

Akzeptierter Beitrag:

Konferenz: „Elektronische Geschäftsprozesse 2004“, Klagenfurt, Österreich

Unterstützung dynamischer E-Finance-Geschäftsprozesse

Rainer Berbner, Andreas Mauthe, Ralf Steinmetz

Multimedia Communications Lab KOM
Technische Universität Darmstadt
Merckstraße 25
64283 Darmstadt
Deutschland

[berbner|mauthe|steinmetz]@kom.tu-darmstadt.de

Zusammenfassung

Unternehmen müssen ihre Geschäftsprozesse flexibel adaptieren können, um Produkte und Dienstleistungen den anspruchsvoll gewordenen Kunden bedarfsgerecht “on demand” anbieten zu können. Der folgende Beitrag untersucht, wie flexible Geschäftsprozesse optimal durch die zugrunde liegende IT-Infrastruktur unterstützt werden können. Betrachtet wird in diesem Zusammenhang auch die IT-Unterstützung für eine Auslagerung von Prozessschritten (sog. Prozessbausteine) an externe Service Provider.

Der Beitrag beschreibt die prototypische Implementierung dieses Ansatzes auf Grundlage der Web Service Technologie. Als Referenzprozess aus dem Finanzsektor dient ein generisch gehaltener Kreditprozess. Zur Modellierung dieses Kreditprozesses wird BPEL4WS (*Business Process Execution Language for Web Services*) verwendet. Die semantische Beschreibung der Web Services wird mit OWL-S (*Ontology Web Language for Services*) realisiert.

Des Weiteren wird in diesem Beitrag ein “Integration Lifecycle” vorgestellt, der auch die Integration von extern bereitgestellten Web Services in den Workflow berücksichtigt.

Der in diesem Beitrag vorgestellte Ansatz wird im Rahmen des E-Finance Lab entwickelt. Das E-Finance Lab ist ein Zusammenschluss der Universität Frankfurt, der Technischen Universität Darmstadt sowie mehreren Industriepartnern¹.

1 Einführung

Globalisierung und Deregulierung der Märkte haben zu einem immensen Kostendruck und einem ständig steigenden Wettbewerb geführt. Um in diesem Wettbewerb bestehen zu können, müssen Unternehmen in der Lage sein, ihre Produkte und Dienstleistungen den anspruchsvoll gewordenen Kunden bedarfsgerecht, d.h. “on demand”, liefern zu können. Hierzu sind flexible Geschäftsprozesse eine Grundvoraussetzung. Der Erfolg eines Unternehmens ist eng verknüpft mit der Leistungsfähigkeit und Flexibilität der Geschäftsprozesse [LeRo00].

¹ <http://www.efinancelab.de>

Um Kosten zu senken und flexibel auf Änderungen reagieren zu können, versuchen viele Unternehmen, sich auf die eigene Kernkompetenz zu konzentrieren und Bereiche außerhalb dieser Kernkompetenz auszulagern (Outsourcing). Nachdem ursprünglich hauptsächlich IT-Systeme im Fokus der Outsourcing-Aktivitäten standen, gehen viele Unternehmen mittlerweile dazu über, auch komplette Geschäftsprozesse an externe Service Provider zu vergeben [Kat+04]. Diese Entwicklung bezeichnet man auch als Business Process Outsourcing (Abkürzung: BPO). Jedoch ist diese Vorgehensweise nicht gänzlich unumstritten. Schließlich besteht für das Unternehmen die Gefahr, sich in Abhängigkeit eines Service Providers zu begeben. Zudem wird der Koordinationsaufwand zur Kontrolle des Service Providers häufig unterschätzt [Kat+04].

Im Rahmen des E-Finance Lab untersucht ein Forscherteam unter Leitung von Prof. Steinmetz, wie flexible Geschäftsprozesse effektiv durch die zugrunde liegende IT-Systeme und IT-Infrastruktur unterstützt werden können. In diesem Zusammenhang wird evaluiert, wie Geschäftsprozesse in Teile (sog. Prozessbausteine) zerlegt werden und wie diese Prozessbausteine implementiert werden können. Ein besonderes Augenmerk liegt hierbei auf der Fragestellung, wie Legacy-Anwendungen eingebunden und die Integration externer Dienstleistungen unterstützt werden können. Das E-Finance Lab ist eine gemeinsame Initiative der Universität Frankfurt, der Technischen Universität Darmstadt sowie von privaten Unternehmen, wie z.B. Deutsche Bank, Postbank, Accenture, IBM, Microsoft, Siemens, und T-Systems. Ziel dieser Initiative ist es, die Industrialisierung des Finanzgewerbes voranzutreiben.

Der Beitrag ist wie folgt gegliedert: Der dieser Arbeit zugrunde liegende konzeptionelle Ansatz wird in Kapitel 2 vorgestellt. In Kapitel 3 werden mit Web Services eine Technologie beschrieben, die zur Implementierung des Ansatzes verwendet wird. Die Erfahrungen bei der prototypischen Umsetzung mit Web Services werden in Kapitel 4 anhand eines generischen Kreditprozesses dargestellt. In Kapitel 5 wird ein Ausblick auf weitere Forschungsvorhaben im Rahmen des E-Finance Lab gegeben.

2 Konzeptionelle Ansatz

In dieser Abhandlung wird ein Ansatz vorgestellt und untersucht, der die Abbildung bestimmter Teile eines Geschäftsprozesses (Prozessbausteine, Abb. 1) auf sog. Services erlaubt. Die Funktionalität der abstrakten Prozessbausteine wird durch sog. Services implementiert. Unter einem Service wird hierbei eine abgeschlossene, unabhängige Komponente verstanden, die eine klar definierte Funktionalität anbietet [YaPH02]. Ein Service abstrahiert von zugrunde liegenden Entitäten, Objekten und Klassen. Die Unabhängigkeit eines Services ermöglicht seine Wiederverwendbarkeit in verschiedenen Zusammenhängen.

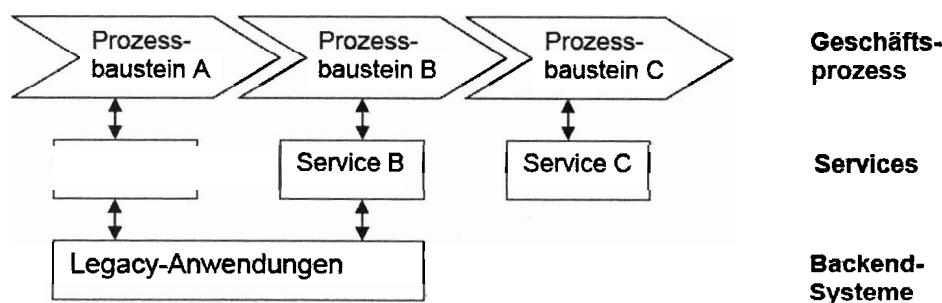


Abb. 1: Abbildung von Prozessbausteinen auf Services

Services können die Funktionalität der Legacy-Anwendungen “wrappen”, d.h. die Funktionalität der Legacy-Anwendungen wird durch Web Services gekapselt. Aufgrund dieser Modularisierung der Geschäftsprozesse können Prozessbausteine (bzw. Services) auch von externen Providern bereitgestellt werden.

Der Geschäftsprozess als solcher kann daher als eine Folge von Service-Aufrufen verstanden werden (Abb. 1). Die Steuerung des Geschäftsprozesses wird durch eine Workflow Engine vorgenommen.

Eine auf Services als Grundbausteinen basierende Architektur wird auch als *Service-oriented Architecture (SoA)* bezeichnet [Nati03]. Charakteristisch für eine SoA ist das Prinzip der “Lösen Kopplung”. Unter Kopplung wird der Grad der Abhängigkeit zwischen den verwendeten Services verstanden. Lose gekoppelte Systeme sind zeitunabhängig, d.h. sie können sowohl synchron als auch asynchron aufgerufen werden. Durch Verzeichnisdienste wird zudem die Transparenz des Speicherortes erreicht, so ist gewährleistet, dass Services dynamisch gefunden und genutzt werden können [Hans03]. Lose gekoppelte Systeme haben den Vorteil, dass Services sehr leicht durch andere Services ersetzt werden können.

Der Vorteil dieses Ansatzes besteht darin, dass durch das Beziehen einzelner Prozessbausteine von externen Providern eine Kostenreduktion erreicht werden kann, während die Steuerung des Prozesses im Unternehmen verbleibt. Dieses Konzept gewährleistet somit ein Höchstmaß an Flexibilität und stellt gleichzeitig sicher, dass die Abhängigkeit von externen Service Providern auf ein Minimum begrenzt wird. Der in diesem Beitrag postulierte Ansatz ist somit weitaus flexibler als ein traditionelles Business Process Outsourcing (BPO), das auf die Auslagerung eines vollständigen Geschäftsprozesses an externe Dienstleister abzielt.

3 Web Services als Realisierung einer Service-oriented Architecture (SoA)

Zur prototypischen Implementierung des in Abb. 1 vorgestellten Ansatzes werden Web Services verwendet. Die Web Service Technologie ist eine Realisierung des bereits in Kapitel 2 beschriebenen SoA-Konzeptes.

Web Services sind XML-basierte Softwarekomponenten, die über ein Netzwerk (meist das Internet) aufrufbar und ausführbar sind [Boo+04]. Hierbei wird zwischen der eigentlichen Implementierung des Web Service sowie seiner Servicebeschreibung unterschieden. Die Servicebeschreibung macht Aussagen zu den Schnittstellen des Web Service bzw. zur angebotenen Funktionalität. Web Services basieren auf offenen Standards und sind unabhängig von Plattformen, Betriebssystemen und Programmiersprachen. Dies hat zudem auch die Unabhängigkeit von Herstellern zur Folge.

Web Services implementieren die Funktionalität von Prozessbausteinen (Abb. 1). Die verwendeten Web Services können auch von externen Providern bereitgestellt werden.

Die Komposition der einzelnen Services zu Geschäftsprozessen wird mit BPEL4WS (*Business Process Execution Language*) realisiert [And+03].

3.1 Gründe für den Einsatz von Web Services

Im Wesentlichen sind die folgenden beiden Gründe für den Einsatz der SoA-basierten Web Service Technologie ausschlaggebend:

- *Einfache Integration von Legacy-Anwendungen*

Die flexible Integration externer Service Provider stellt hohe Anforderungen an die zugrunde liegende IT-Infrastruktur, gerade was die Kopplung der IT-Systeme des Outsources mit denen des externen Service Providers betrifft. Für viele Unternehmen, vor allem im Finanzsektor, erweist sich die Heterogenität und Komplexität der IT-Landschaft, die von einer Vielzahl von Legacy-Anwendungen geprägt ist, als große Herausforderung [StBM04].

Bei dem in diesem Beitrag vorgestellten Ansatz entfallen weitgehend sowohl die aufwendige Veränderung der Legacy-Anwendungen als auch die Konfiguration proprietärer EAI-Lösungen. Web Services erlauben die Kapselung der Funktionalität von Legacy-Systemen. Die Erstellung innovativer Dienstleistungen bzw. (Bank-)Produkte kann im Idealfall durch die Integration zusätzlicher Web Services in den Geschäftsprozess bzw. durch eine veränderte Orchestrierung neuer und bereits bestehender Web Services erfolgen [StBM04]. Änderungen an einem mit BPEL4WS modellierten Geschäftsprozess haben keine Auswirkungen auf die zugrunde liegende IT-Landschaft.

- *Auslagern einzelner Prozessschritte an externe Service Provider*

Bei Auswahl der Prozessbausteine ist das Unternehmen aufgrund des Prinzips der "Losen Kopplung" nicht auf Eigenentwicklungen angewiesen. Es können sehr einfach Services externer Provider integriert werden. Ziel sollten Dienstleistungen bzw. (Bank-) Produkte sein, die modular aus Bausteinen kombinierbar sind [Köni02].

3.2 Offene Punkte im Web Service-Umfeld

Bei der praktischen Umsetzung auf Grundlage der Web Service Technologie ist jedoch deutlich geworden, dass deren Einsatz in Produktivumgebungen aufgrund unausgereifter Spezifikationen nicht ohne weiteres möglich ist. In diesem Zusammenhang wurden folgende wesentlichen Schwachpunkte identifiziert:

- *Fehlende semantische Beschreibung der Web Services*

Als Defizit erweist sich die nur unzureichende Beschreibung von Web Services. So können mit WSDL (*Web Service Description Language*) lediglich syntaktisch die Operationen und Methoden des Web Service beschrieben werden². Auch die in den UDDI-Verzeichnisdiensten³ verfügbaren Informationen sind in vielen Fällen nicht ausreichend. So können zwar Informationen über den Service Provider (White Pages) und eine grobe inhaltliche Kategorisierung der Funktionalität (Yellow Pages) verwaltet werden, diese sind jedoch für eine umfassende Beschreibung des Web Service nicht ausreichend. Unternehmen werden Web Services externer Provider nur dann verwenden, wenn garantierte Aussagen zur Dienstgüte des Service verfügbar sind. Hierzu ist aber eine semantische Beschreibung der Web Services unabdingbar. In Kapitel 4.2 wird gezeigt, wie Web Service-Metadaten mit Hilfe von OWL-S (*Ontology Web Language for Services*) verwaltet werden.

² Auf WSDL kann im Folgenden nicht weiter eingegangen werden, da dies den Rahmen dieses Beitrags sprengen würde. Die WSDL-Spezifikation ist unter <http://www.w3.org/TR/wsdl> abrufbar.

³ UDDI (*Universal Description, Discovery and Integration*) ist ein Standard zum Finden und Veröffentlichen von Web Services (<http://www.uddi.org/specification.html>).

- *Fehlende Unterstützung eines Lifecycle-Managements zur Integration der verwendeten Web Services*

Die Integration von Web Services in den Workflow ist ein komplexer Vorgang, der in mehrere Phasen gegliedert werden sollte. In Kapitel 4.3 wird ein Integration Lifecycle vorgestellt, der auch in am Lehrstuhl von Prof. Steinmetz entwickelten Prototypen Eingang findet.

4 Evaluierung des Ansatzes anhand eines generischen Kreditprozesses

Im Rahmen des E-Finance Lab sowie in Kooperation mit anderen Finanzdienstleistern wurde der Kreditprozess als ein Referenzprozess aus dem Finanzsektor identifiziert.

Der im Folgenden verwendete, generische Kreditprozess ist auf einige wesentliche Prozessschritte reduziert und dadurch stark vereinfacht. Zum einen wäre die Modellierung und prototypische Realisierung eines vollständigen Kreditprozesses zu umfangreich. Zum anderen ist es für die Evaluierung des in diesem Beitrag vorgestellten Ansatzes vollkommen ausreichend, sich auf einige, exemplarische Prozessschritte zu beschränken. Zudem unterliegen Details zur konkreten Ausgestaltung der Geschäftsprozesse einer gewissen Geheimhaltung, so dass Banken kein Interesse daran haben, diese zu veröffentlichen. Der im Folgenden verwendete generische Kreditprozess basiert jedoch auf der Analyse verschiedener Kreditprozesse in deutschen Finanzinstituten.

Initiiert wird der dem Prototyp zugrunde liegende generische Kreditprozess durch die Kreditanfrage seitens des Kunden (Abb. 2).

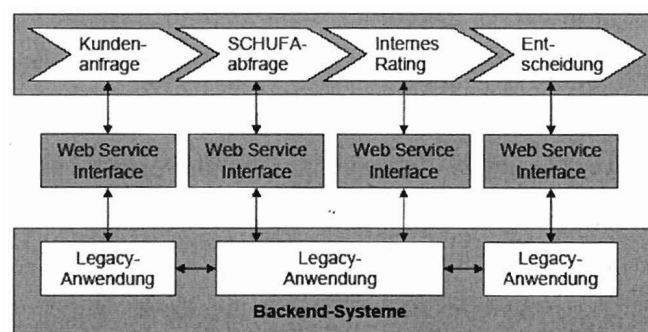


Abb. 2: Realisierung eines Kreditprozess mit Web Services

Hierbei ist es unerheblich, ob der Kunde die Anfrage via Internet selbst startet oder ob ein Sacharbeiter die Kundendaten in das System eingibt. Im Anschluss daran wird die Kreditwürdigkeit des Kunden geprüft. Hierzu wird das Einholen einer SCHUFA-Auskunft über den Antragsteller nachgebildet und ein Rating anhand interner Kriterien erstellt. Auf Grundlage dieser Informationen wird die Entscheidung über die Kreditvergabe getroffen. Dem Kunden wird diese dann abschließend mitgeteilt.

4.1 Orchestrierung der Web Services mit BPEL4WS

BPEL4WS (*Business Process Execution Language for Web Service*) ist eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen und deren Interaktion auf Grundlage der

Web Service Technologie. Die BEPL4WS-Spezifikation [And+03] wurde im August 2002 von Microsoft, IBM und BEA vorgestellt. Im April 2003 wurde zudem eine OASIS-Arbeitsgruppe eingerichtet, die die weitere Ausarbeitung und Standardisierung von BPEL4WS vorantreiben soll [OASI03]. Es ist daher wahrscheinlich, dass BPEL4WS sich als Standard für die Komposition von Web Services zu Geschäftsprozessen etablieren wird.

Konzeptionell betrachtet, entstehen mit BPEL4WS modellierte Geschäftsprozesse durch die Komposition verschiedener Web Services [LeRo02].

Bei den mit BPEL4WS modellierten Geschäftsprozessen kann es sich sowohl um unternehmensinterne als auch um Unternehmensgrenzen überschreitende, verteilte Geschäftsprozesse handeln. Die BPEL4WS-Spezifikation sowie die einzelnen, zur Anwendung kommenden Sprachkonstrukte können nicht im Detail vorgestellt werden, da dies den Rahmen des Beitrags sprengen würde.

Bei der prototypischen Realisierung des generischen Kreditprozesses werden zuerst die am Kreditprozess beteiligten Partner definiert (Abb. 3). Unter einem Partner versteht man Web Services, die durch das BPEL4WS-Skript aufgerufen werden, bzw. Web Services, die selbst den durch das BPEL4WS-Skript beschriebenen Geschäftsprozess aufrufen. Im Zusammenhang mit der Definition der Partner wird ebenfalls festgelegt, welche Rolle der Partner Web Service und welche Rolle der Prozess einnimmt. Hierbei ist wichtig zu erwähnen, dass Partner Web Services noch abstrakt zu verstehen sind, d.h. die Zuweisung zu einer konkreten Web Service Instanz erfolgt erst beim Deployment des Skriptes an der BPEL4WS-Engine.

Abb. 3 zeigt die Definition des Partners *schufaWebService*, der die Rolle *requestSchufa* übernimmt.

```
<partners>
  <partner name="creditCustomer"
    serviceLinkType="tns:creditApprovalLinkType"
    myRole="creditApprover"/>

  <partner name="schufaWebService"
    serviceLinkType="tns:schufaRequestLinkType"
    partnerRole="requestSchufa"/>

  ...
</partners>
```

Abb. 3: BPEL4WS-Partnerkonzept

Zudem wird, wie in Abb. 3 ersichtlich, jedem Partner noch ein *ServiceLinkType* zugewiesen. *ServiceLinkTypes* werden in der WSDL-Datei des Prozesses festgelegt und beschreiben die bilaterale Beziehung zwischen Partner Web Service und dem Geschäftsprozess. Des Weiteren wird für jede Rolle mit dem *portType* Element festgelegt, welche Funktionalität sie anbietet (Abb. 4).

```

<slnk:serviceLinkType name="schufaRequestLinkType">
  <slnk:role name="requestSchufa">
    <portType name="schufa:SchufaWS"/>
  </slnk:role>
</slnk:serviceLinkType>

```

Abb. 4: Auszug aus der WSDL-Datei des Geschäftsprozesses

Im Anschluss daran werden die vier identifizierten Prozessschritte als BPEL4WS-Skript modelliert. Die eigentlichen Prozessschritte (Prozessbausteine) werden durch Web Services implementiert und vom BPEL4WS-Skript aufgerufen. Hierzu wird das `<invoke>` Element verwendet. Abb. 5 zeigt den Aufruf eines Web Services zur Einholung der SCHUFA-Auskunft.

```

<invoke name="callSchufa"
  partner="schufaWebService"
  xmlns:ns16="http://localhost:8080/axis/services/SchufaServiceAgent"
  portType="ns16:SchufaWS"
  operation="validCustomer"
  inputVariable="schufaRequest"
  outputVariable="schufaResponse">
  ...
</invoke>

```

Abb. 5: Aufruf eines Web Service durch das `<invoke>` Element

Die Kooperation zwischen dem Geschäftsprozess und den Partner Web Services basiert auf dem Austausch von Nachrichten. Die Übergabe der Nachrichten wird durch das *inputVariable/outputVariable* Konstrukt realisiert. Das Format der Nachrichten wird in der WSDL-Beschreibung des Prozesses festgelegt. Wie in Abb. 6 zu sehen, können Nachrichten auch aus mehreren Teilen zusammengesetzt sein.

```

<message name="bankTransferRequestMessage">
  <part name="vorname" type="xsd:string"/>
  <part name="nachname" type="xsd:string"/>
  <part name="betrag" type="xsd:int"/>
  <part name="creditID" type="xsd:int"/>
</message>

```

Abb. 6: WSDL-Beschreibung einer Nachricht

4.2 Semantische Beschreibung mit OWL-S

Zur semantischen Beschreibung der Web Services wird OWL-S (*Ontology Web Language for Services*) verwendet. OWL-S stellt eine Ontologie zur Beschreibung von Web Services dar [OWLS03].

Um Inhalt und Fähigkeiten eines Services zu beschreiben, bietet OWL-S eine Vielzahl von Klassen und Eigenschaften an. Ziel von OWL-S ist es, Web Services maschineninterpretierbar zu machen. Dies erleichtert u.a. das automatische Suchen von Web Services und die Kontrolle von Dienstgüteeigenschaften der Web Services.

Die OWL-S Ontologie setzt sich aus drei Teilen zusammen: Das *ServiceModel* beschreibt die Arbeitsweise des Service. Wie auf den Service zugegriffen werden kann, wird durch das *ServiceGrounding* definiert. In diesem Zusammenhang werden Details, wie die verwendeten

Protokolle, Transportmechanismen und Nachrichtenformate, festgelegt. Für die prototypische Implementierung des generischen Kreditprozesses ist vor allem das *ServiceProfil* von Bedeutung, da es die Beschreibung der Eigenschaften eines Web Service ermöglicht.

In Abb. 7 zeigt einen stark vereinfachten Auszug aus einer Ontologie zur Beschreibung von Web Services, deren Aufgabe das Einholen einer Schufa-Auskunft ist. Mit Hilfe des *DatatypeProperty* Konstruktes wird die Ontologie um das Attribut Preis erweitert. Dieses gibt an, wie hoch der Preis zum Aufruf eines Web Service ist.

```
...
<owl:DatatypeProperty rdf:ID="price">
  ...
  <rdfs:domain rdf:resource="#Schufa" />
  <rdfs:range rdf:resource="xsd:integer"/>
  ...
</owl:DatatypeProperty>
...
```

Abb. 7: Ontologie zur Beschreibung der Schufa-Web Services

Abb. 8 zeigt einen Auszug aus der Instanz einer Schufa-Ontologie. Es wird der Preis für den hier beschriebenen Web Service zur Einholung der Schufa-Auskunft festgelegt.

```
...
<schufa: Schufa rdf:ID="Schufa">
  <price rdf:datatype="xsd:integer">5</price>
</schufa: Schufa>
...
```

Abb. 8: Instanz einer Schufa-Ontologie

4.3 Integration Lifecycle

Die Integration von Services in den Workflow ist ein höchst komplexer Vorgang, der bisher in der Literatur nur beschränkt betrachtet worden ist [Sin+03].

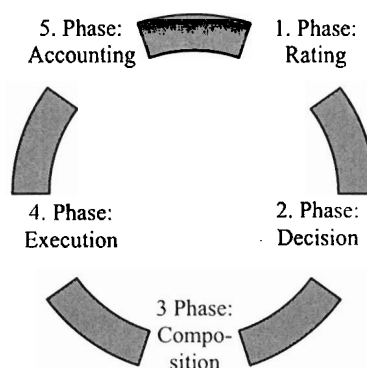


Abb. 9: Integration-Lifecycle

Sinnvoll ist daher ein Lifecycle-Management, das die periodische Integration von Services in den Workflow beschreibt. Es werden die folgende Phasen (Abb. 9) vorgeschlagen, die auch in den bereits beschriebenen Prototyp Eingang finden:

1. Phase: *Rating*

In der Rating-Phase wird der Tatsache Rechnung getragen, dass Unternehmen kaum bereit sind, externe Services als Prozessbausteine zu integrieren, deren Dienstgüte nicht bekannt ist. Wichtig in diesem Zusammenhang sind auch Referenzen, die ein Service Provider vorweisen kann. In Abb. 10 ist eine Bewertungstabelle für ein QoS-basiertes Rating von Prozessbausteinen vorgestellt.

		Bewertung
Gewichtung		(0-10 Punkte)
QoS-Kriterien	30%	
Latenzzeit	10%	7
Verfügbarkeit	10%	5
Performanz	10%	5
Sicherheit	0%	0
Preiskategorie	54%	6
Provider	16%	
externe Referenzen	4%	5
interne Erfahrungen	4%	2
persönlichen Ansprechpartner	4%	5
Vertragliche Ausgestaltung	4%	9
Gesamt	100%	6

Erklärung: 10 Punkte sehr gut

Abb. 10: Bewertungstabelle für Web Services

Als Beurteilungskriterien für ein QoS-basiertes Rating von Web Services können Zuverlässigkeit, Latenzzeit, Performance und Sicherheit herangezogen werden [MaNa02]. Bei dieser Vorgehensweise ist gewährleistet, dass nur Services in den Workflow integriert werden, die einem gewissen Qualitätsstandard genügen. Alle Services, die die vorgegebenen Ansprüche erfüllen, werden in einem internen Verzeichnis referenziert und können in den Workflow integriert werden. So kann beispielsweise definiert werden, dass nur Prozessbausteine mit einer bestimmten Mindestbewertung in das interne Verzeichnis aufgenommen werden.

2. Phase: *Decision*

In dieser Phase werden Kriterien definiert und gewichtet, die ausschlaggebend dafür sind, welche der zur Verfügung stehenden Prozessbausteine tatsächlich in den Workflow integriert werden.

Ziel ist hierbei, die Entscheidung (teil-)automatisiert zu realisieren. Ausgehend von den in Phase 1 vorgestellten Kriterien können bestimmte Entscheidungsregeln definiert werden. So wäre eine Entscheidungsregel denkbar, die nur Web Services berücksichtigt, deren Verfügbarkeit mit einem Wert größer 7 auf einer Skala von 0 bis 10 bewertet wurde.

Um Bewertungskriterien untereinander zu gewichten, bietet sich die Möglichkeit, diese Kriterien jeweils mit einem Faktor zu gewichten, der der Präferenz des Entscheidungsträgers entspricht. Dies kann beispielsweise durch folgende Funktion zum Ausdruck gebracht werden:

$$\begin{aligned}
 \text{Gesamtbewertung (WS)} = & \text{Faktor}_1 * \text{Bewertung}(\text{Latenzzeit (WS)}) \\
 & + \text{Faktor}_2 * \text{Bewertung}(\text{Verfügbarkeit(WS)}) \\
 & + \dots \\
 & + \text{Faktor}_n * \text{Bewertung}(\text{Sicherheit(WS)})
 \end{aligned}$$

3. Phase: *Composition*

In der Composition-Phase wird der ausgewählte Baustein technisch in den Workflow integriert. Bei dem hier beschriebenen Prototyp bedeutet dies, dass der selektierte Web Service an der BPEL4WS-Engine angemeldet wird.

4. Phase: *Execution*

Nach der Integration des ausgewählten Prozessbausteines kann der Workflow ausgeführt werden. Bei dem hier vorgestellten Prototyp hat dies zur Folge, dass eine Client-Anwendung den mit BPEL4WS modellierten Workflow aufruft.

5. Phase: *Accounting*

Werden Web Services externer Provider in den Workflow integriert, ist es für das Unternehmen wichtig zu wissen, wann und wie lange dieser Service genutzt werden. Nur so kann die Abrechnung des Service Providers kontrolliert werden. Zudem ist es für das Unternehmen hilfreich, nachvollziehen zu können, welche Organisationseinheiten den Service in Anspruch genommen haben. Dies ist vor allem für eine interne verursachungsgerechte Abrechnung von Bedeutung.

5 Ausblick

Der Beitrag beschreibt einen Ansatz, wie Teile eines Geschäftsprozesses (sog. Prozessbausteine) auf eine *Service-oriented Architecture* (SoA) abgebildet werden. Die prototypische Implementierung eines generischen Kreditprozesses basiert auf der Web Services Technologie. Der in diesem Beitrag propagierte Ansatz ermöglicht sowohl die Einbindung von Legacy-Anwendungen als auch Integration von Web Services externer Provider im Kontext des Business Process Outsourcing.

Im Rahmen des E-Finance Lab werden der in diesem Beitrag vorgestellte Ansatz und die prototypische Implementierung sukzessive erweitert. So wird eine proxy-basierte dynamische Integration von Web Services in den Workflow zur Laufzeit mit BPEL4WS realisiert. Dies hat den Vorteil, dass zur Laufzeit Web Services mit schlechter Performance deaktiviert und ersetzt werden können. Hierzu wird eine Proxy-Komponente implementiert, die neben der dynamischen Integration von Web Services auch noch deren QoS-Parameter misst und kontrolliert. In diesem Zusammenhang wird dann zudem überprüft werden, ob diese gemessenen Werte mit denen in der SLA (Service Level Agreement) übereinstimmen.

Das E-Finance Lab hat sich zum Ziel gesetzt, die Industrialisierung der Finanzbranche entscheidend voranzutreiben. Dieses Vorhaben durch die Evaluierung und Anwendung geeigneter Technologien und Architekturen zu unterstützen, ist Schwerpunkt der Forschungstätigkeit des Teams unter Leitung von Prof. Steinmetz. Im Mittelpunkt wird hierbei vor allem die Unterstützung organisationsübergreifender Prozesse stehen, die zu neuen dynamischen Kooperationsformen innerhalb des Finanzsektors (sog. *Financial Value Networks*) führen können.

Literatur

- [And+03] T. Andrews et al.: Specification - Business Process Execution Language for Web Services Version 1.1, 05.05.2003.
- [Boo+04] D. Booth et al.: Web Services Architecture, W3C Working Group Note (February 2004).
- [Hans03] J. Hanson: Vorteile lose gekoppelter Web-Services. Internet: <http://www.zdnet.de/builder/architect/0,39023548,20000793,00.htm> [Stand: 22.06.2004].
- [Kat+04] T. Katzensteiner et al.: Business Process Outsourcing: Grenze fließend. In: Handelsblatt, Beitrag vom 12.03.2004.
- [Köni02] W. König: Industrialisierung des Bankgeschäfts, WI-Editorial. In: Wirtschaftsinformatik 44 (2002) 6, S. 517–518.
- [Kreg01] H. Kregar: Web Services Conceptual Architecture (WSCA 1.0), IBM Software Group (Mai 2001).
- [LeRo00] F. Leymann, D. Roller: Production Workflow. Concepts and Techniques. Prentice Hall (2000).
- [LeRo02] F. Leymann, D. Roller: Business processes in a Web Services world, A quick overview of BPEL4WS, IBM (August 2002).
- [MaNa02] A. Mani, A. Nagarajan: Understanding quality of service for Web services. Improving the performance of your Web services, IBM (Januar 2002).
- [Nati03] Y.V. Natis: Service-Oriented Architecture Scenario, Gartner Group, AV-19-6751, 16.04.2003.
- [OASI03] OASIS Web Services Business Process Execution Language (WSBPEL), Technical Committee, 29.04.2003.
- [OWLS03] The OWL Services Coalition: OWL-S: Semantic Markup for Web Services, Technical White Paper, November 2003, Internet: <http://www.daml.org/services/owl-s/1.0/owl-s.html> [Stand: 28.06.2004].
- [Siv+03] K. Sivashanmugam et al.: Framework for Semantic Web Process Composition, Technical Report 03-008, LSDIS Lab, Computer Science Dept., UGA (June 2003).
- [StBM04] R. Steinmetz, R. Berbner, I. Martinovic: Web Services zur Unterstützung flexibler Geschäftsprozesse in der Finanzwirtschaft. Erscheint in: Z. Sokolovsky, S. Löschenkohl (Hrsg.): Industrialisierung der Finanzwirtschaft, Gabler-Verlag (2004).
- [WebS04] Web Services Architecture Working Group: Web Services Glossary, W3C Working Group Note 11.02.2004.
- [YaPH02] J. Yang, M.P. Papazoglou, W.J. van den Heuvel: Tackling the Challenges of Service Composition in E-Marketplaces. In: Proceedings of 12th International

Workshop on Research Issues in Data Engineering: Engineering
E-Commerce/E-Business Systems (RIDE'02).