor, noch großer Nachholbedarf [HeKö03; Kell02, 28f; StBM04].

Die Heterogenität der IT-Landschaften ist umso gravierender, wenn man bedenkt, dass im Zuge der Globalisierung und des *Business Process Outsourcing* (BPO) [Frös99; Ried03] verteilte, organisationsübergreifende Geschäftsprozesse eine immer größere Bedeutung erlangen [LeRo00, 4ff.]. Diese Entwicklung stellt insofern eine besondere Herausforderung für die beteiligten Unternehmen dar, als zusätzlich zur Überwindung der internen Heterogenität im eigenen Unternehmen noch die Kopplung mit den unterschiedlichsten IT-Systemen der Geschäftspartner realisiert werden muss [LiWi04].

In zunehmendem Maße gewinnen Webservices als Technologie zur Realisierung verteilter Geschäftsprozesse an Bedeutung [ACDG03; ACKM03, 245ff; LeRo02a; PrSc02]. Webservices sind lose gekoppelte, wiederverwendbare Softwarekomponenten, die über das Internet aufgerufen werden können und über den Austausch von Nachrichten miteinander kommunizieren [z. B. EbFi03, 66; W3C04a]. Webservices unterschieden sich von anderen Integrationstechnologien dadurch, dass sie auf offenen XML-Standards wie SOAP (*Simple Object Access Protocol*) [W3C03], WSDL (*Webservice Definition Language*) [W3C01] und UDDI (*Universal Description, Discovery and Integration*) [UDDI03] beruhen

Kernpunkte

Webservices eignen sich zur Realisierung verteilter Geschäftsprozesse, sofern die Dienstgüte der beteiligten Webservices garantiert wird. Dieser Beitrag beschreibt eine Web-Service-Architektur, die Dienstgüte wie folgt unterstützt:

- Ein Lebenszykluskonzept gewährleistet, dass nur Webservices in kritischen Geschäftsprozessen verwendet werden, deren Dienstgüte den Anforderungen des Nutzers genügt.
- Die Dienstgüteeigenschaften der Webservices müssen in Form von Service Level Agreements (SLAs) zugesichert werden.
- Die Auswahl konkreter Webservices erfolgt dynamisch zur Laufzeit unter Berücksichtigung der zugesicherten Dienstgüteeigenschaften.
- Die Einhaltung der SLAs wird zur Laufzeit vom System überwacht.

Stichworte: Web-Service-Architektur, Web-Service-Dienstgüte, Lebenszykluskonzept für Webservices, Service Level Agreement (SLA)

und somit in verteilten, heterogenen Umgebungen eine Unabhängigkeit von den eingesetzten Plattformen, Betriebssystemen und Programmiersprachen ermöglichen. Der Web-Service-Technologie liegt das Paradigma der *Service-oriented Architecture* (SoA) zugrunde, das sich durch die lose Kopplung sowie die Ortunabhängigkeit der beteiligten Dienste auszeichnet [Papa03].

Unsere Arbeiten [z. B. BeMS04a; StBM04] im Rahmen des Projektes *E-Finance Lab* (http://www.efinancelab.de) haben gezeigt, dass Webservices grundsätzlich als Technologie zur Realisierung verteilter Geschäftsprozesse geeignet sind, sofern deren Dienstgüte (*Quality of Service* (QoS)) von den jeweiligen Web-Service-Providern garantiert wird. Die Unterstützung der Dienstgüte durch die zugrunde liegende Web-Service-Architektur bleibt aber in vielen Veröffentlichungen zu Web-Service-basierten Geschäftsprozessen [z. B. ACKM03, 245–293; PrSc02; Schm03] unberücksichtigt bzw. wird nur unzureichend beachtet. Der Fokus des *Webservice Modelling Framework* (WSMF) liegt auf der Überwindung syntaktischer und semantischer Heterogenität bei der Komposition von Webservices zu Web-Service-Anwendungen [FeBu02]. Die Dienstgüte der Webservices bleibt hierbei unberücksichtigt. In [CaSh02; CaSM02; SMSV03] wird Dienstgüte neben der Suche geeigneter Webservices als Herausforderung bei der Integration von Webservices in Workflows adressiert. Im Mittelpunkt stehen hier die Wahl geeigneter Dienstgüteeigenschaften und deren semantische Beschreibung mit Hilfe von Ontologien. In [ShAS03] wird beschrieben, wie Dienstgüte für Webservices durch eine Priorisierung der Web-Service-Aufrufe realisiert werden kann. Verschiedene Dienstgüteeigenschaften für Webservices werden in [Ran03; KaKL03; ZBDK03] diskutiert, deren technische Unterstützung durch ein System bleibt jedoch weitgehend unbeachtet.

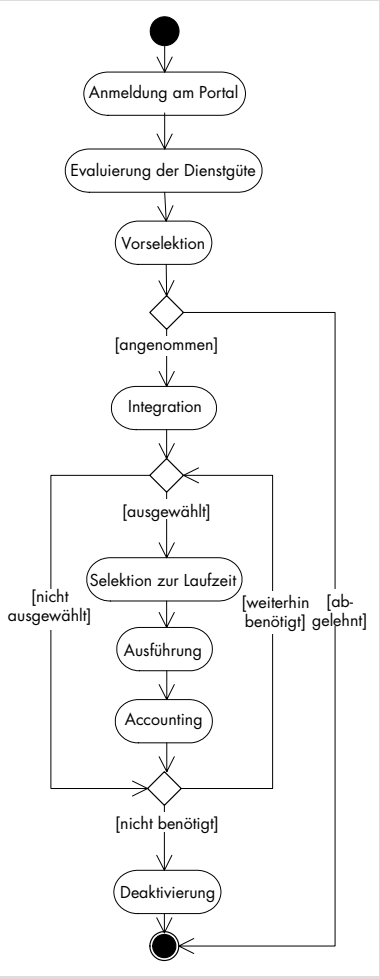

Bild 1 Lebenszykluskonzept zur Dienstgüteunterstützung für Webservices

In diesem Beitrag stellen wir eine Dienstgüte unterstützende Architektur zur Realisierung flexibler Geschäftsprozesse sowie deren prototypische Implementierung vor. Unsere Architektur unterscheidet sich von anderen Arbeiten durch eine umfassende Unterstützung von Dienstgüte:

- Externe Provider müssen sich an dem Portal des Nutzers registrieren, um einen Webservice anmelden zu können.
- Dienstgüteeigenschaften werden seitens des Web-Service-Providers in *Service Level Agreements (SLAs)* garantiert.
- Der Architektur liegt ein von uns entwickeltes Lebenszykluskonzept zugrunde, das die verschiedenen Phasen beschreibt, die ein Webservice durchläuft, bevor er in kritischen Geschäftsprozessen verwendet wird. Dieses Lebenszykluskonzept gewährleistet, dass nur Webservices in kritischen Geschäftsprozessen eingesetzt werden, deren zugesicherte Dienstgüte den Anforderungen des Nutzers genügt.
- Die Auswahl konkreter Webservices erfolgt dynamisch zur Laufzeit unter Berücksichtigung der zugesicherten Dienstgüteeigenschaften.
- Die Einhaltung der SLAs wird zur Laufzeit überwacht.

Wie Dienstgüte technisch auf Seiten des Web-Service-Providers bzw. auf Netzwerkebene sichergestellt wird, ist nicht Gegenstand dieses Beitrags. Diesbezüglich wird auf [Heck04; Schm01] verwiesen.

Der Beitrag ist wie folgt gegliedert: In Abschnitt 2 werden im Rahmen eines Lebenszykluskonzeptes die verschiedenen Phasen beschrieben, die ein extern bereitgestellter Webservice durchläuft, bevor er in einem Geschäftsprozess zum Einsatz kommt. Die Komponenten der von uns entwickelten Web-Service-Architektur und deren Zusammenwirken werden in Abschnitt 3 dargestellt. In Abschnitt 4 wird die prototypische Implementierung unserer Architektur beschrieben. Abschließend werden in Abschnitt 5 eine kurze Zusammenfassung und ein Ausblick auf unsere weiteren Forschungsaktivitäten gegeben.

2 Lebenszyklus eines Webservice im Rahmen unserer Architektur

Es ist davon auszugehen, dass in den nächsten Jahren die Zahl der kommerziell nutzbaren Webservices stark ansteigen wird. Dies wird zur Folge haben, dass eine Vielzahl von Webservices mit gleicher oder ähnlicher Funktionalität von verschiedenen Providern angeboten wird. In diesem Zusammenhang werden die nichtfunktionalen Dienstgüteeigenschaften eines Webservice zunehmend an Bedeutung gewinnen [MaNa02]. Die Vorstellung, dass Unternehmen Webservices ohne Kenntnis deren Dienstgüteeigenschaften, angeboten von oftmals unbekannten Providern, in kritischen Geschäftsprozessen ihrer Produktivsysteme verwenden, entspricht nicht der betrieblichen Realität. Im Folgenden stellen wir ein von uns entwickeltes Lebenszykluskonzept[1] zur Dienstgüteunterstützung (Bild 1) vor, das sicherstellt, dass nur solche Webservices in kritischen Geschäftsprozessen zum Einsatz kommen, deren garantierte Dienstgüteeigenschaften auch den Vorstellungen des Nutzers entsprechen.

1. Phase: Anmeldung am Portal des Nutzers

Möchte ein externer Web-Service-Provider einen Webservice anbieten, so muss er sich zuvor, falls noch nicht in der Vergangenheit geschehen, am Portal des Nutzers registrieren. Hierbei wird der Provider aufgefordert, Informationen über sein Unternehmen, wie z. B. Anschrift und Ansprechpartner, zu hinterlegen. Nach der Registrierung kann der Service-Provider einen von ihm angebotenen Webservice am Portal anmelden. Hierzu muss er verbindliche Angaben zu den Dienstgüteeigenschaften des von ihm offerierten Webser-

vice in Form eines SLAs machen. Bei einem SLA [RFC01] handelt es sich hierbei um einen Vertrag zwischen einem Dienstleister und einem Kunden, der die Parameter einer zu erbringenden Dienstleistung und deren Qualitätsniveau spezifiziert. Die jeweiligen Parameter werden in Form von *Service Level Objectives* (SLOs) definiert. Optional kann das SLA auch die durch die Erbringung einer Dienstleistung entstehenden Kosten festlegen.

Die Anmeldung der Webservices kann als Reaktion auf veröffentlichte Ausschreibungen seitens des Nutzers geschehen. Zudem besteht aber auch die Möglichkeit, dass der Nutzer selbst in öffentlichen UDDI-Verzeichnisdiensten nach geeigneten Webservices externer Provider sucht.

2. Phase: Evaluierung der Dienstgüte

Nachdem der Webservice mit zugehöriger SLA am Portal angemeldet wurde, erfolgt die Bewertung dessen Dienstgüte durch den Nutzer. Als Bewertungsgrundlage werden in Anlehnung an [GoKP03; KaKL03; MaNa02; TGNR03] die folgend aufgeführten Parameter verwendet.

- *Verfügbarkeit (availability)*
  Die Verfügbarkeit eines Webservice gibt an, mit welcher Wahrscheinlichkeit dieser verfügbar ist, wenn ein Nutzer ihn aufruft. Ein Webservice gilt dann als verfügbar, wenn er in der Lage ist, eine Anfrage innerhalb einer festgelegten Zeitspanne zu bearbeiten.
- *Leistungsfähigkeit (performance)*
  Leistungsfähigkeit als Oberbegriff kann durch die Auswertung zweier Teilkriterien definiert werden: Der Durchsatz (*throughput*) ist die Anzahl der Anfragen, die in einer vorgegebenen Zeitspanne bearbeitet werden können. Die Antwortzeit (*response time*) setzt sich zusammen aus Übertragungs- und Bearbeitungszeit (*transmission/processing time*) und entspricht der Zeit, die benötigt wird, um die angeforderte Antwort zu erhalten.
- *Fehlerhäufigkeit (error rate)*
  Die Fehlerhäufigkeit bezeichnet die Anzahl der Fehler innerhalb eines bestimmten Zeitintervalls. Grundsätzlich können in diesem Zusammenhang verschiedene Fehlerarten, wie z. B. fehlerhafte Ergebnisse oder Übertragungsfehler, unterschieden werden.
- *Sicherheit (security)*
  Sicherheit ist ein umfassendes Bewertungskriterium und wird daher anhand der Teilkriterien Authentizität (der beteiligten Partner), Autorisierung (der Nutzer), Vertraulichkeit und Datenverschlüsselung bewertet.
- *Reputation des Webservice bzw. Service-Providers*
  In diesem Zusammenhang sind die bisherigen Erfahrungen mit dem jeweiligen Service bzw. dessen Provider zu bewerten. Dieses Kriterium ermöglicht, einen Service höher zu bewerten, wenn in der Vergangenheit schon positive Erfahrungen mit ihm bzw. mit dessen Provider gemacht wurden. Durch das Kriterium Reputation können zudem auch Referenzen des Providers mit in die Bewertung einbezogen werden.
- *Kosten der Web-Service-Nutzung*
  Auch wenn Kosten nicht direkt zu den eigentlichen Dienstgütekriterien zählen, stellen sie doch eine wichtige nichtfunktionale Eigenschaft dar, wenn es darum

Webservice Nr. 34790

	Beurteilung	**Gewichtungsfaktor (0–10 Punkte)**	**Bewertung (0–5 Punkte)**
Sicherheit			
Authentifizierung	X.509	9	5
Autorisierung	Triple-DES	10	5
Verschlüsselte Datenübertragung	nicht unterstützt	3	0
Reputation			
Bisherige Erfahrungen (Provider)	Gute Erfahrungen	5	4
Bisherige Erfahrungen (Webservice Nr. 34790)	Keine Erfahrung	0	–
Externe Referenzen	Automobilindustrie	6	4
Kosten			
Tarif	Pauschaltarif, 50 €/Monat	8	3

Gewichtung: 0 Punkte: irrelevant 10 Punkte: relevant
Bewertung: 0 Punkte: mangelhaft 5 Punkte: herausragend

Bild 2 Auszug eines Webservice SLA (quantitativ nicht messbare Kriterien)

$$b_{WS_i}^{\text{Antwortzeit}} = \frac{\max\left(v_{WS_1}^{\text{Antwortzeit}}; \ldots; v_{WS_n}^{\text{Antwortzeit}}\right)}{v_{WS_i}^{\text{Antwortzeit}}}$$

$v_{WS_i}^{\text{Antwortzeit}}$ Antwortzeit des Webservice i

$b_{WS_i}^{\text{Antwortzeit}}$ normalisierte Antwortzeit des Webservice i

Bild 3 Normalisierung am Beispiel der Antwortzeit

$$b_{WS_i}^{\text{Durchsatz}} = \frac{v_{WS_i}^{\text{Durchsatz}}}{\min\left(v_{WS_1}^{\text{Durchsatz}}; \ldots; v_{WS_n}^{\text{Durchsatz}}\right)}$$

$v_{WS_i}^{\text{Durchsatz}}$ Durchsatz des Webservice i

$b_{WS_i}^{\text{Durchsatz}}$ normalisierter Durchsatz des Webservice i

Bild 4 Normalisierung am Beispiel des Durchsatzes

$$\begin{aligned}\text{Gesamtbewertung}\,(WS_i) = {} & \text{Gewichtungsfaktor}_{\text{Kriterium 1}} * b_{WS_i}^{\text{Kriterium}_1} \\ & + \text{Gewichtungsfaktor}_{\text{Kriterium 2}} * b_{WS_i}^{\text{Kriterium}_2} \\ & + \ldots \\ & + \text{Gewichtungsfaktor}_{\text{Kriterium } n} * b_{WS_i}^{\text{Kriterium}_n} \\ & \text{wobei } 0 \leq \text{Gewichtungsfaktor} \leq 10\end{aligned}$$

Bild 5 Bewertungsfunktion

geht, einen geeigneten Webservice auszuwählen. Die Abrechnung der Webservices kann grundsätzlich nach der Anzahl der Aufrufe, der Nutzungsdauer, des verarbeiteten Datenvolumens oder nach einem Pauschaltarif erfolgen [GoKP03].

Diese Kriterien lassen sich in quantitativ messbare und „weiche" Kriterien einteilen. Zu den quantitativ messbaren Kriterien gehören Verfügbarkeit, Leistungsfähigkeit und Fehlerhäufigkeit. Demgegenüber sind die Bewertungskriterien Sicherheit und Reputation „weiche" Kriterien, da sie nicht quantitativ vom System gemessen werden können. Daher müssen diese „weichen" Kriterien von IT-Verantwortlichen auf Nutzerseite anhand einer Skala zwischen 0 Punkten (mangelhaft) und 5 Punkten (herausragend) bewertet werden [BeMS04a]. Hierzu wird in unserem Ansatz eine Bewertungsmatrix (Bild 2) verwendet. Wird ein neuer Webservice am Portal angemeldet, so ist es Aufgabe eines IT-Verantwortlichen, die „weichen" Kriterien anhand dieser Bewertungsmatrix zu evaluieren. Um die „weichen" Bewertungskriterien miteinander in Beziehung zu setzen, werden diese jeweils mit einem Faktor gewichtet, welche der Präferenz des Entscheidungsträgers auf Nutzerseite entspricht. Sind z. B. Authentifizierung und Autorisierung entscheidungsrelevante Kriterien, während die verschlüsselte Übertragung der Daten vernachlässigbar ist, so kann dies durch die Festlegung der Gewichtungsfaktoren berücksichtigt werden.

Zur Bewertung der quantitativ messbaren Kriterien ist keine manuelle Bewertung anhand einer Bewertungsmatrix notwendig, lediglich die Gewichtungsfaktoren müssen festgelegt werden. Es stellt sich allerdings das Problem, dass die quantitativ messbaren Kriterien in unterschiedlichen Metriken vorliegen, wie z. B. Millisekunden oder Prozent. Daher werden diese Werte normalisiert. Bei Bewertungskriterien, deren optimale Werte möglichst gering sein sollen, wie z. B. die Antwortzeit, wird der normalisierte Wert eines Webservice WS_i wie folgt berechnet (Bild 3): Unter n zur Auswahl stehenden Webservices wird die Antwortzeit des Webservice mit der längsten Antwortzeit durch die Antwortzeit des jeweiligen Webservice WS_i dividiert.

Bei anderen Bewertungskriterien wie etwa dem Durchsatz sind möglichst hohe Werte wünschenswert. Deren Normalisierung wird erreicht, indem z. B. der Durchsatz des jeweiligen Webservice WS_i durch den Wert des Webservice mit dem geringsten Durchsatz dividiert wird (Bild 4).

Durch diese Normalisierung wird die Vergleichbarkeit der Webservices bezüglich ihrer Dienstgüte möglich. Zur endgültigen Bewertung der Web-Service-Dienstgüte wird aus Gründen der Einfachheit eine lineare Bewertungsfunktion (Bild 5) verwendet. Hierbei lässt sich der Nutzen eines Webservice hinsichtlich seiner Dienstgüteeigenschaften als Summe der gewichteten Einzelkriterien $b_{WS_i}^{\text{Kriterium}}$ auffassen. Bei den quantitativ messbaren Kriterien entspricht $b_{WS_i}^{\text{Kriterium}}$ dem normalisierten Wert des jeweiligen Dienstgütekriteriums, bei den „weichen" Kriterien wird die Bewertung aus der Bewertungsmatrix herangezogen.

So ist sichergestellt, dass Webservices verschiedener Provider miteinander auf Basis ihrer Dienstgüteeigenschaften verglichen werden können. Zudem werden Präferenzen und einsatzspezifische Besonderheiten durch Gewichtungsfaktoren berücksichtigt.

3. Phase: Vorselektion

In dieser Phase wird auf Grundlage der bisher vorgenommen Bewertung entschieden, ob der Webservice in das interne Verzeichnis übernommen wird. Hierzu können Ausschlussregeln definiert werden, die sich entweder auf die Gesamtbewertung des Webservice oder auf einzelne Teilkriterien beziehen. Eine Ausschlussregel bezüglich der Gesamtbewertung könnte beispielsweise lauten: „Webservices, deren Gesamtbewertung kleiner als n Punkte ist, werden nicht in das interne Verzeichnis übernommen." Bezogen auf einzelne Teilkriterien können Ausschlussregeln definiert werden, die z. B. beinhalten, dass Webservices, deren externe Referenzen mit weniger als 4 Punkten bewertet wurden, abgelehnt werden. Prinzipiell ist es auch denkbar, Ausschlussregeln miteinander zu verknüpfen. Zudem können statt absoluten auch relative Regeln definiert werden, die beispielsweise Webservices ausschließen, deren Gesamtbewertung nicht unter den TOP 10 liegt. Unabhängig von den Bewertungen können in der Phase Vorselektion auch solche Webservices ausgeschlossen werden, mit deren Provider der Nutzer

2051365 Wirtschaftsinformatik Heft 4/2005 3B2 Pos. 28 (Autor: Rainer Berbner) w:/p_3/wir/wir05_04/0383/wir0383u.3d
Stand: 16. 6. 2005 (gelieferte Daten) Bearbeiter: Hasert

z. B. aus strategischen Gründen nicht zusammenarbeiten möchte.

4. Phase: Integration des Webservice in das interne Verzeichnis
Webservices, welche die Phase Vorselektion erfolgreich durchlaufen haben, werden in das interne Verzeichnis übernommen und stehen zur Integration in die jeweiligen Geschäftsprozesse zur Verfügung. Die Provider der nicht berücksichtigten Webservices können über Gründe für die Ablehnung benachrichtigt werden.

5. Phase: Selektion eines Webservice zur Laufzeit
Ausschlaggebend, ob und wie häufig ein Webservice aus dem internen Verzeichnis tatsächlich ausgewählt wird, ist die Bewertung seiner Dienstgüteeigenschaften aus Phase 2. Grundsätzlich wird derjenige Webservice ausgewählt, der die höchste Gesamtbewertung in seiner Kategorie (z. B. Einholung eines Kunden-Ratings) erhalten hat. Im Zusammenhang mit Geschäftsprozessen, an die besonders hohe Anforderungen gestellt werden, können aber auch noch für die in Frage kommenden Webservices zusätzliche Nebenbedingungen definiert werden, welche die Ausschlussregeln aus der Phase Vorselektion verschärfen. So ist es Unternehmen möglich, flexibel auf äußere Einflüsse zu reagieren. Sollte beispielsweise kurzfristig mit einem starken Anstieg von Kundenanfragen für einen bestimmten Webservice zu rechnen sein, so kann durch eine zusätzliche Nebenbedingung festgelegt werden, dass der zum Einsatz kommende externe Webservice einen besonders hohen Durchsatz garantiert.

6. Phase: Ausführung
Derjenige Webservice, der in der vorigen Phase als optimal bezüglich der vorgegebenen Kriterien ermittelt wurde, wird nun dynamisch zur Laufzeit in den Workflow integriert und ausgeführt.

7. Phase: Accounting
Für extern bezogene Webservices erhält das Unternehmen eine Abrechnung, die auf unterschiedlichen Preismodellen (z. B. nutzungsabhängige und -unabhängige Tarife) basieren kann [TGNR03]. Zur Erstellung dieser Abrechnung führt der externe Web-Service-Provider ein Accounting durch. Unter Accounting wird hierbei die verursachungsgerechte Zurechnung von Aktivitäten eines Informationssystems zu einer Ressource verstanden [ATIS01]. Aber auch für den Nutzer der Webservices ist ein Accounting von Bedeutung, da so eine verursachungsgerechte Kostenzuordnung zu den internen Organisationseinheiten, die den Service tatsächlich in Anspruch genommen haben, möglich wird.

Zudem werden die durch das Accounting gewonnenen Daten für die Berechnung der Dienstgütekriterien (z. B. Verfügbarkeit) verwendet.

8. Phase: Deaktivierung des Webservice
Wird die von einem Webservice bereitgestellte Funktionalität nicht mehr benötigt, so kann der Nutzer diesen aus dem internen Verzeichnis entfernen. Die Deaktivierung kann auch dann notwendig werden, wenn der Service Provider den entsprechenden Webservice aus seinem Angebot nimmt.

3 Architekturkomponenten

In diesem Abschnitt werden die Komponenten der von uns entwickelten Web-Service-Architektur beschrieben (Bild 6).

Die Aufnahme externer Webservices in den internen Verzeichnisdienst erfolgt über ein vom Nutzer bereitgestelltes *Portal*. An diesem Portal muss sich ein externer Provider registrieren, möchte er einen Webservice anbieten. Zudem muss er ein SLA hinterlegen, das verbindliche Aussagen zu den in Abschnitt 2 aufgeführten Dienstgütekriterien macht. Dieses SLA wird nach der Anmeldung des Webservice von der *SLA-Management-Komponente* verwaltet. Hierbei extrahiert sie die Dienstgüteeigenschaften und die Gültigkeitsdauer aus dem SLA und legt sie in der Datenbank (DB) ab. Die *Assessment-Komponente* realisiert die Evaluierung und Bewertung der Dienstgüteeigenschaften (Phase 2). Des Weiteren integriert sie diejenigen Webservices in das interne Verzeichnis, die die Vorselektion (Phase 3) erfolgreich durchlaufen haben.

Das interne Verzeichnis beinhaltet Referenzen auf die am Portal angemeldeten Webservices. Hierbei kann es sich sowohl um interne als auch externe Webservices handeln. Intern bereitgestellte Webservices werden bevorzugt eingesetzt, um die Funktionalität von Legacy-Anwendungen zu kapseln [LeRo02b]. Der dynamische Aufruf von Webservices erfolgt unter Berücksichtigung deren Dienstgüteeigenschaften und wird durch die *Proxy-Komponente* und die *Auswahl-Komponente* realisiert. Die *Proxy-Komponente* nimmt Web-Service-Aufrufe des Geschäftsprozesses entgegen und leitet sie an die *Auswahl-Komponente* weiter. Diese wählt denjenigen Webservice aus, der entsprechend der vorgegebenen Kriterien die beste Bewertung erhalten hat und führt ihn aus. Wird ein Webservice von der *Auswahl-Komponente* aufgerufen, so protokolliert die *Accounting-Komponente*, wann der Service gestartet und wie lange er genutzt wird. Auftretende Fehler werden ebenfalls dokumentiert. Die *Accounting-Komponente* erfasst hierzu Informationen und Messwerte in einer Datenbank, wie z. B. Name des Providers, Name des Webservices, Beginn und Ende der Web-Service-Nutzung und eventuell auftretende Fehlermeldungen. Des Weiteren können die im Rahmen des Accounting gewonnen Daten als Grundlage verwendet werden, um Aussagen über die Dienstgüteeigenschaften des Webservice, wie z. B. seine Verfügbarkeit, zu treffen.

Die *QoS-Monitoring-Komponente* überwacht die in der SLA zugesicherten Dienstgüteeigenschaften. Hierzu wertet sie die von der *Accounting-Komponente* durchgeführten Messungen aus. Mögliche Verletzungen der SLA werden an den Provider gemeldet. Zudem kann die *QoS-Monitoring-Komponente* veranlassen, einen Webservice, dessen Dienstgüte abfällt, durch einen anderen Webservice gleicher Funktionalität zu ersetzen. Hierzu sendet sie eine Nachricht an die *Auswahl-Komponente*, welche die Änderung durchführen kann. So kann z. B. ein Web-Service zur Einholung eines Kunden-Ratings, dessen Verfügbarkeit von der *QoS-Monitoring-Komponente* überwacht wird, durch einen anderen Web-Service mit gleicher Funktionalität ersetzt werden, sobald seine Verfügbarkeit z. B. von 99,5 % auf 99,0 % absinkt, selbst wenn dies nicht mit einer Verletzung des SLA einhergeht.

4 Umsetzung

Die in diesem Abschnitt vorgestellten Komponenten der Web-Service-Architektur sind in Java implementiert. Als Applikationsserver kommt Apache Tomcat 5.0 zum Einsatz. Apache Axis 1.1 wird als SOAP Engine verwendet. MySQL dient als Datenbankserver.

Zur Modellierung von SLAs wird das von IBM Research entwickelte *Webservice Level Agreement (WSLA)-Framework* [DaLP03; KeLu02] eingesetzt. WSLA basiert auf XML-Schema [W3C04b] und ist in drei Teile gegliedert: Im Abschnitt *Parties* werden die beteiligten Organisationen beschrieben. Die relevanten Parameter sowie die Art und Weise ihrer Berechnung werden im Abschnitt *Service Descriptions* geregelt.

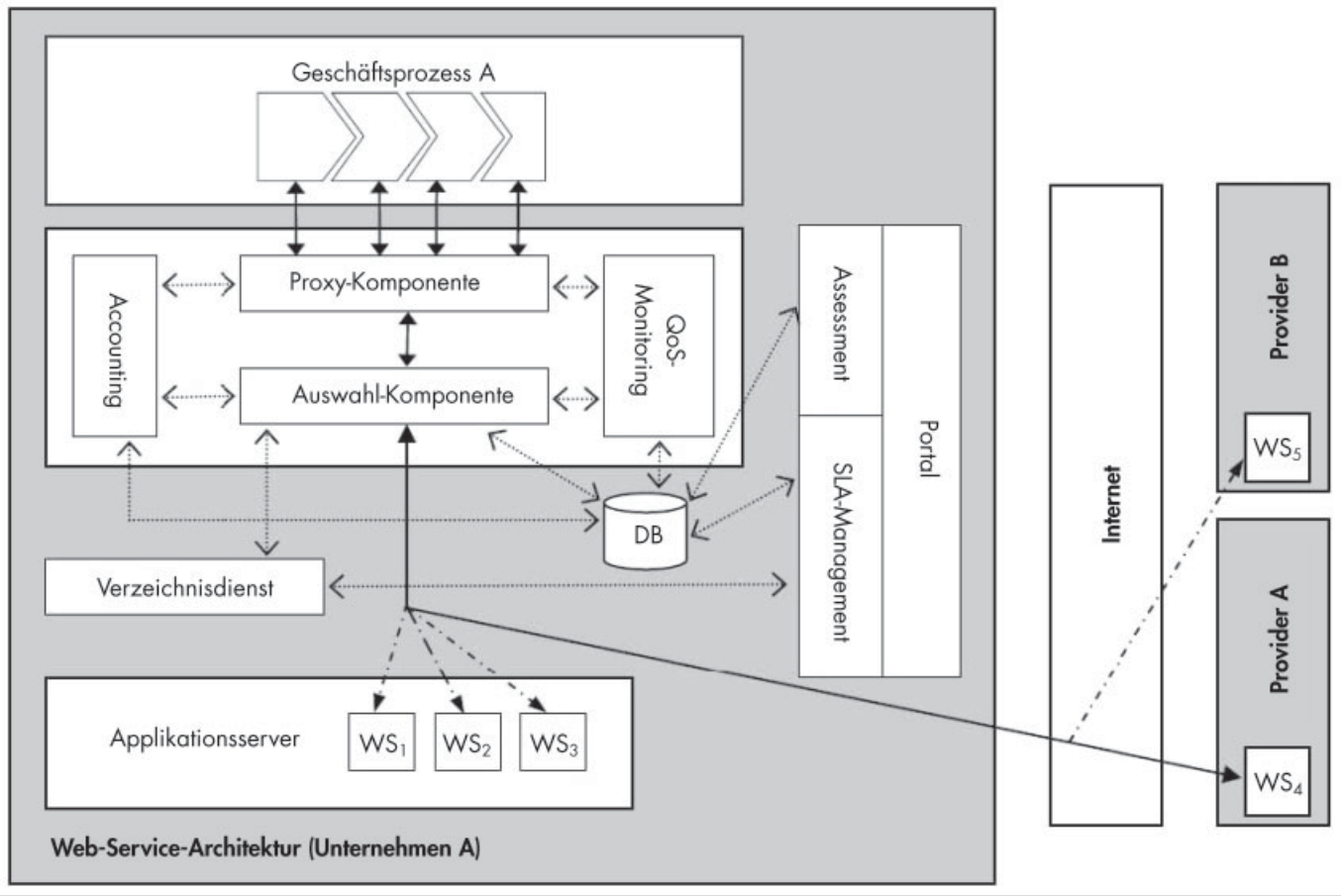

Bild 6 Architekturübersicht

Im Abschnitt *Obligations* werden die vom Provider einzuhaltenden Kriterien durch SLOs definiert. Das WSLA-Framework hat sich für unsere Arbeiten als geeignet erwiesen, da es eine einfache Modellierung der SLAs erlaubt und eine automatisierte Auswertung der SLAs leicht realisiert werden kann. Zudem ermöglicht das WSLA-Framework eine flexible Anpassung der von Service-Provider und Service-Nutzer vereinbarten Kriterien.

Bild 7 zeigt einen Auszug eines mit WSLA modellierten SLA für einen Webservice zur Einholung eines Kunden-Rating. So wird eine Verfügbarkeit (*availability)* von 99.5 % zugesichert und die Gültigkeitsdauer (2004-08-01 bis 2004-08-31) definiert. Nach Registrierung der SLA am Portal des Nutzers wird das WSLA-Dokument von der *SLA-Management-Komponente* ausgewertet. Hierbei werden die Angaben zu den Webservices, wie z. B. Name des Providers, Name des Webservice, zugesicherten Dienstgüteeigenschaften und die Gültigkeitsdauer, extrahiert und der Datenbank abgelegt. Die Bewertung der weichen Kriterien erfolgt durch einen IT-Verantwortlichen. Hierzu füllt dieser via Web-Front-End die in Kapitel 2 vorgestellte Bewertungsmatrix aus, deren Inhalt ebenfalls in der Datenbank abgelegt wird.

Für die Komposition der Webservices zu Geschäftsprozessen wird BPEL4WS (*Business Process Execution Language for Webservices*) [ACDG03] verwendet, da sich diese Spezifikation zu einem de facto Standard etabliert hat [OASI03]. Bei BPEL4WS handelt es sich um eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen und deren Interaktion auf Grundlage der Web-Service-Technologie. Konzeptionell betrachtet, entstehen mit BPEL4WS modellierte Geschäftsprozesse durch die Komposition verschiedener Webservices. Zur Ausführung der mit BPEL4WS modellierten Geschäftsprozesse wird die BPEL4WS-Engine von IBM AlphaWorks [IBM04] eingesetzt. Bild 8 zeigt den mit ^BPEL4WS modellierten Aufruf eines Webservice zur Einholung eines Kunden-Rating. Dieser Aufruf wird durch das <invoke>-Konstrukt modelliert.

In unserer Web-Service-Architektur ruft die BPEL4WS Engine allerdings nicht direkt einen

Rating-Webservice auf, sondern stattdessen die *Proxy-Komponente*. Um dies zu erreichen, wird lediglich bei der Aufnahme der Webservices in das interne Verzeichnis deren WSDL-Datei so modifiziert, dass die physikalische Adresse des eigentlichen Webservice (z. B. http://www.rating-company-a.de/web/servcices/rating) durch die Adresse der *Proxy-Komponente* (z. B. http://localhost:8081/axis/WSproxy) ersetzt wird. Diese Modifizierung der physikalischen Web-Service-Adresse wird durch die *SLA-Management-Komponente* vorgenommen. Die *Proxy-Komponente* fungiert als ein Server und wartet mit einem Daemon-Thread (z. B. auf Port 8081) auf eingehende Web-Service-Aufrufe, die in

Form von http-Requests eintreffen. Diese enthalten die eigentliche SOAP-Nachricht zum Aufruf des entsprechenden Webservice. Geht ein Web-Service-Aufruf auf dem entsprechenden Port ein, so startet der Daemon-Thread einen neuen Client-Thread, um den eingehenden Aufruf der BPEL4WS-Engine abzuarbeiten. Der Client-Thread liest zunächst die als http-Request eingehenden Daten (u. a. Web-Service-URL und die SOAP-Nachricht) ein und legt diese in einem hierfür generierten Request-Objekt ab. Bild 9 zeigt den Ausschnitt eines http-Request zum Aufruf des Rating-Web-Service mit der URL „/wsproxy/rating“ und den SOAP-Parametern „_surname“ und „_firstname“.

Das Request-Objekt wird vom Client-Thread der *Proxy-Komponente* an die *Auswahl-Komponente* weitergegeben. Die Aufgabe der *Auswahl-Komponente* besteht darin, den bezüglich der Bewertungskriterien optimalen Webservice auszuwählen und diesen aufzurufen. Dieser Vorgang läuft in mehreren Schritten ab:

- Im internen Verzeichnis wird für die angefragte URL (z. B. /wsproxy/rating) die zugehörige Web-Service-Kategorie (z. B. kreditprozess.rating) ermittelt.
- Aus dieser Web-Service-Kategorie wird nun ein sog. Target generiert. Dieses Target wird in der Form <Protokoll>:<Web-Service-Kategorie>:<Suchkriterium, Nebenbedingung> angegeben. Soll nun der Rating-Web-Service mit der höchsten Gesamtbewertung ohne zusätzlich definierte Nebenbedingung aufgerufen werden, so sieht das Target wie folgt aus: <DB>:<kreditprozess.rating>:<maxBewertung, keineNB>
- Um nun den optimalen Webservice zu bestimmen, wird das *Target* an eine *SearchFactory* übergeben. Die *SearchFactory* liefert ein Suchobjekt entsprechend dem definierten Protokoll zurück. Das Protokoll ist abhängig vom Verzeichnisdienst, der die Webservices verwaltet. Aus Gründen der Einfachheit kann hierzu eine SQL-Datenbank verwendet werden. Unsere Lösung unterstützt zudem auch einen UDDI-Server als Verzeichnisdienst. In diesem Fall ist als Protokoll im *Target* „UDDI“ anzugeben.
- Nachdem die *Auswahl-Komponente* mittels des Suchobjekts die Adresse (Endpoint-URL) des optimalen Webservice ermittelt hat, sendet es die ursprünglich von der BPEL4WS-Engine erhaltene Anfrage an diese Endpoint-URL weiter und ruft damit den konkreten Webservice auf. Zugleich wird auch die *Accounting-Komponente* gestartet, die Informationen, u. a. die Aufrufzeit und die Bearbeitungsdauer, mitprotokolliert und in der Datenbank ablegt.
- Liefert der Webservice ein Ergebnis zurück, so wird dieses von der *Auswahl-Komponente* wieder an den Client-Thread gesendet. Dieser übermittelt das Ergebnis an die BPEL4WS-Engine. Damit ist der Web-Service-Aufruf komplett abgearbeitet und der Client-Thread wird beendet.
- Die *QoS-Monitoring-Komponente* überprüft nun aufgrund der protokollierten Aufrufdaten, ob es zu einer SLA-Verletzung gekommen ist. In diesem Falle wird eine Warnmeldung ausgegeben.

```
<ServiceDefinition name="RatingService">
      <ServiceLevelObjective name=”SLO_For_Availability”>
            <Obliged>Company</Obliged>
            <Validity>
                  <Start>2004-08-01</Start> <End>2004-08-31</End>
            </Validity>
            <Expression>
                  <Predicate type=”Greater”>
                        <SLAParameter>Availability</SLAParameter>
                        <Value>99.5</Value>
                  </Predicate>
            </Expression>
            <EvaluationEvent>NewValue</EvaluationEvent>
      </ServiceLevelObjective>
      …
</ServiceDefinition>
```

Bild 7 Auszug eines SLA für einen Webservice zur Einholung eines Kunden-Rating

```
<invoke   name="callRating"
          partner="Rating" portType="ns1:RatingWS"
          operation="validCustomer"
          inputVariable="ratingRequest"
          outputVariable="ratingResponse">
      …
</invoke>
```

Bild 8 Aufruf eines Web-Service durch den BPEL4WS-Prozess

```
POST /wsproxy/rating HTTP/1.0
SOAPAction: ""
User-Agent: Axis/1.1
Host: 192.168.12.200:8081
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="…" …>
   soapenv:Body>
      <m:validCustomer xmlns:m="…" …>
           <_surname xsi:type="xsd:string">Mustermann</_surname>
           <_firstname xsi:type="xsd:string">Peter</_firstname>
      </m:validCustomer>
   </soapenv:Body>
</soapenv:Envelope>
```

Bild 9 http-Request für den Aufruf eines Rating-Web-Service

Das in diesem Abschnitt beschriebene Zusammenwirken der beteiligten Architekturkomponenten wird in Bild 10 noch einmal in Form eines Kollaborationsdiagramms grafisch dargestellt. Auf weitere Details bezüglich der Implementierung kann an dieser Stelle nicht eingegangen werden, da dies den Rahmen des Beitrags sprengen würde. Hierzu sei auf [BeMS04b] verwiesen.

5 Zusammenfassung und Ausblick

Webservices haben als Technologie zur Realisierung flexibler, verteilter Geschäfts-

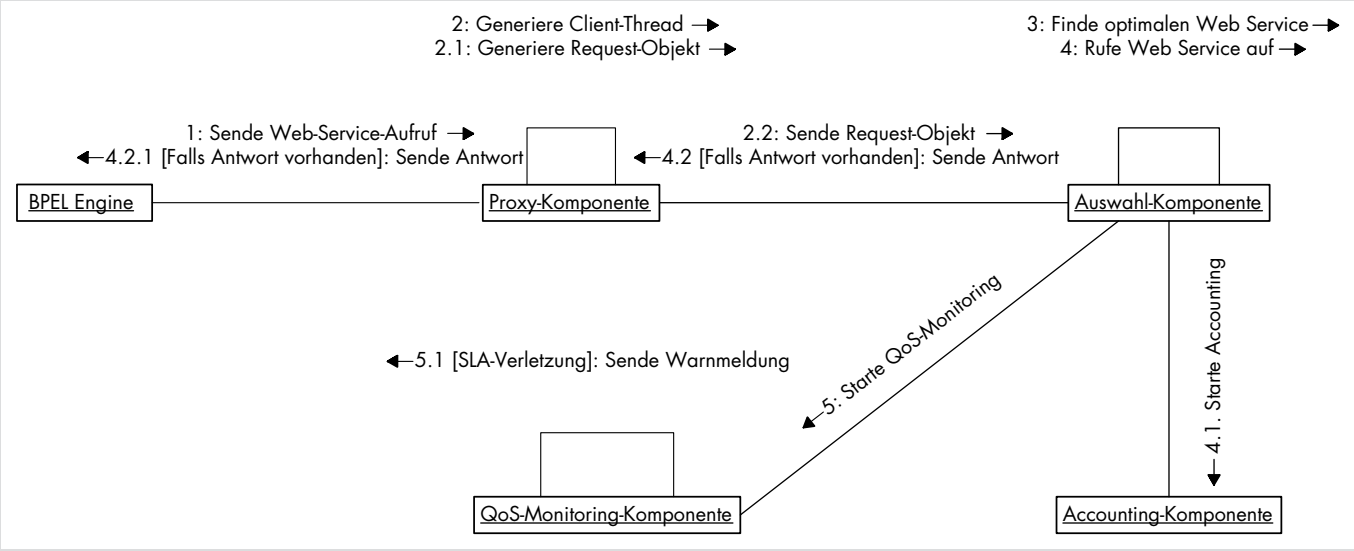

Bild 10 Interaktion der beteiligten Architekturkomponenten

prozesse im Zuge der Globalisierung und des BPO an Bedeutung gewonnen. In diesem Zusammenhang ist die Dienstgüte der Webservices von entscheidender Bedeutung, da Unternehmen nur bereit sind, externe Webservices in kritischen Geschäftsprozessen zu verwenden, wenn Garantien bezüglich der Dienstgüteeigenschaften, wie z. B. Verfügbarkeit, Antwortzeit und Durchsatz, vorliegen. In diesem Beitrag wird daher eine von uns entwickelte, ganzheitliche Web-Service-Architektur und deren prototypische Implementierung beschrieben, die Dienstgüte im Rahmen flexibler Geschäftsprozesse umfassend unterstützt. So wird durch ein Lebenszykluskonzept sichergestellt, dass nur Webservices in Geschäftsprozessen zum Einsatz kommen können, deren Dienstgüteeigenschaften den Anforderungen des Nutzers entsprechen. Die Dienstgüteeigenschaften externer Webservices müssen durch SLAs zugesichert werden. Unsere Architektur unterstützt des Weiteren die dynamische Auswahl von Webservices unter Berücksichtigung ihrer Dienstgüteeigenschaften. Die Einhaltung der SLAs wird ebenfalls durch das System zur Laufzeit überwacht.

Ein Schwerpunkt unserer zukünftigen Forschungsaktivität liegt neben der weiteren Evaluation der hier vorgestellten Architektur im Rahmen des *E-Finance Lab* auf der Erforschung sog. *selbst-konfigurierbarer Geschäftsprozesse*. Hierunter versteht man Geschäftsprozesse, welche die zur Erreichung eines bestimmten Geschäftsziels notwendigen Prozessbausteine selbständig in Abhängigkeit des Kontextes zusammenstellen.

Anmerkungen

[1] In der Betriebswirtschaftslehre versteht man unter dem Lebenszyklus Gesetzmäßigkeiten bezüglich des Umsatzverlaufes eines Produktes während einer als begrenzt angenommenen Lebensdauer [ThAc01, 166]. In unserem Beitrag beschreibt das Lebenszykluskonzept verschiedene Phasen, die ein Webservice während seines Einsatzes in einem Unternehmen durchläuft, nicht aber den Lebenszyklus des Webservices im klassischen betriebswirtschaftlichen Sinne.

Danksagung

Wir bedanken uns bei Herrn Michael Spahn, der uns bei der prototypischen Implementierung unterstützt hat. Unser weiterer Dank gilt unseren Partnern im E-Finance Lab für die kooperative Zusammenarbeit und die konstruktiven Diskussionen.

Literatur

[ACDG03] *Andrews, Tony; Curbera, Francisco; Dholakia, Hitesh; Goland, Yaron; Klein, Johannes; Leymann, Frank; Liu, Kevin; Roller, Dieter; Smith, Doug; Thatte, Satish; Trickovic, Ivana; Weerawarana, Sanjiva:* Business Process Execution Language for Webservices. Version 1.1, 05.05.2003. http://www.ibm.com/developerworks/library/ws-bpel, Abruf am 2004-10-26.

[ACKM03] *Alonso, Gustavo; Casati, Fabio; Kuno, Harumi; Machiraju; Vijay:* Web Services. Concepts, Architectures and Applications. Springer, Berlin 2003.

[Alon02] *Alonso, Gustavo:* Myths around Webservices. In: Bulletin of the Technical Committee on Data Engineering 25 (2002) 4, S. 3–9.

[ATIS01] ATIS Committee. Accountability. http://www.atis.org/tg2k/_accountability.html, Abruf am 2004-11-10.

[BeKa03] *Becker, Jörg; Kahn, Dieter:* The Process in Focus: In: *Becker, Jörg; Kugeler, Martin; Rosemann, Michael (Hrsg.):* Process Management. A Guide for the Design of Business Processes. Springer, Berlin u. a. 2003, S. 1–12.

[BeMS04a] *Berbner, Rainer; Mauthe, Andreas; Steinmetz, Ralf:* Unterstützung dynamischer E-Finance-Geschäftsprozesse. In: *Horster, P. (Hrsg.):* Elektronische Geschäftsprozesse 2004. Klagenfurt, Österreich. Syssec Verlag 2004, S. 44–54.

[BeMS04b] *Berbner, Rainer; Mauthe, Andreas; Steinmetz, Ralf:* Eine Webservice basierte Workflow Architektur. Technical Report KOM 07/2004. TU Darmstadt. Oktober 2004.

[BuKu03] *Buhl, Hans-Ulrich; Kundisch, Dennis:* Transformation von Finanzintermediären durch IT. In: WIRTSCHAFTSINFORMATIK 45 (2003) 5, S. 503–508.

[CaSh02] *Cardoso, Jorge; Sheth, Amit:* Semantic e-Workflow Composition. Technical Report 02-004. LSDIS Lab, Computer Science Department. University of Georgia, Athens, USA Juli 2002.

[CaSM02] *Cardoso, Jorge; Sheth, Amit; Miller, John:* Workflow Quality of Service. In: Proceedings of the International Conference on Enter-

prise Integration and Modeling Technology and International Enterprise Modeling Conference (ICEIMT/IEMC 2002). Valencia, Spanien 2002, S. 303–311.

[DaLP03] *Dan, Asit; Ludwig, Heiko; Pacifici, Giovanni:* Webservices Differentiation with Service Level Agreements. IBM developerworks. Mai 2003. http://www-106.ibm.com/developerworks/webservices/library/ws-slafram, Abruf am 2004-10-27.

[EbFi03] *Eberhart, Andreas; Fischer, Stefan:* Webservices. Grundlagen und praktische Umsetzung mit J2EE und .NET. Carl-Hanser-Verlag München, Wien 2003.

[FeBu02] *Fensel, Dieter; Bussler, Christoph:* The Webservice Modeling Framework WSMF. In: Electronic Commerce: Research and Applications 1 (2002) 2, S. 113–137.

[Frös99] *Fröschl, Friedrich:* Vom IuK-Outsourcing zum Business Process Outsourcing. In: WIRTSCHAFTSINFORMATIK 41 (1999) 5, S. 458–460.

[GoKP03] *Gouscos, Dimitris; Kalikakis, Manolis; Georgiadis, Panagiotis:* An Approach to Modeling Webservice QoS and Provision Price. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISEW 2003). Rom, Italien 2003, S. 121–130.

[Heck04] *Heckmann, Oliver:* A System-oriented Approach to Efficiency and Quality of Service for Internet Service Providers. Dissertationsschrift, Technische Universität Darmstadt 2004.

[HeKö03] *Heinzl, Armin; König, Wolfgang:* Interview mit Dietrich Voigtländer zu „Industrialisierung im Bankgeschäft: Outsourcing und Standardsoftware". In: WIRTSCHAFTSINFORMATIK 45 (2003) 2, S. 143–146.

[IBM04] *IBM AlphaWorks:* IBM Business Process Execution Language for Webservices Java Run Time (BPWS4J). http://www.alphaworks.ibm.com/tech/bpws4j, Abruf am 2004-11-02.

[KaKL03] *Kalepu, Sravanthi; Krishnaswamy, Shonali; Loke, Seng Wai:* Verity: A QoS Metric for Selecting Webservices and Providers. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISEW 2003). Rom, Italien 2003, S. 131–139.

[KeLu02] *Keller, Alexander; Ludwig, Heiko:* The WSLA Framework: Specifying and Monitoring Service Level Agreements for Webservices. In: Journal of Network and Systems Management 11 (2003) 1, S. 57–81.

[Kell02] *Keller, Wolfgang*: Enterprise Application Integration. Erfahrungen aus der Praxis. dpunkt, Heidelberg 2002.

[KRLB01] *Koetzle, Laura; Rutstein, Charles; Liddell, Heather; Buss, Christian; Nakashima, Taichi:* Reducing Integration's Cost. Forrester Research Report. Juni 2001. http://www.forrester.com/go?docid=11981, Abruf am 2004-10-26.

[LeRo00] *Leymann, Frank; Roller, Dieter:* Production Workflow. Concepts and Techniques. Prentice Hall Upper Saddle River 2000.

[LeRo02a] *Leymann, Frank; Roller, Dieter:* Webservices: Business Processes in a Web Services World. A quick overview of BPEL4WS. IBM DeveloperWorks. http://www-106.ibm.com/developerworks/webservices/library/ws-bpelwp, Abruf am 2004-10-26.

[LeRo02b] *Leymann, Frank; Roller, Dieter:* Using flows in information integration. In: IBM Systems Journal 41 (2002) 4, S. 732–742.

[LiWi04] *Lindert, Frank; Wiedeler, Markus:* Organisationsübergreifendes Geschäftsprozessmanagement. In: IT Information Technology 46 (2004) 4, S. 175–183.

[MaNa02] *Mani, Anbazhagan; Nagarajan, Arun:* Understanding Quality of Service for Webservices. IBM DeveloperWorks. Januar 2002. http://www-106.ibm.com/developerworks/webservices/library/ws-quality.html, Abruf am 2004-10-27.

[MCLP03] *Malone, Thomas W.; Crowston, Kevin; Lee, Jintae; Pentland, Brian T.; Dellarocas, Chrysanthos; Wyner, George M.; Quimby, John; Bernstein, Abraham; Herman, George A.; Klein, Mark; Osborn, Charles S.; O'Donnell, Elisa:* Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. In: Management Science 45 (1999) 3, S. 425–443.

[OASI03] *OASIS:* OASIS Webservices Business Process Execution Language (WSBPEL), Technical Committee, 29.04.2003. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, Abruf am 2004-10-27.

[Papa03] *Papazoglou, Mike P.:* Service -Oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003). Rom, Italien 2003, S. 3–12.

[PrSc02] *Preuner, Günther; Schrefl, Michael:* Integration of Webservices into Workflows through a Multi-Level Schema Architecture. In: Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2002). Newport Beach, USA 2002, S. 51–60.

[Ran03] *Ran, Shuping:* A model for Webservices discovery with QoS. In: ACM SIGecom Exchanges 4 (2003) 1, S. 1–10.

[RFC01] *Request for Comment 3198:* Terminology for Policy-Based Management 2001. http://www.faqs.org/rfcs/rfc3198.html, Abruf am 2004-11-26.

[Ried03] *Riedl, Rene:* Begriffliche Grundlagen des Business Process Outsourcing. In: Information Management & Consulting 18 (2003) 3, S. 6–10.

[Schm01] *Schmitt, Jens B.:* Heterogeneous Network Quality of Service Systems. Kluwer Academic Publishers 2001.

[Schm03] *Schmidt, Rainer:* Webservices Based Architectures to Support Dynamic Inter-organizational Business Processes. In: Proceedings of the International Conference on Webservices – Europe (ICWS-Europe 2003). Erfurt 2003, S. 123–136.

[ShAS03] *Sharma, Amit; Adarkar, Hemant; Sengupta, Shubhashis:* Managing QoS through Prioritization in Webservices. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISEW 2003). Rom, Italien 2003, S. 140–148.

[SMSV03] *Sivashanmugam, Kaarthik; Miller, John; Sheth, Amit; Verma, Kunal:* Framework for Semantic Web Process Composition. Technical Report 03-008. LSDIS Lab, Computer Science Department. University of Georgia, Athens, USA Juni 2003.

[StBM04] *Steinmetz, Ralf; Berbner, Rainer; Martinovic, Ivan:* Webservices zur Unterstützung flexibler Geschäftsprozesse in der Finanzwirtschaft. In: *Sokolovsky, Z.; Löschenkohl, S. (Hrsg.):* Industrialisierung der Finanzwirtschaft. Gabler Wiesbaden 2004.

[TGNR03] *Tian, Min; Gramm, Andreas; Naumowicz, Tomasz; Ritter, Hartmut; Schiller, Jochen:* A concept for QoS Integration in Webservices. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISEW 2003). Rom, Italien 2003, S. 149–155.

[ThAc01] *Thommen, Jean-Paul; Achleitner, Ann-Kristin:* Allgemeine Betriebswirtschaftslehre.

Abstract

A QoS-enabled Webservice Architecture to Support Flexible Business Processes

Webservices as a technology to enable distributed business processes gain in importance. However, the support of Quality of Service (QoS) is crucial in this context. Thus, we present an integrated Webservice architecture with comprehensive QoS support. The architecture we introduce in this paper supports the assessment of Webservices to assure that only Webservices will be used in critical business processes that satisfy the requirements defined by the user. The selection and execution of a certain Webservice depends on its QoS-properties that must be guaranteed in *Service Level Agreements (SLAs)*. The compliance with SLAs is monitored by the system as well. Furthermore we introduce a prototypical implementation of our Webservice architecture.

Keywords: Webservice Architecture, QoS-enabled Webservices, Lifecycle Management for Webservices, Service Level Agreement (SLA)

Umfassende Einführung aus managementorientierter Sicht. 3. Aufl., Gabler, Wiesbaden 2001.

[UDDI03] *UDDI Spec Technical Committee:* UDDI Version 3.0.1. Technical Committee Specification, 14.10.2003. http://uddi.org/pubs/uddi_v3.htm, Abruf am 2004-10-26.

[W3C01] *W3C*: Webservices Description Language (WSDL) 1.1. W3C Note 15.03.2001. http://www.w3.org/TR/wsdl20, Abruf am 2004-10-26.

[W3C03] *W3C:* SOAP Version 1.2. W3C Recommendation 24.06.2003. http://www.w3.org/TR/soap12/, Abruf am 2004-10-26.

[W3C04a] *W3C:* Webservices Architecture Requirements. W3C Working Group Note 11.02.2004. http://www.w3.org/TR/wsa-reqs/, Abruf am 2004-10-28.

[W3C04b] *W3C:* XML Schema Part 0: Primer Second Edition. 28.10.2004. http://www.w3.org/TR/xmlschema-0/, Abruf am 2004-11-08.

[ZBDK03] *Zeng, Liangzhao; Benatallah, Boualem; Dumas, Marlon; Kalagnanam, Jayant; Sheng, Quan Z.:* Quality Driven Webservices Composition. In: Proceedings of the 12th International Conference on World Wide Web. Budapest, Ungarn 2003, S. 411–421.