

Porkert, Kurt

Research Report

Web-Services: Mehr als eine neue Illusion?

Beiträge der Fachhochschule Pforzheim, No. 104

Provided in Cooperation with:

Hochschule Pforzheim

Suggested Citation: Porkert, Kurt (2002) : Web-Services: Mehr als eine neue Illusion?, Beiträge der Fachhochschule Pforzheim, No. 104, Fachhochschule Pforzheim, Pforzheim, <https://nbn-resolving.de/urn:nbn:de:bsz:951-opus-808>

This Version is available at:

<https://hdl.handle.net/10419/97581>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de>

Fachhochschule
Pforzheim

Hochschule
für Gestaltung
Technik und
Wirtschaft



*Pforzheim University
of Applied Sciences*

**BEITRÄGE DER
HOCHSCHULE PFORZHEIM**

Kurt Porkert

**Web-Services – mehr als eine neue
Illusion?**

Nr. 104

Herausgeber: Ansgar Häfner, Norbert Jost, Karl-Heinz Rau,
Roland Scherr, Christa Wehner, Helmut Wienert
(geschäftsführend; wienert@fh-pforzheim.de)

Sekretariat: Alice Dobrinski
Fachhochschule Pforzheim,
Tiefenbronner Str. 65
75175 Pforzheim
dobrinski@vw.fh-pforzheim.de
Telefon: 07231/28-6201
Telefax: 07231/28-6666

Ausgabe: Mai 2002

Kurt Porkert

Web-Services – mehr als eine neue Illusion?

Prof. Dr. Kurt Porkert
Hochschule Pforzheim
Fachbereich 7 „Personalmanagement, Volkswirtschaftslehre
und Wirtschaftsinformatik“
Studiengang Wirtschaftsinformatik
porkert@fh-pforzheim.de

Kurt Porkert, geboren 1950 in Heiligenstadt (Eichsfeld), studierte und promovierte in Jena mit Studien- bzw. Forschungsschwerpunkten zur Auftragsbildung bei lagergesteuerter Fertigung (Diplomarbeit), zur Losgrößenoptimierung (Dissertation A) und zur Datenmodellierung für die Produktionsplanung und -steuerung (Dissertation B). Als EDV-Organisator bei Carl Zeiss Jena beschäftigte er sich nach der Promotion mit der Software- und Datenbankentwicklung für PPS-Systeme. Seit 1991 ist er Professor für Wirtschaftsinformatik an der Hochschule Pforzheim mit den Arbeitsschwerpunkten ERP-Systeme, Anwendungsintegration, Dokumenten- und Informationsmanagement. Seine jüngsten Veröffentlichungen behandeln Themen zur Dokumentenarchivierung, zum Modellieren von Geschäftsprozessen, zum Management von Informationssystemen, zur Enterprise Application Integration und zu Web-Services.

Inhaltsverzeichnis

Zusammenfassung	2
1 Einleitung	3
2 Merkmale XML-basierter Web-Services	4
2.1 Web-Service-Architektur.....	4
2.2 Ziele eines Serviceeinsatzes	6
2.3 Initiativen zur Einschränkung von Mängeln der Web-Service-Architektur	7
3 Einsatzmöglichkeiten für Web-Services.....	9
3.1 Klassifikation der Einsatzmöglichkeiten.....	9
3.2 Gründe für die Serviceverwendung aus Anwendersicht.....	10
3.2.1 Services mit besonderer Funktionalität	10
3.2.2 Services mit Kostenvorteilen	12
3.3 Gründe für Serviceangebote	12
3.3.1 Service-Providing als Geschäftsbeziehung.....	13
3.3.2 Unterstützung von Mitarbeitern und Partnern durch Web-Services	15
3.4 Anwendungsbereiche und Anbieter der Services.....	16
3.5 Kombination der Services mit vorhandenen Anwendungen	17
3.5.1 Services zum Ergänzen oder Ersetzen vorhandener Anwendungskomponenten.....	17
3.5.2 Services aus vorhandenen Anwendungen für andere Anwendungen.....	19
3.5.3 Services zum Datenaustausch mit Geschäftspartnern	20
3.5.4 Services für das Lifecycle Management von Anwendungssystemen	20
4 Lebenszyklus eines Web-Service	21
4.1 Beschreiben des Service	21
4.2 Veröffentlichen des Service.....	23
4.3 Finden des Service.....	25
4.4 Erstellen und Binden einer den Service anfordernden Anwendung	26
4.5 Aufrufen des Service	29
5 Fazit und Ausblick.....	31
Quellenverzeichnis.....	33

Zusammenfassung

Der Terminus *XML-basierter Web-Service* beschreibt eine auf Standards beruhende Softwarearchitektur, die das Zusammenspiel dreier Rollen (Service-Provider, Service-Verzeichnis und Service-Requestor) einschließt. Der Provider definiert eine Beschreibung für den Service und übergibt sie einem Requestor oder veröffentlicht sie in einem Verzeichnis. Der Requestor benutzt Suchfunktionen, um eine geeignete Service-Beschreibung zu finden. Diese verwendet er dafür, den Zugang zum Provider einzurichten und den Service anzufordern. Mit dieser Art der Vermittlung von Diensten für beliebige Benutzer, Mitarbeiter oder Unternehmen soll es möglich werden, Softwarefunktionen anzubieten, die dann über das Internet oder ein Intranet verfügbar sind.

Obwohl ihre technischen Grundlagen gerade erst entstehen, ist die Frage nützlich, wie Web-Services Geschäftsleute dabei unterstützen können, ihre Tätigkeit besser zu organisieren. Verschiedene Vorteile stellen mögliche Gründe für das Anbieten oder Benutzen von Web-Services dar. Service-Provider sind in der Regel Softwareunternehmen. Aber auch andere Unternehmen können Anbieter eines internen oder externen Service sein. Web-Services werden dabei Softwarepakete nicht ganz ersetzen. Aber sie können die Funktionalität vorhandener Softwareprodukte komplettieren und ergänzen oder Anwendungen integrieren. Hilfreich ist dies unter der Voraussetzung, dass bestimmte Geschäftsfälle dadurch schneller, besser, billiger oder nur auf diese Weise bearbeitet werden können.

Summary

The term XML Web Service describes a new standards-based software architecture that involves interactions between three roles: service provider, service registry and service requestor. The provider defines a service description and publishes it to a requestor or registry. The requestor uses a search operation to retrieve a suitable service description. He uses it to establish a connection with the provider and interact with the service. With this kind of service delivery to individuals, employees and businesses it should be possible to make software applications available over the Internet or an intranet.

Although their technical bases are just emerging, the question is how usefully Web Services could help business people better manage their business domains. Various advantages present possible reasons for the supplying or use of Web Services. Service providers are typically software businesses. However, any business can be a provider of an internal or external service as well. Web Services will not entirely replace packaged software. But they could complement and supplement existing software products or integrate applications. This would be helpful only if in certain instances businesses were able to accomplish a task faster, better and cheaper or if something could only be accomplished in this way that would otherwise not have been possible.

1 Einleitung

Durch die modernen Informationstechnologien verbessern sich die Möglichkeiten, nützliche Information zu finden, bereitzustellen, aufzubewahren, zu transformieren und zu präsentieren in rasantem Tempo. Anbieter von Hard- und Software bzw. von IT-Dienstleistungen, Informationsvermittler sowie andere in solche Entwicklungen Involvierte verkünden regelmäßig faszinierende Aussichten als Ergebnisse von Innovationen im IT-Bereich. Eine dieser Verheißungen betrifft eine neue Stufe der Bereitstellung von Softwarefunktionalität über das Internet oder ein Intranet – die sogenannten Web-Services. Damit ist ein Konzept zum Anbieten und Verwenden nützlicher Softwarefunktionen durch die Kommunikation zwischen Anwendungen auf der Basis bestimmter Schnittstellen bzw. Standards gemeint.

Eine herausragende Rolle spielt dabei die Extensible Markup Language (XML), die eine Auszeichnungssprache wie HTML darstellt. Im Unterschied zu HTML ist XML aber ein Sprachbaukasten (Metasprache) zur Erzeugung von bedarfsgerechten Auszeichnungssprachen für elektronische Dokumente. Sie wird vom World Wide Web Consortium (W3C) als "data format for structured document interchange on the Web" entwickelt und umfasst eine Gruppe verwandter Einzelstandards. In Abgrenzung gegenüber ganz unterschiedlichen anderen Dienstleistungen über das Internet bilden daher XML-basierte Web-Services den Gegenstand der nachfolgenden Ausführungen. Diese sollen verdeutlichen, wodurch Web-Services gekennzeichnet sind sowie wofür und wie sie genutzt werden könnten. Der Leser erhält einen Überblick, der die Einschätzung erleichtert, inwieweit sich die hoch gesteckten Erwartungen verwirklichen lassen. Den Schwerpunkt bildet die Frage, unter welchen Voraussetzungen Web-Services erfolgreich sein können – technische Details der Web-Service-Architektur sind über andere Quellen¹ zugänglich.

¹ vgl. z.B. <http://www.w3.org>, <http://www.uddi.org>, <http://www.webservicearchitect.com>, <http://www.webservices.org>. Veröffentlichungen zum Thema sowie ergänzende Angaben zu Einsatzmöglichkeiten finden sich in den Beiträgen von Schufft (2002) und Fleischer (2002)

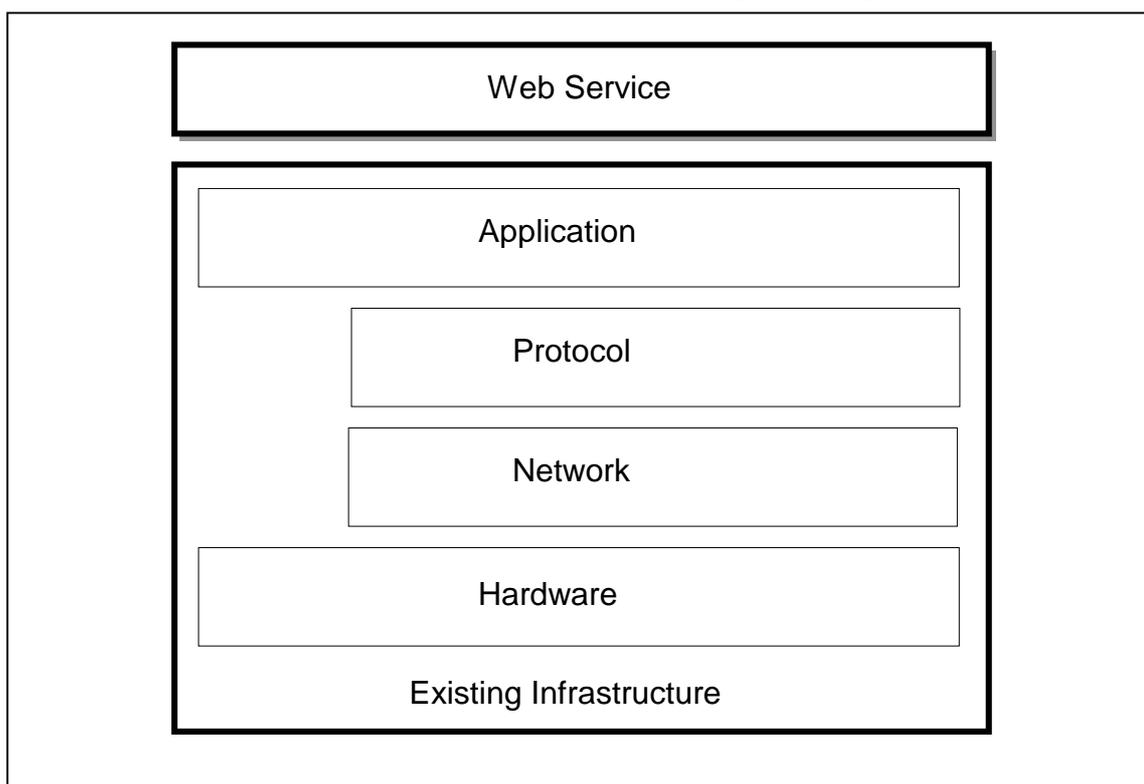
2 Merkmale XML-basierter Web-Services

2.1 Web-Service-Architektur

Im Unterschied zu vielen traditionellen Computeranwendungen bildet ein XML-basierter Web-Service kein monolithisches Softwareprodukt, das auf einem ganz bestimmten Computersystem lokalisiert eingesetzt wird. Andererseits stellt er auch nicht eine neue Vermittlungsform einer schon länger bekannten Dienstleistung für das Outsourcing von Prozessen der Informationsverarbeitung dar, die unter dem Namen Application Service Providing bekannt ist. Vielmehr verkörpert er eine Dienstleistung, die auf einer komponentenbasierten verteilten Anwendungsarchitektur mit standardisierten Schnittstellen beruht. Der Service resultiert aus einer unabhängigen und in sich abgeschlossenen bzw. gekapselten Anwendung, die definierte Aufgaben erfüllt und dabei Implementierungsdetails verbirgt. Er bildet eine standardisierte Schnittstelle für bestimmte Anwendungen und kann, wie aus Übersicht 1 ersichtlich, als zusätzliche Schicht in Anwendungssystemen angesehen werden.

Übersicht 1

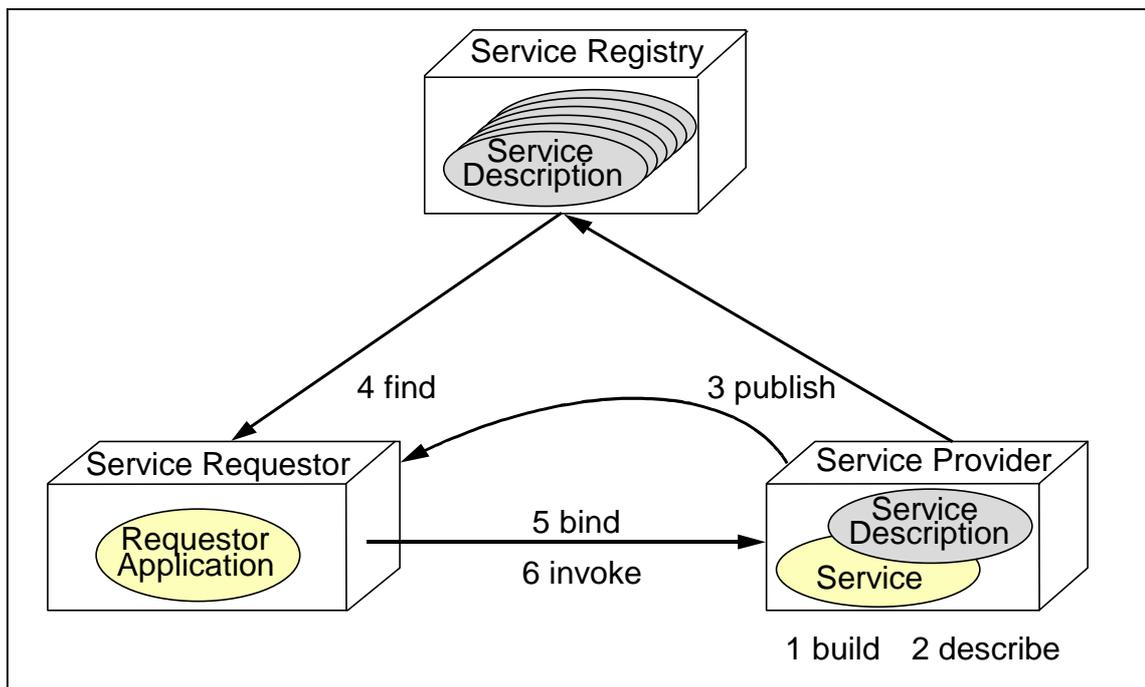
Web-Services als neue Schicht in Enterprise Applications



Der Web-Service wird von einem Anbieter (provider) bereitgestellt (build) und beschrieben (service description), eventuell in einem Verzeichnis (registry) veröffentlicht und registriert (publish) sowie von mindestens einem Anwender (requestor) angefordert (invoke), was die Übersicht 2 zeigt.

Übersicht 2

Generelle Architektur eines Web-Service



Aus technischer Sicht ist ein Service-Provider die Schnittstelle für eine Anwendung, die den Service bietet, und aus organisatorischer Sicht ist er der Anbieter einer Plattform für den Servicezugang. Die Servicebeschreibung durch eine spezielle Sprache ermöglicht einen syntaktisch abgestimmten Informationsaustausch mit den Service-Requestor-Anwendungen, die den Service anfordern. Sie wird dem Service-Requestor, der aus organisatorischer Sicht ein Interessent für die mit einem Service gebotene Funktionalität ist, direkt zur Verfügung gestellt oder mit einem besonderen Mechanismus zugänglich gemacht. Im zweiten Fall sucht der potenzielle Service-Requestor im Serviceverzeichnis nach einem geeigneten Service oder lässt sich über veränderte Serviceangebote informieren (find).

Soll der Service verwendet werden, muss eine Service-Requestor-Anwendung erstellt oder ergänzt werden, die in der Lage ist, den Service anzufordern. Dies ermöglichen die beschriebenen programmierbaren Schnittstellen sowie definierte Internet-Protokolle, insbesondere das Simple Object Access Protocol (SOAP) für XML-basiertes Messaging, zur Kommunikation zwischen den Anwendungen. Inwieweit Web-Services im Vergleich zu anderen Lösungsansätzen Vorteile bieten können, ergibt sich aus ihren Vorzügen und Schwächen.

2.2 Ziele eines Serviceeinsatzes

In zahlreichen Veröffentlichungen wird die Erwartung geäußert, dass Web-Services einen Weg für Unternehmen eröffnen, sich effizienter als bislang über das Internet zu präsentieren und ihr Intranet wirksamer zu nutzen. Dadurch ließe sich sowohl eine neue Stufe verbesserter Zusammenarbeit zwischen Partnern in Geschäftsprozessen und innerbetrieblicher Teamarbeit als auch eine erhebliche Aufwandsreduzierung für die Integration von Anwendungen erreichen. Begründet werden diese Hoffnungen mit der Orientierung der Web-Service-Architektur auf XML-bezogene und andere Web-Protokolle, die die in der Übersicht 3 aufgeführte Nutzeffekte ermöglichen könnten.

Übersicht 3

Erhoffte Effekte des Einsatzes von Web-Services (Auswahl)

Angestrebte Nutzeffekte	Vorteilhafte Voraussetzungen
Zusätzliche Bereitstellung nützlicher Information durch erweitertes Funktionsangebot	Interoperabilität von Anwendungen in heterogener Implementierungsumgebung
Leichter und/oder kostengünstiger Informationszugang	Flexibilität bei sich verändernden Realisierungsbedingungen
Erweiterte Auswahl für geeignete Software	Reduzierte Redundanz angebotener Softwarefunktionen
Schnellere und kostengünstigere Bereitstellung einer Integrationslösung	Erweiterbarkeit/Kombinationsfähigkeit des Konzepts
Verbesserte Betriebsunterstützung und Wartung der Integrationslösung	Leichte Anpassbarkeit der Schnittstellen

Die entscheidende Voraussetzung für die erhofften Nutzeffekte bildet das Schnittstellenkonzept, das auf XML-bezogenen Spezifikationen wie SOAP, WSDL und UDDI sowie auf anderen Web-Protokollen beruht, worauf im Abschnitt 4 eingegangen wird. Es ermöglicht die Interoperabilität der Web-Services in einer allgegenwärtigen (ubiquitous) Umgebung und gestattet die Integration solcher Anwendungen, die in unterschiedlichen Sprachen geschrieben und auf differenzierten Plattformen eingesetzt werden.

Durch seine zunehmende Verbreitung könnten zahlreiche Softwareentwickler ausreichend Erfahrungen sammeln, um eine flexible Integration von Anwendungen mit heterogenen sprachlichen Grundlagen erheblich zu erleichtern. Dadurch vergrößern sich die Auswahlmöglichkeiten für geeignete Software. Das bedeutet einen Schritt, um der Praktikabilität des Best-of-Breed-Konzepts beim Softwareeinsatz, also der Integration der besten Softwareprodukte zu einer Gesamtlösung, näher zu kommen. Darüber hinaus könnten die Qualität sowie der Support und die Wartung des Service durch die Übertragung der Verantwortung an kompetente und spezialisierte Provider günstig beeinflusst werden. Allerdings weist das Web-Service-Konzept der ersten Generation auch erhebliche Defizite auf, die eine allgemeine Nutzbarkeit dieses Konzepts bislang erheblich behindern.

2.3 Initiativen zur Einschränkung von Mängeln der Web-Service-Architektur

Vom World Wide Web Consortium oder von anderen Gremien geförderte Initiativen² richten sich darauf, die Praktikabilität des Web-Service-Konzepts für Business-to-Business-Kommunikation (B2B-Kommunikation) zu verbessern. Eine Auswahl solcher Bemühungen fasst die Übersicht 4 zusammen. Welche der aufgeführten Integrationsansätze in welchem Umfang in zukünftigen Lösungen, insbesondere auch für die Business-to-Business-Integration (B2B-Integration), Beachtung finden wird, soll an dieser Stelle nicht erörtert werden.

² Beschreibungen sind u.a. zugänglich über w3.org (2000)

Übersicht 4

**Mängel bisheriger Web-Services bei der B2B-Integration und
Lösungsansätze (Auswahl)**

Defizite bisheriger Web Services	Lösungsansätze zur Beseitigung
Unzureichende Sicherheit (Integrität, Vertraulichkeit, Authentizität)	Security Assertion Markup Language (SAML), eXtensible Access Control Markup Language (XACML), XML Key Management Specification (XKMS), XML Signature, WS-Security/WS-License (von Microsoft)
Keine Unterstützung asynchroner Kommunikation	WebSphere Message Queuing (von IBM), WS-Routing (von Microsoft)
Kein verteiltes Transaktionsmanagement	Transaction Authority Markup Language (XAML)
Keine standardisierte Lösung für die semantische Heterogenität	Electronic Business XML (ebXML) Repository, BizTalk-Repository (von Microsoft), RosettaNet Business Dictionary/RosettaNet Partner Interface Processes (PIP), Open Application Group Integration Specification (OAGIS), XMLEDI Repositories, XML Common Business Library (xCBL von CommerceOne), cXML (von Ariba), BMEcat
Unzureichendes Workflow Management	Wf-XML, Web Service Flow Language (WSFL, von IBM), XLANG (von Microsoft)
Keine Unterstützung des Lifecycle-Management	

Die Offenheit des Web-Service-Konzepts gestattet es jedoch, Ergebnisse aller Initiativen als Grundlage von Vereinbarungen zum Inhalt ausgetauschter Information zu verwenden. Dadurch entstehen Lösungen, die bestimmte Merkmale verschiedener Ansätze für die Anwendungsintegration kombinieren. Unter welchen Voraussetzungen bestimmte Realisierungsvarianten von Web-Services zweckmäßig sein könnten, ist anhand ihrer Einsatzmöglichkeiten zu untersuchen.

3 Einsatzmöglichkeiten für Web-Services

3.1 Klassifikation der Einsatzmöglichkeiten

Die Vielfalt der Einsatzmöglichkeiten soll an ausgewählten Aspekten verdeutlicht werden, die die Übersicht 5 zusammenfasst.

Übersicht 5

Aspekte zur Klassifikation von Web-Services (Auswahl)

Klassifikationsaspekte	Servicearten
Gründe für das Benutzen eines Service	<ul style="list-style-type: none"> - Nützliche Ergänzung der verfügbaren Softwarefunktionalität - Aufwandsärmere Bereitstellung von Softwarefunktionalität
Gründe für das Anbieten eines Service	<ul style="list-style-type: none"> - Service-Providing als Geschäftsbeziehung - Unterstützung von Mitarbeitern, Partnern oder beliebigen Benutzern
Art der Vergütung des Service	nach Zahlungsmodalitäten
Servicereichweite	<ul style="list-style-type: none"> - organisatorisch: unternehmensintern oder unternehmensextern - geographisch: lokal oder global
Benutzer des Service	<ul style="list-style-type: none"> - Für Mitarbeiter: Service-to-Employee - Für externe Partner: Service-to-Business - Für beliebige Internet-Benutzer: Service-to-Consumer
Einsatz in Lebensbereichen	nach Einsatzgebieten
Einsatz in Wirtschaftsbereichen bzw. in Branchen	nach Wirtschaftsbereichen bzw. Branchen
Bezug auf betriebliche Anwendungssysteme, -komponenten und -funktionen	nach Anwendungssystemen, -komponenten und -funktionen
Bezug auf Informationsobjekte und Daten	nach Informationsobjekten und Daten
Arten von Serviceanbietern	nach Arten von Serviceanbietern
Kombination mit Anwendungen	<ul style="list-style-type: none"> - Zum Ergänzen oder Ersetzen vorhandener Software-Komponenten - Aus vorhandenen oder neuen Software-Komponenten - Zum Datenaustausch zwischen Anwendungen - Für das Lifecycle Management von Anwendungen

Für welche Verwendung ein bestimmter Service empfehlenswert sein könnte, ist aus der Sicht sowohl der potenziellen Anbieter als auch der möglichen Anwender interessant.

3.2 Gründe für die Serviceverwendung aus Anwendersicht

Einen Web-Service bereitzustellen lohnt sich nur dann, wenn er auch für ausreichend viele Benutzer attraktiv ist. Dies kann der Fall sein, wenn der Service

- einen Zugriff auf Funktionalität erlaubt, die gebraucht wird, aber nicht auf andere Weise anforderungsgerecht bereitgestellt werden kann, oder wenn er
- geringere Zeitaufwände und Kosten bzw. eine bessere Qualität beim Bereitstellen gewünschter Softwarefunktionalität bewirkt.

Aber welche Arten von Anwendungen eignen sich überhaupt für solche Services? Sollten bestehende Lösungen, beispielsweise EDI-Lösungen oder gar ERP-Systeme zukünftig zu derartigen Serviceangeboten werden, was mitunter vorgeschlagen wird?

3.2.1 Services mit besonderer Funktionalität

Mit der ersten Generation von Web-Services werden inhaltlich weniger anspruchsvolle Informationsdienste, wie das Abrufen von Börsenkursen oder das Anzeigen der Temperatur an einem bestimmten Ort, angeboten. Diese sogenannten "Hase"-Services³ sind schnell entwickelt und dienen in erster Linie zum Ausprobieren der Web-Service-Technologie. Sie bieten oft einfache Funktionalität, die auch auf andere Weise anforderungsgerecht angeboten wird. Sie allein werden kaum bewirken, dass Web-Services verbreitet zum Einsatz gelangen. Dies setzt vielmehr voraus, dass sich das Angebot an Software mit nützlicher Funktionalität durch Web-Services erweitert und einen neuen oder umfassenderen Informationszugang ermöglicht.

³ vgl. Clark (2001/2)

Diese verbesserte Qualität der Informationsbereitstellung zeigt sich vor allem im Umfang, in der Integrität und in der Aktualität des Informationsangebots. Die Services könnten dazu beitragen, drei oft auftretende Mängel bei der Informationsbeschaffung zu beheben:

- Lückenhaftes bzw. mangelhaft aufbereitetes Informationsangebot
- Inkonsistente Daten und Dokumente
- Verspätete Informationsverfügbarkeit

Derartige Verbesserungen werden eher durch sogenannte "Schildkröten"-Services erreicht, die aufgrund einer ausgereiften Anforderungsanalyse Langlebigkeit bewirken. Sie berücksichtigen u.a. folgende Punkte:

- Es sollten vorwiegend neue Informationsangebote angestrebt werden, die das Internet oder ein Intranet als vernetzte Informationsquelle verwenden, statt vorhandene Lösungen durch Web-Services zu ersetzen.
- Web-Services sind Standard-Lösungen, die nicht ganz speziellen Anwendungserfordernissen gerecht werden können, sondern einer ausreichend großen Anzahl von wiederholt und bei mehreren Benutzern auftretenden Einsatzanforderungen entsprechen.
- Die Komplexität des Dienstes muss mit dem Funktionsbedarf der Benutzer übereinstimmen. Ungeeignete Granularität beeinträchtigt die Benutzbarkeit oder den Anpassungsaufwand.

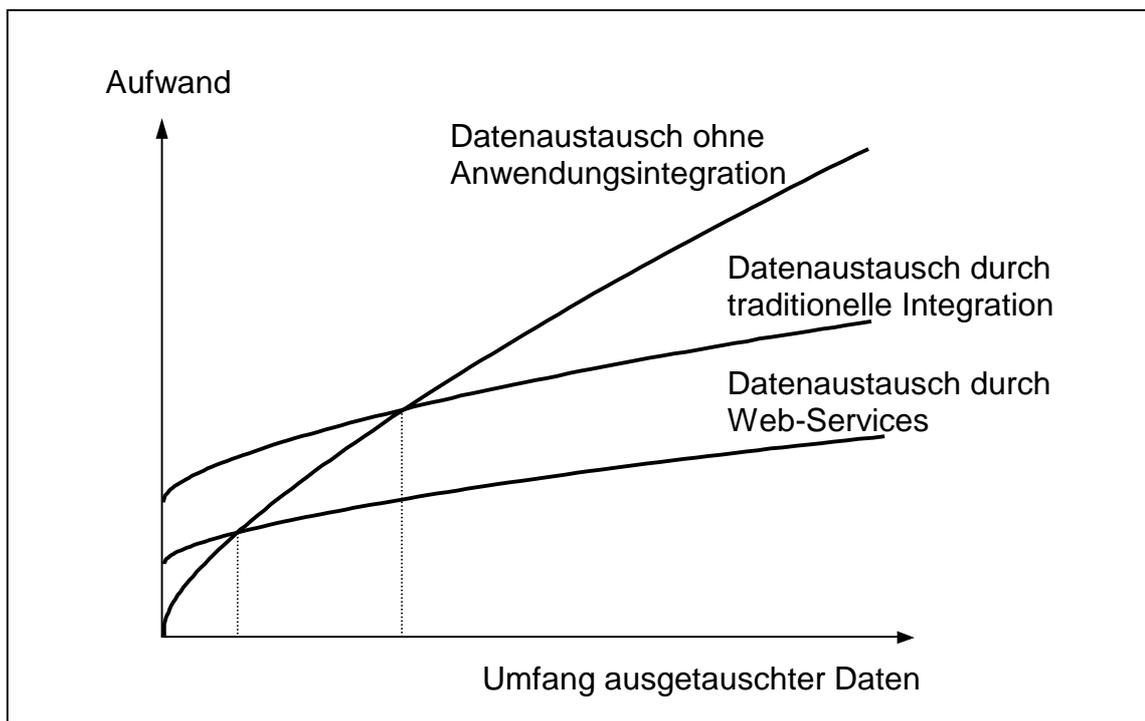
Mitunter lässt sich die gewünschte Information durch einen Web-Service auch einfacher, schneller bzw. mobiler als auf anderen Wegen zur Verfügung stellen, was den Aufwand für die Informationsverarbeitung reduziert.

3.2.2 Services mit Kostenvorteilen

Besonders hervorgehoben werden in der Literatur oft deutliche Einsparungen bei den einmaligen Entwicklungskosten gegenüber traditionellen Integrationsansätzen, die eine Anwendungsintegration und speziell die B2B-Integration bereits bei einem geringeren Aufkommen zwischen Geschäftspartnern auszutauschender Dokumente rentabel erscheinen lassen könnten (vgl. Übersicht 6). Wenn geeignete Softwarefunktionen gefunden worden sind, muss ein potenzieller Provider klären, wem er diese Funktionen zu welchen Konditionen als Web-Service anbieten will und wie dies technisch realisiert werden kann.

Übersicht 6

Vergleich von Varianten der Anwendungsintegration nach Aufwandsverläufen



3.3 Gründe für Serviceangebote

Aus einem von zwei Hauptgründen bietet ein Provider seinen Web-Service an: Er möchte entweder mit dem Angebot Geld verdienen, oder er unterstützt Part-

ner (in Geschäftsprozessen) oder Mitarbeiter, damit diese bestimmte Aufgaben besser erfüllen können.

3.3.1 Service-Providing als Geschäftsbeziehung

In diesem Fall liegt ein Geschäftsmodell vor, das weit gehende Ähnlichkeiten mit dem Application Service Providing aufweist. Bestimmte Software wird als Dienstleistung online aufgrund geschäftlicher Vereinbarungen zwischen Service-Provider und Service-Requestor angeboten. Konsumenten des Service sind andere Unternehmen bzw. Organisationen, die eine Steigerung der Effizienz ihrer Informationsverarbeitung durch das Angebot erwarten. Es entsteht eine Geschäftsbeziehung, weil die angebotene Funktionalität zu den festgelegten Konditionen (Zahlungsmodalitäten, Service Level Agreement, geographische Reichweite, Schnittstellenfestlegungen usw.) für den Service-Requestor attraktiv ist.

Übersicht 7

Varianten zur Berechnung von Gebühren für Web-Services⁴

Variante	Erläuterung
pro Benutzung (Prepaid)	Der Kunde kauft eine bestimmte Anzahl an Benutzungen. Jede Benutzung wird registriert. Wenn das Limit erreicht ist, wird der Kunde automatisch gefragt, ob er weitere Benutzungen kaufen möchte.
pro Periode (Subscription)	Der Kunde bezahlt für eine unbegrenzte Benutzung innerhalb einer definierten Periode (z.B. Monat, Quartal, Halbjahr oder Jahr). Abhängig von der vorher getroffenen Vereinbarung wird der Kunde ggf. nach Ablauf der Periode automatisch gefragt, ob er die Benutzung fortsetzen möchte.
einmalige Zahlung (Prepaid)	Der Kunde zahlt einmalig für eine unbegrenzte Benutzung für die gesamte Lebensdauer des Service, wenn diese einige Tage, Wochen oder Monate beträgt (z.B. im Zusammenhang mit Olympischen Spielen oder anderen Ereignissen).
keine Zahlung (Freeware)	Der Service ist kostenlos; entweder für die gesamte Lebensdauer oder für eine kurze Versuchsperiode.

⁴ nach Clark (2001/3)

Zu den Zahlungsmodalitäten gehören der Modus der Gebührenberechnung (Zeitpunkt der Abrechnung: vor, mit oder nach der Benutzung und die Berechnungsweise) sowie die Zahlungsweise (Elektronische Geldbörsen, E-Cash, SmartCards, Geldkarte, Elektronischer Scheck, Inkasso, Rechnung, Lastschrift u.a.). Bei der Gebührenberechnung differenziert Clark aus der Fülle von Möglichkeiten die in der Übersicht 7 aufgeführten Varianten.

Das Service Level Agreement regelt vor allem Vorgaben zur Verfügbarkeit (Zuverlässigkeit), zur Performance (Antwortzeiten) sowie zur Sicherheit (Vertraulichkeit, Requestor- und Provider-Authentifikation, Integrität, Bestätigung des Absendens und Empfangens übermittelter Daten sowie Rekonstruierbarkeit des Inhalts übermittelter Information einschließt). Ein hohes Service-Niveau, eine professionelle Unterstützung, schnelle Verfügbarkeit aktualisierter und verbesserter Software und andere typische Vorzüge eines "guten" Application-Service-Providing sollten auch Merkmale des Web-Service-Providing darstellen. Weitere Festlegungen betreffen die Schnittstellen (Datentypen, Nachrichtenschema, Austauschmethoden usw.), die durch die Servicebeschreibung sichtbar werden und auf die über Serviceverzeichnisse zugegriffen werden kann.

Wenn der Service international angeboten werden soll, müssen nationale bzw. regionale Besonderheiten in der Sprache (Dokumentation, Schriftzeichen usw.), in der Zahlungsweise oder in den rechtlichen Rahmenbedingungen ebenso Beachtung finden wie die globale Erreichbarkeit.

Neben der Erstellung und Veröffentlichung des Service sind auch Marketing- und Verkaufsaktivitäten vorzusehen, wenn er zu Einnahmen führen soll. Zur Einsatzvorbereitung gehören ebenso die Akkreditierung des Providers, die externe Kontrolle des Service durch einen Auditor und das Einrichten von Abrechnungskonten⁵. Die Akkreditierung soll die Vertrauenswürdigkeit des Angebots bestätigen und beruht auf bestimmten Kriterien für die Beurteilung des Providers sowie der Qualität seines Service. Der Auditor soll eine neutrale In-

⁵ vgl. Clark (2001/1)

stanz sein, die regelmäßig die Einhaltung des Service Level Agreement kontrolliert. Abrechnungskonten zeigen den Umfang der Benutzung von Services und das Ende der vereinbarten Nutzungsdauer. Sie können auf einer technischen Plattform für mehrere Provider eingerichtet werden, so dass für einen Anwender als Kunden ein zentraler Abrechnungspunkt für die von ihm genutzten Services entsteht.

3.3.2 Unterstützung von Mitarbeitern und Partnern durch Web-Services

Dieser Fall fasst die nicht primär auf Gewinnerzielung ausgerichteten Service-Angebote zusammen, die jedoch für Benutzer nützlich sind, weil sie bestimmte Funktionen schneller, einfacher, besser oder überhaupt nur auf diese Weise verfügbar machen. Dieses Angebot kann für Mitarbeiter eines Unternehmens, für Geschäftspartner oder für beliebige Benutzer des Internet gedacht sein. Meistens soll dadurch eine Zusammenarbeit von Partnern, beispielsweise durch leistungsfähigeren bzw. rationelleren Datenaustausch unterstützt werden. Nach den Adressaten des Service lassen sich also unterscheiden

- Service-to-Consumer (S2C)
- Service-to-Employee (S2E)
- Service-to-Business (S2B)

Welche Arten von Benutzern adressiert sind, bestimmt die Komplexität des Service und die an ihn zu stellenden Anforderungen im Service-Level erheblich. Einfache Abfragen wie das Umrechnen von Beträgen verschiedener Währungen, das Abrufen von Börsenkursen oder das Anzeigen der Temperatur an einem bestimmten Ort richten sich vorwiegend an beliebige Internetbenutzer. Externe Geschäftspartner zu unterstützen, stellt in Bezug auf die Erfordernisse der Sicherheit und Performance oft eine erheblich größere Herausforderung dar als Services für interne Benutzer.

3.4 Anwendungsbereiche und Anbieter der Services

Welche Arten von Funktionen und Anwendungen sich überhaupt als Web-Services eignen, wird zurzeit noch kontrovers diskutiert. Einigkeit besteht darin, dass ein Service nicht zu spezielle Anwendungsbedingungen voraussetzen darf, damit er für ausreichend viele Benutzer attraktiv ist. Hinsichtlich der angebotenen Funktionalität lassen sich die Serviceangebote nach verschiedenen Aspekten, beispielsweise nach dem Einsatzgebiet, dem Wirtschaftsbereich oder den Informationsobjekten, klassifizieren.

Eine Untersuchung veröffentlichter Beschreibungen von Web-Services lässt erkennen, dass die meisten Lösungsbeispiele bislang nur Testzwecken dienen⁶. Allerdings ist seit dem 4. Quartal des Jahres 2001 ein deutlicher Anstieg publizierter Services mit wirtschaftlichem Hintergrund festzustellen. Derartige Dienste beziehen sich auf die Wirtschaft und andere Lebensbereiche, beispielsweise auf die öffentliche Verwaltung oder das Gesundheits- und das Bildungswesen. Bei den Wirtschaftsbereichen sind vor allem für das Kredit- und Versicherungsgewerbe, für das Verkehrswesen und die Nachrichtenübermittlung sowie für das Verarbeitende Gewerbe und den Handel Anwendungsbeispiele dokumentiert.

Das erste Unternehmen, das die Idee des Web-Service der Öffentlichkeit präsentierte, war die Firma HP mit ihrem E-Speak-Projekt aus dem Jahr 1999. Aber erst nachdem Microsoft im Jahr 2000 die „.NET-Strategie“ vorgestellt hatte, gewann das Thema eine breitere Beachtung. Inzwischen verfolgen immer mehr Softwareanbieter, voran die Firma IBM, eigene Web-Service-Strategien mit mehr oder weniger ausgereiften Lösungskonzepten. Diese Strategien können ein Angebot von Tools zum Entwickeln, Betreiben und Benutzen der Services aber auch von unmittelbar benutzbaren Services einschließen.

Als wichtigste Anbieter solcher Services zeichnen sich aufgrund vorliegender Lösungen Softwarehäuser und andere Firmen ab, die bestimmte Informations-

⁶ vgl. Fleischer (2002)

dienstleistungen für Kunden oder ihre Mitarbeiter bieten. Den Aufbau eigener kostenpflichtiger Services bietet beispielsweise Microsoft unter dem Namen *My Services* (vormals *HailStorm*) an. Von beliebigen Endgeräten, den Hailstorm-Endpoints, kann über das Internet auf Services wie Terminkalender, Adressbuch und Dokumentenspeicher zugegriffen werden.

Das Angebot eines Web-Service kann dadurch entstehen, dass eine servicefähige Anwendung speziell entwickelt oder eine vorhandene Lösung als Service nutzbar gemacht wird. In beiden Fällen ist zu klären, wie die Services mit den bestehenden Anwendungen zu kombinieren sind.

3.5 Kombination der Services mit vorhandenen Anwendungen

Nach der Art der Kombination mit bestehenden Anwendungen differenziert dienen Web-Services vor allem dazu, vorhandene Anwendungskomponenten zu ergänzen oder zu ersetzen, verschiedene Anwendungen zu kombinieren, Daten mit Geschäftspartnern auszutauschen und das Lifecycle-Management von Anwendungen zu unterstützen.

3.5.1 Services zum Ergänzen oder Ersetzen vorhandener Anwendungskomponenten

Jedes Anwendungssystem weist in der Breite und Tiefe seiner Funktionalität bestimmte Mängel auf, deren Behebung durch Web-Services erwogen werden könnte, wobei u.a. die in Übersicht 8 zusammengefassten Einsatzvarianten relevant erscheinen. Eine günstigere Kostenstruktur lässt sich eventuell dadurch erreichen, dass selten benötigte oder sehr teure Komponenten einer Anwendung bei Bedarf über einen Web-Service genutzt werden. Aber auch fehlende oder spezielle Anwendungsfunktionen könnten durch solche Services verfügbar werden, die Besonderheiten bestimmter Branchen bzw. Prozess-typen berücksichtigen oder zusätzliche Möglichkeiten verbesserter Informationsverarbeitung eröffnen. Derartige Beispiele beschreibt McKie⁷.

⁷ vgl. McKie (2001), S. 22 ff.

Übersicht 8

**Varianten des Einsatzes von Web-Services zum Anreichern
vorhandener Anwendungen (Auswahl)**

Einsatzziele	Einsatzvarianten	Anwendungsbeispiele
Vermeidung hoher Investitionskosten	<ul style="list-style-type: none"> - Selten benötigte Funktionalität - Aufwendige Software-Komponenten 	<ul style="list-style-type: none"> - Länderversionen - ePayment-Abwicklung
Erweiterung nützlicher Funktionalität	<ul style="list-style-type: none"> - Externe Information mit hoher Dynamik - Alternative Funktionen für besondere Anforderungen - Zusatzfunktionen für umfassendere Automatisierung der Informationsverarbeitung 	<ul style="list-style-type: none"> - Auswertung von Wetterberichten für die Routenplanung im Versand - Externe Analyse des Zahlungsverhaltens von Kunden - Berechnung von Durchlaufzeiten für spezielle Abläufe - Abgleich von Kontaktterminen mit Terminplaner-Lösung
Kombination der besten oder geeignetsten Software mehrerer Anbieter	Best-of-Breed-Lösung	Kombination von ERP-, CRM-, PLM- und BI-System

Eine Voraussetzung für die nahtlose Integration solcher Services bildet eine ausreichende Harmonisierung der Komponentenarchitekturen und Entwicklungskonzepte zwischen Service-Provider und -Requestor. Anderenfalls stimmen angebotene und benötigte Funktionalität nicht ausreichend überein. Dies gilt umso mehr für das Best-of-Breed-Konzept beim Softwareeinsatz, dessen Umsetzung nach verbreiteter Auffassung durch Web-Services praktikabel werden könnte. Insbesondere einige Anbieter von ERP-Software verweisen jedoch auf das damit verbundene enorme Integrationsproblem. Sie bezweifeln deshalb, dass Softwarekombinationen als Multi-Vendor-Lösungen von mehr als einer Hand voll Softwareanbietern zukünftig sinnvoll werden. Bereits bestehende Schwierigkeiten beim Lifecycle-Management von Anwendungssystemen

erhöhen sich durch die Einbeziehung zahlreicher Web-Services weiter erheblich. Irani⁸ erläutert dies aus der Perspektive des Versionsmanagements.

3.5.2 Services aus vorhandenen Anwendungen für andere Anwendungen

Für attraktive Web-Services wird es in den meisten Fällen vorteilhaft sein, servicefähige Anwendungen speziell zu entwickeln. Unter bestimmten Voraussetzungen ist es auch zweckmäßig, eine vorhandene Lösung als Service nutzbar zu machen. Dafür bieten sich zwei Möglichkeiten: Entweder wird die Lösung, z.B. ein ERP-System, so gekapselt, dass sie über beschriebene, programmierbare Schnittstellen sowie über ein definiertes Internet-Protokoll, insbesondere das Simple Object Access Protocol (SOAP) für XML-basiertes Messaging, zur Kommunikation zwischen Anwendungen erreichbar ist. Einen Ansatz dafür bieten vorhandene API's für ERP-Systeme und andere komplexe Anwendungssysteme. So können u.a. alle Business Application Programming Interfaces (BAPIs) der SAP-Software aufgrund ihrer Schnittstellenbeschreibungen⁹ als Web-Services verfügbar gemacht werden.

Beispiele für Services aus dem ERP-System sind Verfügbarkeitsermittlungen und Reservierungen sowie Preisermittlungen für (mobile) SFA-Lösungen oder die Bereitstellung von Organisations- und von Mitarbeiterdaten für eine Projektmanagementlösung. Eine bestehende Anwendung kann aber auch in feiner granulierten Komponenten umgewandelt werden, die allein existenzfähig sind und sich mit anderen Komponenten kombinieren lassen, worin ein wesentlicher Vorzug der Web-Services gegenüber traditionellen monolithischen Systemen gesehen wird. Das bedeutet, ein Anwendungssystem als ein Bündel zusammengehörender Services anzusehen und zu verwalten.

Durch Web-Services können Anwendungen leicht und schnell in ein Enterprise Information Portal integriert werden oder zu einem Provider (Third-party Business, Channel Partner) ausgelagert werden. Ein Beispiel dafür bildet ein Enter-

⁸ vgl. Irani (2001)

⁹ vgl. <http://ifr.sap.com>

prise Information Portal, das intern für Mitarbeiter oder für bestimmte externe Partner dazu genutzt wird, bestimmte Angaben über Kunden des Unternehmens aus zwei servicefähigen Anwendungen zusammenzustellen. Je nach den gewünschten Angaben werden die Daten aus einem ERP-System oder einem CRM-System oder aus beiden Systemen abgerufen. Diese Systeme fungieren als Service-Provider und das Portal als Service-Requestor.

3.5.3 Services zum Datenaustausch mit Geschäftspartnern

Durch Web-Services kann dazu beigetragen werden, Daten zwischen Anwendungen innerhalb eines Unternehmens oder verschiedener Partner auszutauschen. Vor allem bei der B2B-Integration entstehen Möglichkeiten, Anwendungen aufgrund der konzeptionellen und der technischen Basis solcher Dienste miteinander zu verknüpfen.

Diesem Anliegen dienen bestimmte Initiativen¹⁰ zur Vereinfachung, Verbesserung und Verbreitung der B2B-Integration. Sie haben zum Ziel, die relativ hohen Entwicklungsaufwände für Integrationslösungen spürbar zu verringern. Ihre Ergebnisse stellen Spezifikationen von Schnittstellen zur Anwendungsintegration, Hilfen für die Entstehung solcher Spezifikationen oder Frameworks dar, die mehrere Aspekte beim Entstehen von B2B-Lösungen unterstützen.

3.5.4 Services für das Lifecycle Management von Anwendungssystemen

Die Nutzenpotenziale durch Anwendungssysteme lassen sich u.U. durch Web-Services besser erschließen, die das Management des Systemlebenszyklusses erleichtern. Einige Anwendungsbeispiele hierfür zeigt Übersicht 9. Diese Art der Servicenutzung zielt in erster Linie auf eine spürbare Aufwandsreduzierung und bietet sich für verschiedene Softwarearten an.

Übersicht 9

**Varianten des Einsatzes von Web-Services zum Lifecycle-Management
für Anwendungssysteme (Auswahl)**

Einsatzziele	Einsatzvarianten	Anwendungsbeispiele
Beschleunigte und aufwandsärmere Software-einführung	Remote Customizing	Systemkonfiguration
Verbesserte Benutzer- und Betreiberunterstützung	Remote Help Desk	Anfrage an Experten
Schnellere Verfügbarkeit erweiterter oder verbesserter Funktionalität	Upgrade-Alternative	Funktions-Demo-Version

Für die Beurteilung der Zweckmäßigkeit eines Web-Services interessiert nicht nur, wofür er geeignet ist, sondern auch, wie er entsteht. Diese Frage wird im folgenden Abschnitt behandelt.

4 Lebenszyklus eines Web-Service

4.1 Beschreiben des Service

Ein wesentliches Merkmal der Web-Services ist, dass sie eine Definition ihrer Schnittstellen aufweisen und sich somit selbst beschreiben. Vor allem mit Hilfe der Web Service Description Language (WSDL) und gegebenenfalls mit der Web Service Conversational Language (WSCL) entsteht ein XML-Dokument, das es Anwendern erlaubt, danach eine geeignete, den Service nutzende Client-Anwendung zu entwickeln.

Am einfachen Beispiel eines Web-Service, der Umrechnungen zwischen EURO und DM ermöglicht, zeigt Schufft¹¹ die erforderliche XML-Definition für die Service-Schnittstelle (vgl. Übersicht 10).

¹⁰ Beschreibungen sind u.a. zugänglich über <http://www.basda.org>, <http://cXML.org>, <http://bmecat.org>, <http://biztalk.org>, <http://ebxml.org>, <http://rosettanet.org>, <http://openapplications.org>, <http://commerceOne.com>, <http://xmledi-group.org>

¹¹ vgl. Schufft (2002), S. 13 f.

Übersicht 10

Beispiel für eine Schnittstellendefinition nach Schufft

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EuroCalcRemoteInterface"
  targetNamespace="http://asch0002-5.sub001.asprivat.de:82/definitions/EuroCalcRemoteInterface"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://asch0002-5.sub001.asprivat.de:82/definitions/EuroCalcRemoteInterface"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="getEuroFromDmRequest">
    <part name="dmWert" type="xsd:double"/>
  </message>
  <message name="getEuroFromDmResponse">
    <part name="result" type="xsd:double"/>
  </message>
  <message name="getDmFromEuroRequest">
    <part name="euroWert" type="xsd:double"/>
  </message>
  <message name="getDmFromEuroResponse">
    <part name="result" type="xsd:double"/>
  </message>
  <portType name="EuroCalc">
    <operation name="getEuroFromDm" parameterOrder="dmWert">
      <input message="tns:getEuroFromDmRequest" name="getEuroFromDmRequest"/>
      <output message="tns:getEuroFromDmResponse" name="getEuroFromDmResponse"/>
    </operation>
    <operation name="getDmFromEuro" parameterOrder="euroWert">
      <input message="tns:getDmFromEuroRequest" name="getDmFromEuroRequest"/>
      <output message="tns:getDmFromEuroResponse" name="getDmFromEuroResponse"/>
    </operation>
  </portType>
  <binding name="EuroCalcBinding" type="tns:EuroCalc">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getEuroFromDm">
      <soap:operation soapAction="" style="rpc"/>
      <input name="getEuroFromDmRequest">
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://asch0002-5.sub001.asprivat.de:82/net.schufft.eai.samples.EuroCalc"
          use="encoded"/>
      </input>
      <output name="getEuroFromDmResponse">
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://asch0002-5.sub001.asprivat.de:82/net.schufft.eai.samples.EuroCalc"
          use="encoded"/>
      </output>
    </operation>
    <operation name="getDmFromEuro">
      <soap:operation soapAction="" style="rpc"/>
      <input name="getDmFromEuroRequest">
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://asch0002-5.sub001.asprivat.de:83/net.schufft.eai.samples.EuroCalc"
          use="encoded"/>
      </input>
      <output name="getDmFromEuroResponse">
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://asch0002-5.sub001.asprivat.de:83/net.schufft.eai.samples.EuroCalc"
          use="encoded"/>
      </output>
    </operation>
  </binding>
</definitions>

```

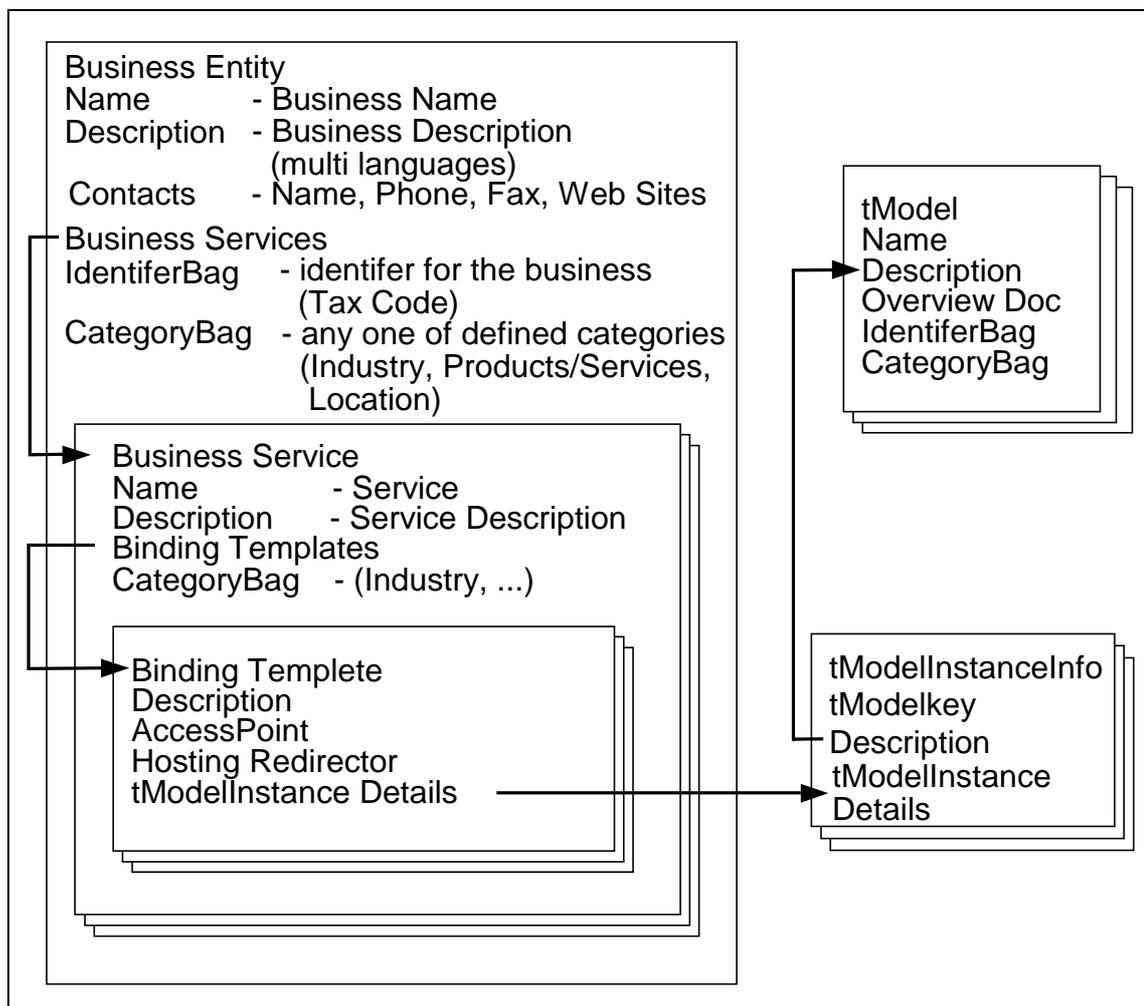
Der Service EuroCalc bietet die Funktion (Methode) getEuroFromDm zum Umrechnen von DM-Beträgen in Euro-Beträge sowie die Funktion getDmFromEuro zum Umrechnen von Euro-Beträgen in DM-Beträge. Diese Definition kann den Anwendern direkt bereitgestellt werden, wenn diese, wie es beim Business-to-Business Commerce oft der Fall ist, bekannt sind. Um es weiteren potenziellen Anwendern über Suchservices offerieren zu können, ist Universal Description, Discovery and Integration (UDDI) geschaffen worden.

4.2 Veröffentlichen des Service

Die Veröffentlichung der Service-Beschreibung in UDDI-Verzeichnissen erleichtert die manuelle oder programmgesteuerte Suche nach geeigneten Services.

Übersicht 11

Inhalt eines UDDI-Eintrags



Neben Spezifikationen und einem API-Schema für die Entwicklung von Suchservices enthält UDDI ein Firmenverzeichnis (weiße Seiten), mit zugeordneten Serviceverzeichnissen (gelbe Seiten) sowie technischen Beschreibungen der Services (grüne Seiten), was Übersicht 11 zeigt. Ein UDDI-Registry kann

- als unternehmensinternes Verzeichnis,
- innerhalb eines Information-Portals,
- für ausgewählte Partner,
- für bestimmte Wirtschaftsbereiche oder
- als ein für jeden Internet-Benutzer offenes Verzeichnis

betrieben werden. Ein Zugriff ist entweder manuell über die Web-Sites des jeweiligen Betreibers oder über den programmgesteuerten Austausch solcher XML-Dokumente möglich, die dem durch UDDI spezifizierten XML-Schema entsprechen. Im Beispiel des Web-Service zum Bereitstellen der Kundendaten fordert das Portal von der UDDI-Registry die Auskunft an, welche Anwendungen Information über Kunden bereitstellen können. Die UDDI-Registry holt diese Auskunft aus einer Datenbank mit WSDL-Beschreibungen und liefert das Ergebnis an das Portal zurück.

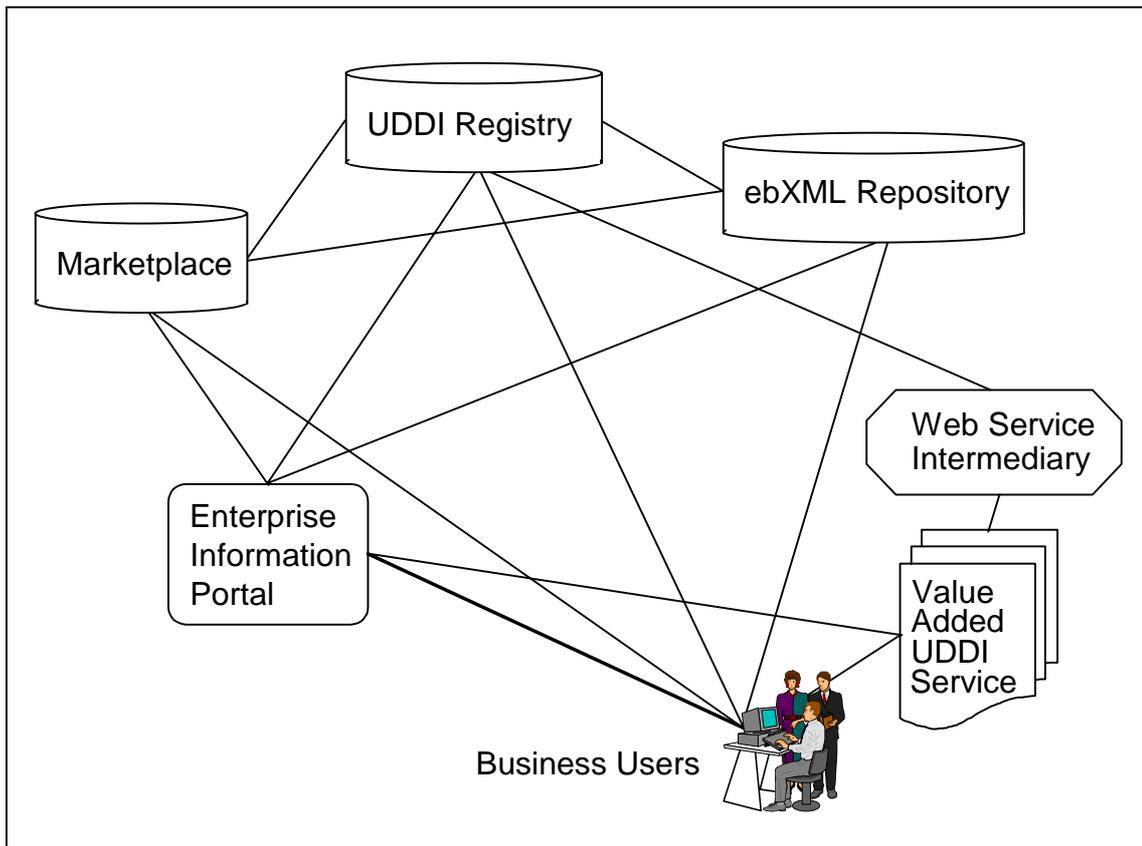
Allein auf diese Weise ist es jedoch nicht aktuell und umfassend genug möglich, Geschäftspartner herauszufinden, die in einem abgegrenzten geographischen Bereich ein spezifisches Produkt oder eine Dienstleistung zu einem bestimmten Preis sowie zu bestimmten zeitlichen und anderen Konditionen anbieten. Wie mit der Übersicht 12 dargestellt, sollten diese Verzeichnisse mit elektronischen Produktkatalogen (beispielsweise über Business-to-Business-Portale oder elektronische Marktplätze) zusammenwirken können, was die UDDI-Version 3 erlauben wird.

Eine derartige Kopplung gestattet es, Anbieter danach auszuwählen, ob sich ihre Services, beispielsweise ein Produktkatalog in seinem CRM-System, leicht

mit den Anwendungssystemen (z.B. dem Beschaffungssystem) des Suchenden verknüpfen lassen¹².

Übersicht 12

Kombinationsmöglichkeiten für die Suche nach Geschäftspartnern und Web-Services



4.3 Finden des Service

Die Möglichkeiten zum Finden und Einschätzen geeigneter Services erweitern sich durch sogenannte UDDI Value Added Service Supplier¹³. Das sind Vermittler, die auf dem UDDI-Repository aufsetzen und beispielsweise folgende Leistungen anbieten:

¹² vgl. von Riegen (2001), S. 2

¹³ vgl. Clark (2001/4)

- Spezielle Such-Filter
- Automatische Benachrichtigungen bei allen für den Konsumenten (Requestor) relevanten Veränderungen von UDDI-Einträgen
- Beurteilungen von Services oder deren Provider
- Überwachung der Einhaltung von Service Level Agreement Contracts innerhalb des UDDI-Verzeichnisses
- Regelmäßige Bereitstellung des UDDI-Newsletter

Die UDDI-Newsletters informieren z.B. über neue Web-Services, die im UDDI-Verzeichnis registriert wurden, geänderte Ratings für bestimmte Services und die Datenkonsistenz des UDDI-Verzeichnisses im sogenannten „UDDI Wether report“. Einen Nutzen aus derartigen Vermittlerdiensten haben sowohl die Konsumenten als auch die Provider von Web-Services, die daher jeweils einen anteiligen Beitrag zu ihrer Finanzierung übernehmen könnten.

Um den angebotenen Service auch benutzen zu können, müssen mehr oder weniger komplexe Anwendungen erstellt werden, die mit geeigneten Schnittstellen ausgestattet sind.

4.4 Erstellen und Binden einer den Service anfordernden Anwendung

Für die anfordernde Anwendung sind zwei Entstehungsvarianten zu unterscheiden:

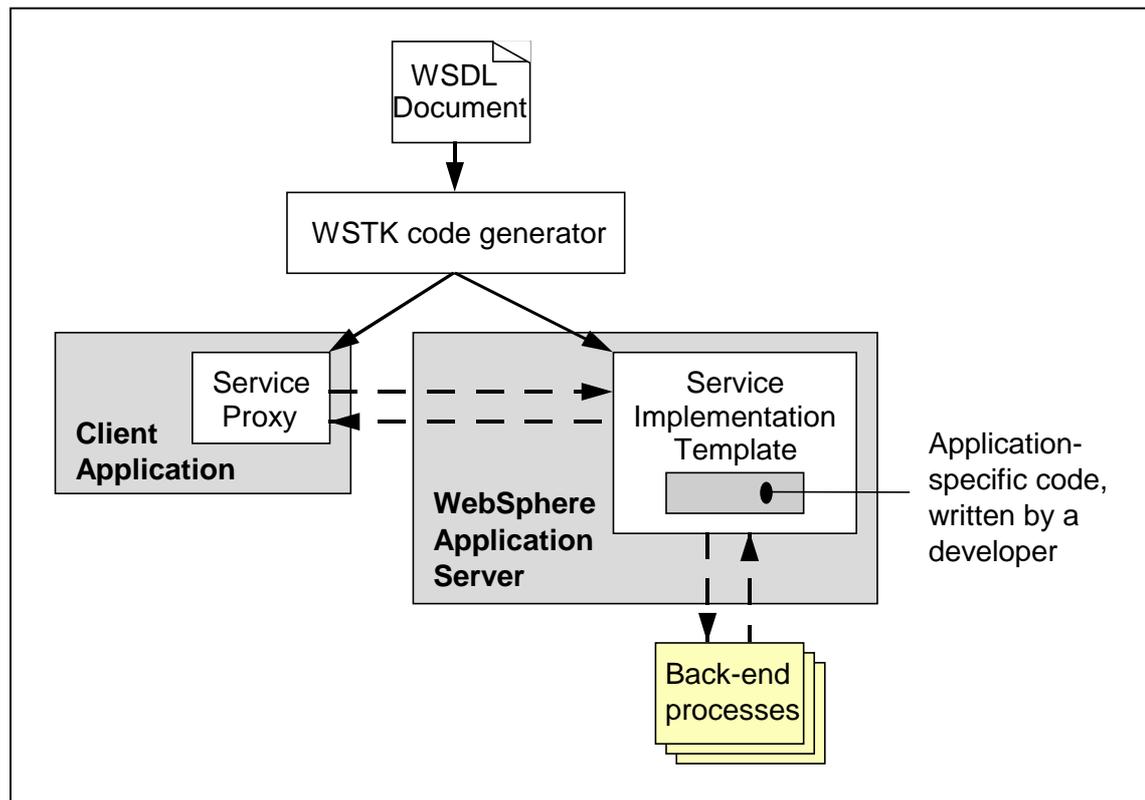
- Entwickeln einer speziellen Client-Anwendung
- Ergänzen einer vorhandenen Anwendung

Bei der ersten Variante werden die anfordernden Anwendungen speziell für den angebotenen Service erstellt. Geeignete Entwicklungsumgebungen zum Bereitstellen von Web-Services und Clients erleichtern dies. So generiert zum Beispiel, wie mit der Übersicht 13 verdeutlicht, IBM's „Web Service Toolkit“ aufgrund des WSDL-Dokuments eine Client-seitige Service-Proxy-Klasse, über die

die erforderliche Programm-Schnittstelle (API) für den Service-Requestor lokal verfügbar wird¹⁴.

Übersicht 13

Rapid Integration mit IBM's Web Service Toolkit



Quelle: IBM

Der Service-Proxy enthält den Code, der für den Aufruf des betreffenden Web-Service erforderlich ist. Durch eine zusätzliche Möglichkeit zur Generierung gekapselter Benutzerschnittstellen lässt sich der Entwicklungsprozess weiter vereinfachen und eine Anwendungsintegration auch auf der Ebene der Benutzerschnittstellen erreichen. Aufgrund einer "Prepackaged Desktop Application" ist kein großer Aufwand zu bewältigen, um den Service für Benutzer zugänglich zu machen.

Die zweite Variante berücksichtigt eine vorhandene Anwendung, zum Beispiel ein Information-Portal für Mitarbeiter oder für Geschäftspartner, die zur Nutzung

¹⁴ Ein Beispiel zeigt Schufft (2002), S. 37

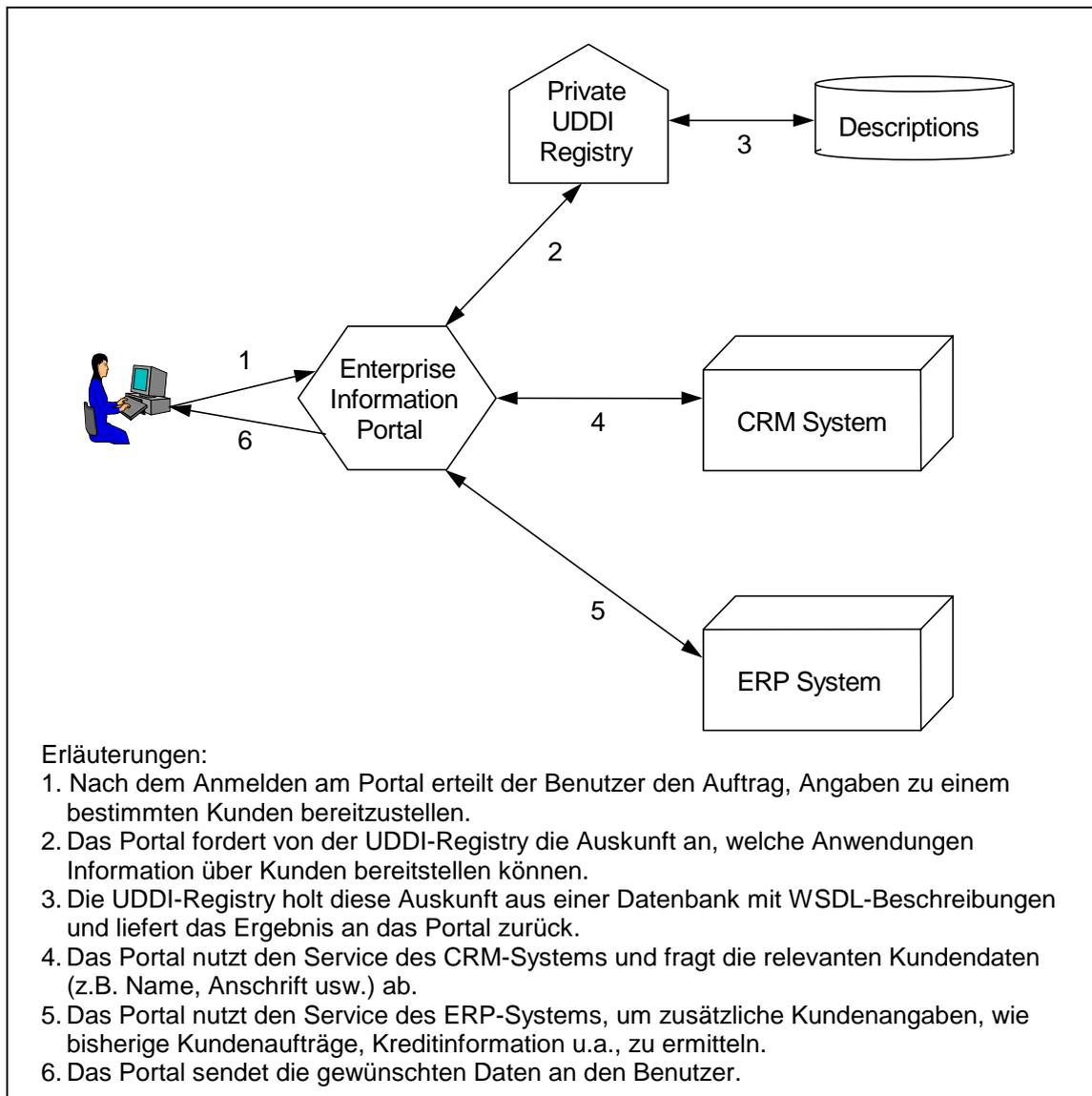
des Service die erforderlichen Erweiterungen bzw. Schnittstellen erhält. Hierbei treten in der Regel erheblich größere Schwierigkeiten auf, die vor allem aus dem Problem der semantischen Heterogenität der kommunizierenden Anwendungen resultieren. Aber auch divergierende Komponentenarchitekturen und Entwicklungskonzepte, unzureichend abgestimmte Funktionalität sowie unzulängliche Fähigkeiten der verfügbaren Softwareentwickler können oft unterschätzte Schwierigkeiten bewirken und mitunter einen erfolgreichen Serviceeinsatz verhindern.

Die anfordernde Anwendung kann statisch oder dynamisch an die Service-Anwendung gebunden werden, wobei folgende Möglichkeiten zu unterscheiden sind:

- Build Time Static Binding: Service-Interface und Implementierung (Provider) sind zur Entwicklungszeit bekannt. Dies ist die häufigste Variante.
- Build Time Dynamic Binding: Service-Interface ist zur Entwicklungszeit bekannt, die Implementierung (Provider) wird zur Laufzeit ermittelt. Diese Variante erfordert einen "Overhead" für die Suche des Providers. Ihre Vorteile sind, dass ein Wechsel des Providers keine Änderungen bei der anfordernden Anwendung nach sich zieht und zur Laufzeit eventuell zwischen mehreren Anbietern gewählt werden kann (z.B. das preiswerteste Angebot).
- Run Time Dynamic Binding: Service-Interface und Implementierung (Provider) sind zur Entwicklungszeit unbekannt. Diese Variante ist ohne äußere Eingriffe nicht realisierbar.

Durch dynamisches Binden entsteht eine flexible Struktur für die Anwendungsintegration ohne feste Zuordnung des Requestor an bestimmte Provider. Das Beispiel des Web-Service zum Bereitstellen von Kundendaten könnte mit dynamischer Bindung nach dem in Übersicht 14 auf der folgenden Seite dargestellten Szenario verlaufen.

Übersicht 14

Beispiel für dynamisches Binden eines Web-Service¹⁵

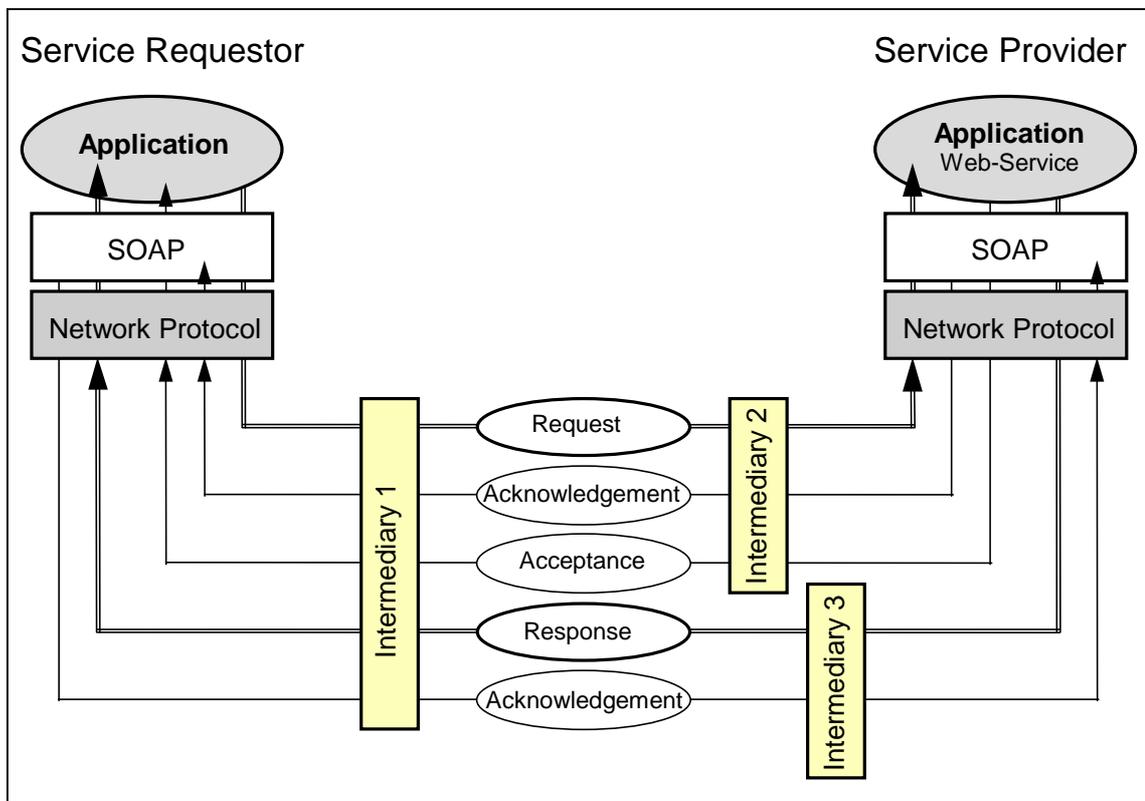
Einen Nutzen erzielt der Service-Requestor erst als Folge anforderungsgerechter Serviceaufrufe, die dem Informationsbedarf entsprechen.

4.5 Aufrufen des Service

Für den beim Serviceaufruf entstehenden Informationsaustausch (vgl. Übersicht 15) sind unterschiedliche Szenarien verbreitet.

¹⁵ Das Beispiel ist angelehnt an Schufft (2002), S. 25 f. und Samtani (2001/1), S. 5 f.

Übersicht 15

XML-Messaging auf der Basis von SOAP und Intermedieries

Über den Online-Datenaustausch entsteht eine als "lose gekoppelt" bezeichnete Anwendungsintegration durch Dienstanforderung, also auf der Ebene der Funktionsintegration. Die Interaktion der Service-Requestor-Anwendung mit dem Service wird dabei durch die angebotene Funktionalität bestimmt. Sie beruht im Normalfall auf einer synchronen Kommunikation zwischen den Anwendungen, die eine Dienstanforderung (Request) und eine Reaktion (Response) oder bei einer Einwegübertragung nur die Anforderung umfasst. Diese Kommunikation wird in der Regel vom Benutzer der empfangenden Anwendung oder systemgesteuert zum Beispiel durch Workflow-Management angestoßen.

Obwohl einzelne Kommunikationsprozesse meist nur zwei Anwendungen einbeziehen, kann beim Verwenden des Service eine Any-to-any-Topologie entstehen. Der Informationstransfer basiert auf Internet-Protokollen, vor allem dem Hypertext Transfer Protocol (HTTP), oder auf Call-Mechanismen wie CORBA im Intranet. Zum Datenaustausch wird dabei überwiegend das Simple Object Access Protocol (SOAP) für XML-basieretes Messaging verwendet.

Die den Service anfordernde Anwendung kann für andere Anwendungen selbst Web-Services anbieten. Unter anderem ist es möglich, dass sie als ein Vermittler fungiert, der beispielsweise die Qualität eines anderen Service kontrolliert. Andere Dienste, die Vermittler erbringen können, sind u.a. die Authentifizierung des Requestors, die Registrierung und Abrechnung von Servicebenutzungen, die Steigerung der Performance des Serviceeinsatzes sowie die Verknüpfung verschiedener Services zur Darstellung über eine einheitliche Schnittstelle.

Eine schwerwiegende Herausforderung an die Nutzbarkeit von Web-Services beim Business-to-Business Commerce besteht für den Service-Provider darin, die Kontinuität des Angebots im vereinbarten Service-Niveau zu gewährleisten. Dazu muss er über ausreichende Fähigkeiten beim Software-Lifecycle-Management verfügen. Das bei traditionellen Softwareprodukten so wichtige Problem der "Upgrades" wird in Veröffentlichungen zu Web-Services bislang weitgehend ignoriert.

5 Fazit und Ausblick

Der zukünftige Erfolg des Einsatzes von Web-Services hängt von geeigneten technischen und inhaltlichen Bedingungen ab. Technisch sind die Einsatzvoraussetzungen nicht ausgereift und befinden sich überwiegend noch in der Testphase. Zahlreiche Herausforderungen, um das anvisierte Ziel zu erreichen, Anwendungen und Komponenten beliebiger Hersteller über ein Intra- oder das Internet anforderungsgerecht vorteilhaft im Vergleich zu anderen IT-Lösungen zu verknüpfen, wurden bislang nicht bewältigt. Überzeugende Beispiele erfolgreicher Multi-Vendor-Lösungen nach diesem Konzept sind bislang kaum zu finden. Allerdings ermöglichen einige Initiativen zur Schaffung weiterer Standards bessere technische Voraussetzungen. Inwieweit diese Möglichkeiten zukünftig auch genutzt werden, richtet sich nach den daraus erwachsenden Konsequenzen für die Wettbewerbsposition der Anbieter.

Entscheidend für den geschäftlichen Erfolg dürfte sein, ob Services angeboten werden, die für ausreichend viele potenzielle Anwender attraktiv sind. Das ist

der Fall, wenn sie bestimmte Funktionen aufwandsärmer und schneller bieten oder diese auf andere Weise nicht anforderungsgerecht bereitstellbar sind. Sie können in unterschiedlichen Einsatzgebieten und Wirtschaftsbereichen sowie für viele betriebliche Anwendungsteilsysteme und -komponenten sinnvolle Angebote darstellen. Solche Angebote begründen entweder selbst eine Geschäftsbeziehung für diese spezielle Art von Softwaredienstleistung oder aber sie erleichtern die Informationsbereitstellung für Geschäftspartner und Mitarbeiter.

Gegenwärtig dominiert ein Vorgehen beim Erstellen von Web-Services, das von den technischen Möglichkeiten ausgeht und dann nach Anwendungsbeispielen sucht, die mit diesen Möglichkeiten realisiert werden können. Dabei kommt die Suche nach wirtschaftlich lebensfähigen Serviceangeboten oft zu kurz. Diesen Services-Konzepten drohen ähnliche Schicksale wie die der zahlreichen "Dotcom-Firmen", die eine ähnliche Strategie verfolgten. Die Anwender, vor allem in der Wirtschaft, sind immer weniger bereit, Geld für neue Informationstechnologien auszugeben, wenn sie sich dadurch keinen vertretbaren "Return on Investment" erwarten können. Daher kann auf Dauer nur derjenige erfolgreich sein, der die Suche nach geeigneten, das heißt vor allem neuartigen Anwendungen mit nützlicher Funktionalität in den Mittelpunkt stellt. Diese Lösungen zu schaffen, obliegt nicht in erster Linie den Softwareentwicklern, sondern den verantwortlichen "Sachverständigen" in den verschiedenen potenziellen Einsatzgebieten für Web-Services.

Als Grundsatz für die Eingrenzung geeigneter Anwendungen kann gelten: Nicht so viel wie möglich, sondern nur so viel wie nötig als Web-Service verarbeiten! Was in traditioneller Weise anforderungsgerecht bearbeitet werden kann, eignet sich selten als Web-Service. Gleiches gilt vorerst für Funktionen, die sehr trivial, zeitaufwendig, zeitkritisch oder datensicherungsbedürftig sind. Es sollten vorwiegend neue Informationsangebote angestrebt werden, die das Internet oder ein Intranet als vernetzte Informationsquelle verwenden. Wenig erfolgversprechend erscheinen dagegen Versuche, praktisch bewährte vorhandene Lösungen durch alternative Web-Services zu ersetzen.

Quellenverzeichnis

- Clark, M. (2001/1): Business Architecture for a Web Services Brokerage. <http://www.webservicearchitect.com/content/articles/clark01print.asp>, 01.08.2001
- Clark, M. (2001/2): Making Money out of Selling Web Services - Part I. <http://www.webservicearchitect.com/content/articles/clark02print.asp>, 22.08.2001
- Clark, M. (2001/3): Making Money out of Selling Web Services - Part II. <http://www.webservicearchitect.com/content/articles/clark03print.asp>, 29.08.2001
- Clark, M. (2001/4): The Birth of the UDDI Value Added Service Supplier. <http://www.webservicearchitect.com/content/articles/clark05print.asp>, 12.12.2001
- Fleischer, J. (2002): Einsatzmöglichkeiten XML-basierter Web Services. Unveröffentlichte Diplomarbeit, FH-Pforzheim
- Irani, R. (2001): Versioning of Web Services, Solving the Problem of Maintenance. <http://www.webservicearchitect.com/content/articles/irani04print.asp>, 08.08.2001
- McKie, S. (2001): Web Services - A Manager's Guide. Contentcan, 2001, <http://www.contentcan.com>
- Mertz, D. (2001): Understanding ebXML. <http://www-106.ibm.com/developerworks/xml/library/x-ebxml/index.html>, 02.01.2002
- Samtani, G. (2001/1): EAI and Web Services. <http://www.webservicearchitect.com/content/articles/samtani01print.asp>, 17.10.2001
- Schufft, A. (2002): Enterprise Application Integration - Web Services, Beitrag zum Diskurs 'EAI – mit welchem Konzept?' Unveröffentlichtes Manuskript FH-Pforzheim
- von Riegen, C. (2001): Mit UDDI zur nächsten Stufe des E-Business. http://www.sapinonet.net/template/tp_show_artikel.php3?ID_ARTIKEL=6582, abgerufen am 20.12.2001
- w3.org (2000): XML Protocol Comparisons, <http://www.w3org./2000/03/29-XML-protocol-matrix>, abgerufen am 06.02.2002