



FAKULTÄT FÜR **INFORMATIK**

Web-Service basiertes Ressourcenmanagement

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsingenieurwesen Informatik
eingereicht von

Marko Sirka

Matrikelnummer 0225938

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer/Betreuerin: ao. Prof. Dr. Jürgen Dorn

Wien, 19.03.2010

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Abstract

In den letzten Jahrzehnten hat die Information gegenüber der Produktion in der gesamten Wirtschaft an Bedeutung gewonnen, sodass sich viele Länder bereits zu einer sogenannten Informations- und Dienstleistungsgesellschaft entwickelt haben. Im ökonomischen Umfeld kam es somit auch zu einer wachsenden Service-Wirtschaft, was – unter anderem – auf die fortschreitende technologische Entwicklung zurückzuführen ist. Servicetechnologien, wie serviceorientierte Architekturen, sind Entwicklungen der vergangenen Jahre, denen immer mehr Aufmerksamkeit geschenkt wird; dieser Komplex wird speziell in dieser Arbeit behandelt. Ausführlich werden demnach auch einige service-nahe Themen besprochen und ergiebig darüber diskutiert, welchen Einfluss diese auf die Forschung und Unternehmenswelt haben.

Innerhalb der Arbeit wird analysiert, ob durch den Einsatz serviceorientierter Architekturen die Innovationen unterstützt werden sowie ob dadurch neue Geschäftsideen und -modelle sehr flexibel umgesetzt und später erweitert werden können. Außerdem wird erörtert, was neben der Implementierung einer serviceorientierten Architektur noch behandelt werden muss, um eine Geschäftsidee realisieren zu können.

Durch die prototypische Umsetzung eines Service-Systems mittels Web Services wurde versucht, dieser Fragestellung Genüge zu tun. Das dabei entwickelte Service-System ermöglicht auch eine anbieterübergreifende Suche, Auswahl und Buchung von Räumlichkeiten.

Weiters wurde erläutert, wie dieses System erweitert werden könnte, um es möglichst generisch für eine Fülle verschiedener Ressourcen einzusetzen. Der Entwicklungsprozess wurde mithilfe von Methoden des "Rational Unified Process" durchgeführt. Dies beinhaltet eine Prozessmodellierung, die Durchführung einer Anforderungsanalyse, die Analyse und Design des Prototyps, die eigentliche Entwicklung sowie diverse Tests.

Mittels serviceorientierter Architekturen können einzelne Services zu Kompositionen zusammengestellt werden und dadurch eine neue Funktionalität kreiert werden. Dies wird als innovativ und flexibel betrachtet, weil dadurch die Umsetzung neuer Geschäftsmodelle und eine agile Anpassungsfähigkeit ermöglicht werden.

Außerdem wurden auch Konzepte vorgestellt, wie zusätzlich eine Geschäftsautomatisierung erreicht werden kann.

Abstract (English)

In recent decades a transformation from a production-based to a service-based economy took place. In many countries the service-sector is still growing and in most of them a so-called information society established. One reason for these changes is the progressive technological development. Recent developments, such as service-oriented architectures, are getting more and more attention in the academic and organizational fields and are one of the major topics in this work. Furthermore service-near topics are mentioned and discussed.

The major research question of the work is, if service-oriented architectures support innovation and if it helps flexible realization and changes of business ideas and models. In addition, I specified what beside the implementation of a service-oriented architecture has to be done to realize a business idea.

A prototypical implementation of a service system using Web Services has been developed to answer this question. The created service system includes functions to search, select and book rooms of different providers. Moreover the work shows and explains how the system can be enhanced to handle generic resources. Methods of the "Rational Unified Process" have been used to fulfill the system development. This included process modeling, a requirements analysis, the design and the development of the prototype and several tests.

Service-oriented architectures can be used to assemble compositions of several individual services. This way a new functionality can be generated. This can be seen as innovative and flexible because it enables the implementation of new business models and an agile adaptability. In addition, within the work additional concepts have been introduced to achieve business automation.

Danksagung

Hiermit möchte ich mich vor allem bei meiner Familie bedanken, die mich ständig bei meinem Studium unterstützt hat und mir fortwährend beiseite gestanden ist. Meinen Freunde gebührt ebenfalls enormer Dank, auf die ich mich ebenfalls jederzeit verlassen konnte und mir unentwegt Kraft schenken. Auch meinem Betreuer bin ich zu Dank verpflichtet, der für jede Frage meiner Fragen ein offenes Ohr und stets gute Anregungen und Ratschläge parat hatte.

Erklärung zur Verfassung der Arbeit

Sirka Marko
Fernkorngasse 55/11
A-1100 Wien

"Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Wien, 25.03.2010

Unterschrift

Inhaltsverzeichnis

1. Einleitung.....	1
1.1 Motivation.....	1
1.2 Fragestellung.....	3
1.3 Vorgehensweise.....	4
2. Grundlagen.....	6
2.1 Der Servicebegriff.....	8
2.2 Dienstleistungsforschung.....	11
2.2.1 Service Science.....	12
2.2.2 Service Dominant Logic.....	19
2.2.3 Service Engineering.....	22
2.3 Servicetechnologien.....	27
2.3.1 Service-orientierte Architekturen.....	29
2.3.2 Web Services.....	37
2.3.3 Ontologien & Semantic Webservices.....	50
2.4 Methoden zur Qualitätssicherung von Services.....	55
2.4.1 Kundenorientierte Messungsmethoden.....	57
2.4.2 Unternehmensorientierte Messungsmethoden.....	58
2.4.3 Service Level Agreements.....	62
2.5 Leistungsbündel und Virtuelle Unternehmen.....	65
3. Systementwurf und Planung des Prototyps.....	71
3.1 Der klassische Buchungsvorgang.....	72
3.2 Vorteile aufgrund des Service-Systems.....	74
3.2.1 Kundenvorteile.....	74
3.2.2 Vorteile für Anbieter.....	74
3.3 Glossar.....	77
3.4 Entwurfdetails.....	78
3.4.1 Grundlagen.....	78
3.4.2 Anforderungen an das System.....	80
3.4.3 Service-Beschreibungen.....	81
3.5 Anwendungsfälle.....	84
3.5.1 Überblick.....	84
3.5.2 Authentifizierung.....	84
3.5.3 System verwalten.....	85
3.5.4 Ressourcen reservieren und buchen.....	86
3.6 Datenmodell.....	89
3.6.1 Userverwaltung.....	89
3.6.2 Serviceverwaltung.....	89
3.6.3 Raumverwaltung.....	90
3.6.4 Kompositionsverwaltung.....	92

3.6.5	Reservierungs- und Buchungsverwaltung.....	92
3.6.6	Gesamtmodell	93
3.7	Abläufe im System	94
4.	Systembeschreibung	96
4.1	Softwarearchitektur	96
4.1.1	Entwicklungsumgebung	96
4.1.2	Datenbank	97
4.1.3	Web Services mit PHP	97
4.1.4	Testumgebung des Prototyps.....	98
4.2	Umsetzung des Prototyps	99
4.2.1	Systemfunktionen	99
4.2.2	Das Anbietersystem.....	99
4.2.3	Das Service-System.....	105
5.	Evaluierung	109
5.1	Zusätzliche funktionale Anforderungen	111
5.1.1	Automatische Portfolioerstellung	111
5.1.2	Zusätzliche Servicebeschreibungen	111
5.1.3	Erweitertes Kompositionsmodell	112
5.1.4	Eingliederung bestehender Web Services zur Ressourcenbuchung.....	112
5.1.5	Vokabulare	113
5.2	Bestehende Lösungen.....	113
5.3	Standardisierung	114
5.4	Nichtfunktionale Anforderungen.....	114
5.4.1	Abstraktion	114
5.4.2	Usability	115
5.4.3	Serviceverträge und Servicequalität	115
5.4.4	Sicherheit & Kontrolle im System.....	117
5.4.5	Fehlerbehandlung	120
5.5	Methoden zur Überprüfung von Web Services	121
5.5.1	Analysetechniken	121
5.5.2	Testtechniken	121
5.5.3	Monitoring.....	121
5.6	Transaktionen	122
5.7	Zahlungsverfahren	126
6.	Schlussfolgerungen, Fragen und Ausblicke	127
7.	Literaturverzeichnis	130
8.	Abbildungsverzeichnis	134
9.	Tabellenverzeichnis	135

1. Einleitung

1.1 Motivation

In den letzten Jahrzehnten hat sich in den führenden Wirtschaftsnationen ein klarer Wandel abgezeichnet. Aufgrund der fortschreitenden technologischen Entwicklung und der gestiegenen Produktivität, sowohl im Agrarsektor als auch im Bereich der Industrie, kam es zur Verschiebung der Wirtschaftsaktivitäten auf das Gebiet der Dienstleistung, wodurch in der gesamten Wirtschaft die Information gegenüber der Produktion an Bedeutung gewonnen hat. Die daraus resultierende Informations- beziehungsweise Wissensgesellschaft betreibt ihre Wertschöpfung vorrangig durch die Erschaffung, Verarbeitung und Weitergabe von mannigfachen Informationen. Der Handel mit Dienstleistungen und digitalen Produkten steht dabei im Vordergrund. Der Funktionswandel von Arbeit innerhalb der hochentwickelten Länder ist ein komplexer Prozess, der oft mit der industriellen Revolution verglichen wird. Beliebte Schlagwörter wie Individualisierung, Globalisierung, Dezentralisierung, unternehmenübergreifende und -interne Netzwirkbildung, Informatisierung und Tertiarisierung werden oft als Ursache dafür genannt. ([Hrad08] et al.)

Unter Tertiarisierung wird die steigende Dominanz des Dienstleistungssektors¹ gegenüber dem primären² und dem sekundären³ Sektor verstanden. In hochentwickelten Volkswirtschaften beträgt der Anteil des Dienstleistungssektors mehr als 70 Prozent am BIP. Der Anteil aller Angestellten, die in diesem Sektor arbeitet, ist etwa gleich hoch, während der Beschäftigungsanteil im Produktionssektor rückläufig ist. Nur ein minimaler Prozentsatz der Menschen in den modernsten Ländern arbeitet noch im Agrarsektor. Informationsberufe dominieren heutzutage deutlich gegenüber anderen Berufssparten. Immer mehr Personen beschäftigen sich mit Produktion, Verarbeitung und Verteilung von Informationen. Der technologische Wandel sowie Veränderungen am Markt haben gravierende Auswirkungen auf die bestehenden Arbeits- und Organisationsformen. Die Gesellschaft wird ständig auch durch die zeit- und ortsunabhängigen Märkte beeinflusst. Immer mehr Unternehmen des industriellen Sektors offerieren auch ergänzende und alleinstehende Dienste, oder stellen kundenorientierte Lösungen mit einem überwiegenden Anteil von Diensten zur Verfügung.

Gefragt sind Arbeitsbedingungen, die ein hohes Maß an Selbstständigkeit gewähren und Berufs- und Privatleben besser in Einklang bringen. Die Globalisierung der Markt- und Wettbewerbsbeziehungen bewirkt eine Globalisierung der Arbeitsformen und -kontakte. Dies erfordert das Zusammenarbeiten trotz großer räumlicher Distanz und unterschiedlicher Zeitzonen. Sprachliche und kulturelle Grenzen müssen überwunden werden. [MeSt05]

Die Orientierung der Unternehmen verändert sich zunehmend von den Produkten zum Kunden. Der Fokus ist service- und kundenorientiert, wodurch der individuellen Leistungsvorstellung der Konsumenten eine große Relevanz eingeräumt wird.

Die Anwendung von Informations- und Kommunikationstechnologien dient dabei sowohl als Erweiterung der Handlungsfähigkeit einzelner Personen und Organisationen, als auch zur Erweiterung grenzüberschreitender Kontakte. Viele Unternehmen sind heutzutage im Electronic

¹ Auch tertiärer Sektor genannt.

² Beinhaltet Urproduktion wie Agrarwirtschaft und Ressourcenabbau.

³ Produzierende und verarbeitende Betriebe sowie die Industrie.

Business tätig, wobei die Geschäftsprozesse vor allem im Internet ausgeführt und Kundenbeziehungen mit Hilfe von elektronischer Informations- und Kommunikationsmittel realisiert werden.

Im Zuge der Entwicklung hin zu einer vernetzten Informationsgesellschaft kann eine Veränderung der Marktsysteme beobachtet werden. Der physische Marktplatz mit physischen Rohstoffen, Produkten und Ressourcen bleibt ja weiterhin bestehen. Hier sind auch physische Wertschöpfungsprozesse notwendig, um materielle Güter zu beschaffen, zu entwickeln und zu verteilen. Durch die Entwicklung des Internets wird der physische Marktplatz allerdings ergänzt, und zwar durch verschiedene digitale Produkte, Online-Handel, etc. Die Unternehmen müssen sich in einem Geschäftsmodell entscheiden, ob und wie der physische und der elektronische Marktraum bearbeitet werden. [MeSt05]

Die Unternehmen müssen entscheiden, ob sie die bestehende Produktpalette durch digitale Informations- und Servicedienste ergänzen, ablösen oder erweitern. Die Digitalisierung ist allerdings nicht in allen Branchen gleich fortgeschritten, da diese vom Markt und von der Akzeptanz der Konsumenten abhängt. Damit auf individuelle Kundenwünsche eingegangen werden kann, ist es notwendig, die angebotenen Leistungen dynamisch und kosteneffizient anpassen zu können. Eine Unterteilung in verschiedene Produktversionen ist dabei zielführend. Um sich am Markt halten zu können, müssen Unternehmen ihre angebotenen Produkte und Dienstleistungen laufend überarbeiten und erweitern. Dies bedeutet kürzere Innovationszyklen und eine verstärkte Konzentration auf Produktentwicklung und Leistungserweiterung.

Des Weiteren ist eine Flexibilisierung der Geschäftsmodelle notwendig, um wettbewerbs- und konkurrenzfähig zu bleiben, da es immer schneller und häufiger zu Veränderungen der Marktsituationen kommt. Dabei handelt es sich um komplexe Herausforderungen, die bewältigt werden müssen, weshalb die Forschung in diesem Bereich immer mehr Relevanz erlangt.

Ferner stellt sich die Frage, wie Informationstechnologien möglichst effizient eingesetzt werden können, um die Geschäftslogik der Unternehmen zu unterstützen, oder ob mit deren Einsatz neue Produkt- und Leistungsvarianten geschaffen werden können. Kooperationen und Auslagerung von Teilbereichen an externe Unternehmen (sog. Outsourcing) sind in diesem Bereich immer häufiger. Gründe dafür: Einerseits der ständig steigende Spezialisierungsgrad der Angestellten, welcher notwendig ist, um die immer komplexeren Aufgaben und die speziellen Kundenwünsche lösen zu können. Andererseits können sich Unternehmen bei solchen Gegebenheiten auf ihr Kerngeschäft konzentrieren und können andere, jedenfalls notwendige, Tätigkeiten an externe Anbieter übergeben. Ein gebräuchliches Beispiel hierfür ist, unter vielen anderen die Auslagerung der Buchhaltung.

Trotz dieser dauerhaften Gesellschaftsentwicklung bestand lange Zeit ein erheblicher Forschungsmangel in der Dienstleistungsinnovation, und zwar weil sich viele Wissenschaftsdisziplinen noch immer auf den industriellen Wirtschaftssektor konzentrierten beziehungsweise die Notwendigkeit der dienstleistungsspezifischen Forschung nicht erkannten. Mittlerweile haben sich die verschiedensten Forschungsdisziplinen auf diesen Bereich konzentriert. Von verschiedenen Seiten wird jedoch der Aufbau einer eigenständigen Forschungsdisziplin gefordert.

Das Vorhaben der Etablierung einer eigenständigen Disziplin kann mit dem Begriff Service Science, Management and Engineering (SSME) bezeichnet werden, bei der es vor allem um die Integration von Erkenntnissen aus der für die Dienstleistungsforschung relevanten Bereichen geht. Durch die Vereinigung von Methoden und Werkzeugen aus betriebswirtschaftlichen und technologisch-orientierten Disziplinen soll es ermöglicht werden, effiziente Service Systeme zu

konstruieren und weiterzuentwickeln. Im Kapitel "Services Science" wird auf SSME noch genauer eingegangen.

Die Dienstleistungsforschung ist eindeutig ein interdisziplinärer Zweig, bei dem alle beteiligten Disziplinen in Kooperation voneinander profitieren könnten. Neue Erkenntnisse sollen ermöglichen, einheitliche Methoden zur Entwicklung, Erstellung, Erweiterung und Vermarktung von Dienstleistungen zu entwickeln. Weiters stellt sich die Frage, ob Dienstleistungen und Dienste in verschiedenster Form standardisierbar und deren Ergebnisse messbar sind, um Dienstempfängern bereits vor dem Konsum eine Beurteilung abliefern zu können. Die Risikominimierung ist nämlich für Konsumenten eine der wichtigsten Themen bereits vor einem Kaufabschluss, um sicherzustellen, dass wirklich die Leistung erbracht wird, die auch gefordert wird. Möglichkeiten dazu werden in dem Kapitel "Methoden zur Qualitätssicherung von Services" diskutiert.

Bei einer neuen Geschäftsidee stellt sich für den Unternehmer immer die Frage, wie er diese bestmöglich umsetzen kann bzw. welche Teile davon überhaupt realisierbar sind. Es gilt also eine Überleitung von der eigentlichen Geschäftsidee zu dessen veräußerbaren Umsetzung zu finden. Die Modellierung der Geschäftsidee und die Gliederung in einzelne Geschäftsprozesse ist dabei eine mögliche Herangehensweise. Daraus ergeben sich einzelne Geschäfts-Services, die angeboten werden können. In weiterer Folge muss entschieden werden, welche dieser Geschäfts-Services überhaupt langfristig rentabel sind und angeboten werden sollen. Dabei muss nachfolgend untersucht werden, welche dieser Geschäfts-Services mit Informationstechnologien unterstützt oder sogar komplett automatisiert werden können.

Häufig lässt sich heutzutage mittlerweile durchaus ein Großteil eines Geschäfts-Services automatisiert durchführen, in den meisten Fällen müssen jedoch noch einige manuelle Tätigkeiten durchgeführt werden, um dies zu tun. So ist es beispielsweise zwar möglich komplett automatisiert einen Raum zu vermieten, jedoch die anschließende Reinigung dieses oder die persönliche Betreuung vor Ort sind müssen noch immer "klassisch" durchgeführt werden.

1.2 Fragestellung

Als grundlegende Fragestellung dieser Arbeit resultiert daraus, ob durch den Einsatz von Web Services im betrieblichen Umfeld bestehende Geschäftsmodelle flexibel angepasst und erweitert werden können, beziehungsweise, ob damit dynamisch völlig neue Geschäftsmodelle erstellt werden können.

Anders formuliert lautet die Forschungsfrage: Eine serviceorientierte Architektur unterstützt die Innovationen, weil dadurch sehr flexibel neue Geschäftsideen und -modelle umgesetzt werden können.

Weiters wird hinterfragt, ob ein Dienstleistungsgeschäft durch ein Web Service implementiert werden kann, oder ob ein Web Service ein Dienstleistungsgeschäft sein kann.

Einerseits gibt es auf technischen Ebene serviceorientierte Architekturen und andererseits im ökonomischen Umfeld die wachsende Service-Wirtschaft, bei der immer nach neuen Innovationen gesucht werden muss, um konkurrenzfähig zu bleiben. Die Implementierung einer serviceorientierten Architektur reicht jedoch nicht aus, um eine Geschäftsidee umzusetzen. Daraus stellt sich auch die Frage, was als Bindeglied zwischen Geschäftsidee und serviceorientierter Architektur benötigt wird, um diese erfolgreich umsetzen zu können.

1.3 Vorgehensweise

Der Inhalt der Diplomarbeit stellt einen Entwurf und die Entwicklung eines generischen Service-Systems dar, welches eine detaillierte, anbieterübergreifende Suche, Auswahl und Buchung von Ressourcen ermöglicht. Anhand dieses Systems soll gezeigt werden, wie es heutzutage möglich ist, die Geschäftsprozesse beziehungsweise Geschäftsmodelle diverser Organisationen mithilfe von Web Services zu erweitern, zu verbessern, zu beschleunigen und einfacher zu gestalten.

Die Aufgabe dieses Systems besteht unter anderem darin, Anwendern betriebswirtschaftlich effiziente Portfolios von Ressourcen entsprechend den Suchkriterien anzubieten und diese buchbar zu machen. Die zeitliche Ersparnis gegenüber konventionellen Methoden steht ebenfalls im Vordergrund. Das System wird für verschiedene Typen von Ressourcen kategorisierbar sein. Die genaue Beschreibung des Systementwurfs beziehungsweise der prototypischen Umsetzung des Service-Systems selbst erfolgt in späteren Kapiteln.

Diese Diplomarbeit beinhaltet neben der Implementierung des Systems klarerweise auch Recherchetätigkeiten und die Analyse aktueller Technologien und Methoden, die bei der Entwicklung des Service-Systems eine gewisse Rolle spielen. Geltende Standards, Methoden, Normen, Technologien und Trends, die bei einer derartigen Implementierung relevant sind, werden in der Arbeit analysiert, diskutiert und bewertet, um ein Basisverständnis für das Umfeld des Systems zu schaffen.

Weiters werden innerhalb dieser Untersuchungen auch einige Anpassungen, Verbesserungsvorschläge und mögliche Erweiterungen aufgezeigt sowie erläutert bzw. Vor- und Nachteile gewisser Standards diskutiert.

Im Anschluss an die Einleitung folgt die Beschreibung und Analyse des aktuellen Forschungsstandes (dem sog. "state of the art"), bei dem vor allem servicenahe Themen (sowohl auf technischer als auch auf betriebswirtschaftlicher Ebene), wie Dienstleistungsforschung (z.B. Services Science), Service-orientierte Architekturen (SOA), Webservices, Methoden zur Qualitätssicherung (z.B. Service Level Agreements) unter die Lupe genommen werden.

Darauf folgt der Systementwurf, bei dem auf den Planungsprozess zur Erstellung des Prototyps des Service-Systems eingegangen wird, wobei wesentliche Punkte davon genauer beschrieben werden. Methoden, die bei einer heute gängigen Softwareimplementierung eingesetzt werden wurden ebenfalls angewendet und deren Ergebnisse in dem Kapitel präsentiert. Beispielsweise wurden im Rahmen einer Anforderungsanalyse einige Anwendungsfälle und ein Domänenmodell erstellt.

Die anschließende Systembeschreibung beinhaltet eine genaue Erläuterung des prototypisch implementierten Webservice-Systems. Aufgrund der Informationen aus der Entwurfsphase wurde ein Datenbankmodell erstellt. Die Schnittstellendefinitionen und Klassendiagramme zum Entwurf systemwichtiger Klassen der Geschäftslogik sind ebenfalls wichtige Elemente, die näher beschrieben werden. Weiters wird auf die verwendete Softwarearchitektur eingegangen und angeführt, worauf man beim Einsatz dieser achten muss, wenn man die Entwicklung eines ähnlichen Systems durchführt.

In der anschließenden Evaluierung wird erläutert, welche zusätzlichen Überlegungen und Möglichkeiten bei der Entwicklung eines vergleichbaren Systems eine Rolle spielen, beziehungsweise welche Vorkehrungen getroffen werden müssen und wie das bei ähnlichen Systemen vergleichsweise realisiert wurde. Eine große Relevanz haben auch die nichtfunktionalen Anforderungen wie Zuverlässigkeit (Systemreife, Wiederherstellbarkeit,

Fehlertoleranz), Aussehen und Handhabung (Look and Feel), Benutzbarkeit (Verständlichkeit, Erlernbarkeit, Bedienbarkeit), Leistung und Effizienz (Antwortzeiten, Ressourcenbedarf), Betrieb und Umgebungsbedingungen, Wartbarkeit und Änderbarkeit (Analysierbarkeit, Stabilität, Prüfbarkeit). Klare Bestimmungen, wie diese Themen im System bzw. während der Implementierung behandelt werden, sind ebenfalls unerlässlich. Auch kritische Fragen der Fehlerbehandlung und Kontrolle im System werden beantwortet.

Um das System auch ausreichend vor Missbrauch zu schützen, erhält auch das Thema Sicherheit einen großen Stellenwert, da es gerade beim Einsatz solcher Technologien verschiedenste Möglichkeiten gibt, dieses Thema zu behandeln. Nach wie vor werden lang bekannte Sicherheitslücken diverser Architekturen (vor allem im Web) von vielen Entwicklern nicht bedacht.

Als möglicher Erweiterung des Prototyps werden beispielsweise auch Modelle zur Preisgestaltung aufgeschlüsselt, die eingesetzt werden, um Rabatte oder Kontingente zu berücksichtigen. Auch die Abwicklung der Bezahlung nach einer erfolgreichen Buchung ist ebenfalls ein wesentlicher Punkt. Ein Ausblick auf weitere mögliche Suchkriterien – beispielsweise mittels der Verwendung von externen Diensten – ist ebenfalls darin enthalten.

Mittels der Evaluierung soll auch die Verwend- und Einsetzbarkeit des Systems getestet und überprüft werden. Sie dient weiters dazu, zu zeigen, ob die definierten Ziele erfüllt wurden. Weiters wird gezeigt, wie es möglich ist, solche Systeme umfangreich zu testen.

Innerhalb eines abschließenden Resümees werden die wichtigsten Punkte nochmals zusammengefasst. Ferner wird ein Ausblick auf mögliche zukünftige Entwicklungen und Tendenzen in diesem Bereich gewährt.

2. Grundlagen

Dieses Kapitel dient der Beschreibung des Forschungsstands aus sowohl betriebswirtschaftlicher als auch technischer Sicht. Ein wichtiger Punkt ist auch die Differenzierung des Servicebegriffs aus verschiedenen Betrachtungswinkeln, da es, je nach Disziplin, Unterschiede bei der Interpretation gibt.

Das spektakuläre Wirtschaftswachstum des Servicebereichs in den letzten fünfzig Jahren ist ersichtlich, sowohl durch die BIP-Statistik der einzelnen Nationen als auch aufgrund der jährlichen Berichte der erzeugenden Unternehmen, welche steigende Serviceeinkünfte belegen. Sowohl die steigende Anzahl an Dienstleistungsunternehmen, als auch der steigende Anteil an Erlösen von sogenannten Nicht-Servicebetrieben zeigen die veränderte Situation der Wirtschaft. Trotz des signifikant messbaren Wachstums der Servicewirtschaft gibt es keine allgemein anerkannte Definition von Service. Außerdem fehlt es an Methoden für die Messbarkeit von Serviceproduktivität, Qualität, Spezifikationen und Innovation, welche erst im Frühstadium der Entwicklung stehen. [Spoh08]

Durch die in den letzten zwei Jahrzehnten stetige Entwicklung des Internets und aufgrund der weltweiten Verbreitung von Computertechnologien in Unternehmen und Haushalten konnte sich ein elektronischer Markt bilden und rasant weiterentwickeln. Dadurch entstehen einige neue Vorteile und Möglichkeiten für Unternehmen und Konsumenten. Zwischenhändler können durch die neuen technischen Möglichkeiten leichter umgangen werden, wodurch eine höhere Gewinnspanne erzielt beziehungsweise ein Preisnachlass an die Konsumenten weitergegeben werden kann.

Die damit verbundene Anwendung von internetbasierten Kommunikationstechnologien ermöglicht einen direkten Zugang zum Kunden, was wiederum die weitere Kundenbindung und -beziehung fördert, und den Organisationen eine adäquatere Analyse des Kundenverhaltens ermöglicht. Dadurch kann auf die Kundenpräferenzen eingegangen werden, indem Kundenresonanz verwendet werden kann, um das Produkt- und Dienstleistungsportfolio bedürfnisgerecht anzupassen und zu erweitern.

Daraus lassen sich, abhängig von diversen Faktoren (Realisierbarkeit, Zahlungsbereitschaft der Kunden, etc.), kundenindividuelle Produkte und Dienstleistungen anbieten. [MeSt05]

Diese Entwicklungen hatten natürlich auch für den Forschungsbereich große Auswirkungen. Trotz einer hohen Beitragsanzahl im Bezug auf Services aus den verschiedensten Disziplinen, sehen viele nach wie vor einen großen Forschungsmangel auf diesem Gebiet.

Beispielsweise schreibt Walter Ganz in dem Beitrag *Research Needs in the European Service Sector* [Ganz08]: "Es gibt limitiertes Wissen, wie servicegetriebene Prozesse systematisch entwickelt, entworfen und modelliert werden können."

Er sieht auf folgenden Gebieten eine spezielle Nachfrage an Forschung:

- Serviceinnovation: Mangel an Wissen über die Verstärkung der Innovationsmöglichkeit und Flexibilität von Serviceanbietern.
- Internationalisierung von Services: Mangel an Wissen über den Einfluss und die Auswirkung von kulturellen Faktoren auf Serviceangebote, die Nachfrage und auf die Servicequalität.

- Transformationsprozesse in der Wirtschaft: Mangel an Wissen über den "Tertiarisierungs-Prozess" und ein Wechsel der Orientierung der Güterproduktion zur Servicebereitstellung.
- Technologie und Services: Mangel an Wissen über die Umsetzbarkeit, den Einfluss und die Auswirkungen durch die Verwendung von Technologie auf Serviceangebote, die Forderung nach und die Qualität von Services.
- Entwurf von Servicearbeit: Mangel an Wissen über menschliche Aspekte und soziale Interaktion in Verbindung mit Services, speziell in hochqualifizierter Wissensarbeit.

Bei folgenden Punkten sieht dieser Autor besonders in Europa Handlungsbedarf:

- Verstärkung des Forschungsprofils von Services: Defizite in der Sichtweise von Services als relevantes Forschungs- und Entwicklungsthema reduzieren.
- Internationale Führungsposition verstärken: Absicherung des internationalen Aufholpotenzials sowie setzen von (neuen) Kernkompetenzen.
- Intensivierung der Forschungs- und Entwicklungs-Kooperation mit der Praxiswelt: Wissen über Nachhaltigkeit und Kontinuität von Forschungsergebnissen in gemeinsamer Anstrengung mit der Praxiswelt entwickeln.
- Erstellung von Service-Kompetenzzentrum: Bündelung von Ressourcen in themenspezifische Kompetenznetzwerke.

Mittlerweile konnten sich schon diverse Forschungsansätze etablieren, die sich mit oben genannter Problematik beschäftigen. Innerhalb dieser Arbeit wird in diesem Kapitel nachfolgend näher auf die Themen "Service Science" und "Service-Dominant-Logic" eingegangen.

Service Science, Management and Engineering (SSME) umfasst einen interdisziplinären Ansatz zur Analyse, Gestaltung und Implementierung von Service-Systemen. [Dorn09]

Die Service-Dominant-Logic (SDL) repräsentiert das Abweichen von der traditionellen, grundlegenden güterdominanten Logik des Handels. In der traditionellen Logik liegt der Fokus auf Güter und Handelswaren, wobei Services (Dienstleistungen) eine spezielle Art von Gütern repräsentierten. Die neue Logik repräsentiert einen Übergang von der güterzentrierten Sichtweise, die physische Ressourcen in den Mittelpunkt stellt (hauptsächlich greifbare und inaktive Ressourcen) hin zu einer servicezentrierten Sichtweise, die vor allem dynamische Ressourcen betont (immaterielle, sogenannte intangible, dynamische Ressourcen, die mit anderen Ressourcen agieren wie beispielsweise Wissen, Fähigkeiten und Kompetenzen). [LuVa04]

Um Informationen liefern zu können, wie sich technische Services (z.B. Webservices) zur Unterstützung kaufmännischer Dienstleistungen einsetzen lassen, werde ich innerhalb dieses Kapitels auf die technologiebasierten Themen "Web Services", "Service-orientierte Architekturen", "Ontologien und Semantic Webservices" und "Service Bundling" genauer eingehen.

Einen weiteren wichtigen Punkt stellen Methoden zur Qualitätssicherung von Services dar, bei dem die Themen "Service Level Agreements" und Messmethoden im Vordergrund stehen.

2.1 Der Servicebegriff

Der Servicebegriff muss differenziert betrachtet werden, da er in vielen unterschiedlichen Disziplinen verwendet wird. Beispielsweise bestehen Unterschiede aus wirtschaftlicher und technischer Sicht. Dies ist in Bezug auf das Service-System wichtig, um ein klares Verständnis zu schaffen, in welcher Form der Servicebegriff unterschiedlich eingesetzt wird.

Service ist gegenwärtig ein Ausdruck, der mit den unterschiedlichsten Bedeutungen – je nach Umstand und abhängig davon, wer ihn verwendet – verbunden wird. Verschiedene Begriffe, die das Wort "Service" beinhalten (Webservice, eService, etc.) werden in dem jeweiligen Fachgebiet nur mit dem Wort "Service" benannt. Bei diesen abweichenden Interpretationen können leicht Probleme bei der Kommunikation auftreten, wenn Personen aus verschiedenen Bereichen involviert sind. [BaGO04]

Im deutschsprachigen Raum wurde der Service-Begriff – er wird jedoch oft auf Leistungen im Bereich der Montage, Reparatur oder Wartung beschränkt [Bull06] – mittlerweile in den Wortschatz übernommen, da unter anderem in manchen Fachbereichen die Übersetzung "Dienstleistung" beziehungsweise "Dienst" nicht adäquat wäre, oder englischsprachige Entwicklungen häufig einfach in den Wortschatz integriert wurden.

Baida, Gordijn, Omelayenko unterteilen den Servicebegriff im Artikel "A Shared Service Terminology for Online Service Provisioning" in drei Kategorien, die den Interpretationen der Managementlehre, der Informatik und der Informationswissenschaft entsprechen. [BaGO04]

In der Managementlehre beziehungsweise in den Wirtschaftswissenschaften wird Service als eine betriebswirtschaftliche Geschäftsaktivität (Handlungen, Prozesse und Leistungen – im Wesentlichen intangible) betrachtet, die von einer Partei einer anderen für eine gewisse Leistung angeboten wird. [ZeBi96][Kotl88]

In der Informationswissenschaft ist ein Service eine komplexe (oder auch einfache) Aufgabe innerhalb einer Organisation, die im Interesse eines Kunden ausgeführt wird [OSEH02].

In der Informatik bezieht sich ein Webservice auf eine programmierbare, selbstbeschreibende, gekapselte und lose gebundene Funktion, auf welche über das Internet zugegriffen und diese dann anschließend ausgeführt wird [W3C04].

In Bezug auf "Services Science" diskutiert Bernd Stauss [Stau08] ebenfalls unterschiedliche Service-Bedeutungen und begrenzt die Interpretation von Service auf fünf relevante Perspektiven:

- Sektorale Perspektive:

Die sektorale Perspektive begründet die Signifikanz von Service Science auf dem Faktum, dass der Sektorservice mit 70-80 % des Bruttoinlandsprodukts und der Erwerbstätigen der größte Sektor der entwickelten Volkswirtschaften ist. Dieses Leitbild beschreibt ein institutionelles Verständnis von Service. Die Verwendung von Service in diesem Sinne bedeutet, dass Serviceforschung in Richtung aller Industrien orientiert ist, die dem Sektorservice der Wirtschaft zugerechnet werden können. Dies beinhaltet den Einzelhandel, Finanzdienstleistungen, Transport, Telekommunikation und andere.

- **Service Industries Science:**
Diese Perspektive konzentriert sich in der Forschung nicht auf alle Serviceindustrien, sondern behandelt nur ausgewählte Industrien (beispielsweise solche mit hohem Wachstumspotential – wie Business Services, Information Services, Medien- und Telekommunikationsservices, Gesundheitsvorsorge-Services oder Öffentliche Services). Diese Perspektive hat den Vorteil, dass der Fokus auf ausgewählte Industrien einen restriktiveren Forschungsinhalt darstellt, obwohl dies die Wachstumsrate der Serviceindustrie beeinflussen wird.
- **Service Transformation Science:**
Services bekommen auch in traditionell industriellen und produktorientierten Unternehmen immer mehr Bedeutung. Die technischen Kernprodukte der im Wettbewerb stehenden Unternehmen sind oft nahezu austauschbar, sodass die Qualität eines produktbezogenen Service den Schwerpunkt eines Angebots ausmacht. Industrieunternehmen erzielen aus diesem Grund einen immer größeren Absatz und Profit aus Services als mit den Erzeugungsprodukten selbst. In den traditionellen Serviceindustrien gibt es eine starke Tendenz industrielle Prinzipien und Methoden zu übernehmen. Die vorhandene Forschung konzentriert sich hauptsächlich auf die industrielle Perspektive mit dem Fokus auf Kosten, Produkt, Effizienz und Technologie.
- **Service Topic Science:**
Der Begriff "Service Topic Science" fordert, dass sich Service Science mit spezifisch selektierten Themen (Innovation, Technologie, Optimierung, Standardisierung und Informationsintegration) beschäftigen soll. Services Science wird in diesem Fall nicht als ein spezifisches Serviceverständnis definiert; stattdessen soll der Transfer von relevanten betriebswirtschaftlichen Themen in Bezug auf Services ins Zentrum rücken.
- **Service Management Science:**
Der Hauptfokus dieser Serviceforschung liegt auf den Problemen, die aufgrund der spezifischen Charakteristiken von interaktiven Services (Intangibilität, Kundenbeteiligung im Serviceproduktionsprozess) entstehen, unabhängig von Branche und Wirtschaftssektor, die das Service anbieten. *Service Management Science* beschäftigt sich mit den Erwartungen, Sichtweisen und dem Verhalten von Kunden und Angestellten sowie den Grundvoraussetzungen für erfolgreiche Interaktionen, und teilt seine konzeptionelle Basis mit der bestehenden Service Management Forschung.

Aus wirtschaftlicher Sicht beschreiben und definieren viele Autoren Services im Sinne von Dienstleistungen mit ähnlichen grundlegenden Charakteristiken. Jim Spohrer fasst in dem Werk *Services Science, Management Engineering and Its Relation to Academic Disciplines* die unterschiedlichen Autorenbeschreibungen zusammen. Als wichtigste Dienstleistungsmerkmale identifiziert er dabei die Nichtgreifbarkeit (Intangibilität), die Heterogenität (der Dienstleistungsprozess läuft detailliert nie gleich ab), die Simultanität (Produktion und Konsum finden gleichzeitig statt und sind dabei untrennbar), Vergänglichkeit (nicht lagerbar) und Kundenpartizipation. Kundenpartizipation wird von manchen Autoren auch als Koproduktion bezeichnet, wobei damit nicht unbedingt die Mitarbeit an einem Produktionsprozess gemeint ist, sondern die Unterstützung durch Informationen und/oder Eigentum. [Spoh08]

Ein erster Schritt, um Missverständnisse um den Service-Begriff herum zu vermeiden, ist ein Bewusstsein darüber zu entwickeln, dass mehrere Interpretationen zu diesem Ausdruck in den unterschiedlichen Forschungsgemeinden (sogar in verwandten) existieren. [BaGO04]

2.2 Dienstleistungsforschung

Manche könnten erwarten, dass sich die wirtschaftliche Relevanz von Dienstleistungen in den akademischen Systemen durch einen starken Fokus auf Dienstleistungsforschung und -bildung widerspiegelt. Das ist nicht der Fall. Trotz einem starken Anstieg an dienstleistungsrelevanten Dokumenten, Analysen, Journalartikeln und internationalen Konferenzen hat sich noch bisher keine unabhängige und eigenständige Forschungsdisziplin etabliert. [SEKL08]

In der Literatur herrscht sowohl auf Seiten der Forschungs- als auch der Unternehmerwelt Einigkeit, dass eine Ausweitung der Dienstleistungsforschung notwendig ist. Uneinigkeit gibt es jedoch darüber, auf welche Art und Weise dies vollzogen werden soll und kann.

Services Science, Management and Engineering (SSME) verbindet Disziplinen aus dem Bereich des Managements, der Wissenschaft und des Ingenieurwesens miteinander, um Aufgaben von Organisationen nutzenmaximierend zu gestalten und beschreibt somit einen interdisziplinären Ansatz zur Analyse, Gestaltung und Implementierung von Service-Systemen. Dabei sind Service-Systeme komplexe Gebilde, in welchen bestimmte Gruppierungen von Menschen oder Organisationen tätig sind, um Wertschöpfung für andere zu Verfügung zu stellen. Außerdem fordert SSME Akademien, die Industrie und Regierungen dazu auf, fokussierter, bedachter und systematischer bei der Innovation des Dienstleistungssektors vorzugehen. [Dorn09]

Die verschiedenen Teilnehmer am Bildungsprozess des SSME vertreten unterschiedliche transdisziplinäre Leitbilder, wobei alle beträchtliche Forderungen an die Partner (Universitäten und Unternehmerwelt) stellen. Unternehmen müssen die Relevanz von Service akzeptieren und ihre Forschungsaktivitäten daran anpassen. Universitäten müssten Service zu einem Hauptforschungszweig deklarieren und transdisziplinäre Strukturen in der Forschung und Ausbildung aufbauen, welche eine Kooperation zwischen den unterschiedlichen Abteilungen, Schulen und Universitäten ermöglicht. Unternehmen müssen Angestellte rekrutieren und trainieren, um sie für die Kollaboration zu qualifizieren. Administrative Regeln müssen dabei verändert werden, um unbürokratische Kooperation zu ermöglichen. Um Service Science institutionalisieren zu können, ist eine leistungsfähige, gemeinsame Initiative der Unternehmenswelt, der Politik und der Universitäten vonnöten. Diese Initiative muss attraktive finanzielle Anreize beinhalten, welche hauptsächlich von der Unternehmerwelt ausgehen können. Außerdem ist ein notwendiger struktureller Wandel der Universitäten sicher zu stellen. [Stau08]

SSME stellt gerade deshalb den ersten wichtigen Punkt innerhalb dieser Arbeit dar und wird anschließend genauer betrachtet.

Servicefunktionen – wie Verteilung, Logistik und Kundenservice – sind in vielen Erzeugerländern jetzt führende Bereiche mit Wettbewerbsvorteil. Innovation und gestiegene Produktivität in der Serviceinfrastruktur (z.B. Kommunikation, Finanzwesen, Transport und Gesundheitsbereich) haben eine enorme Auswirkung auf die Produktivität und Leistung in allen anderen Segmenten der Wirtschaft. [Spoh08]

Viele Unternehmen entwickeln mittlerweile nicht nur Produkte oder Services, sondern kreieren auch Produkt-Service-Bündel. Es ist ein Transformationsprozess im Gange, bei dem eine steigende Verbindung zwischen Produkt- und Serviceindustrie aufgebaut wird.

Durch die Konzentration auf Kernkompetenzen können oder müssen weniger bedeutende Teile der Wertschöpfungskette eines Unternehmens an Kooperationspartner weitergegeben werden, wodurch Spezialisierungseffekte erzielt und Kosten eingespart werden können. Dies führte zu

dem anhaltenden Trend des Outsourcings, bei dem ehemals eigene Services an externe Unternehmen weitergegeben werden, was zu neuen Werteketten führt. Die Konzentration auf Kernkompetenzen bedingt gleichzeitig die Entwicklung von Kooperationen und virtuellen Netzen.

Früher wurde geglaubt, dass Wertschöpfung ausschließlich in den Herstellungsprozess eingebunden sei und dies führte zu der Annahme, dass Firmen Wert für Ihre Kunden erstellen. Während der letzten Jahrzehnte veränderte sich die Auffassung der Wertschöpfung und Kundenorientierung, und Dienstleistungen rückten in den Mittelpunkt. Von einigen Experten wird sogar behauptet, dass Firmen nur ein Nutzenversprechen machen können und der Nutzen und Wert nur vom Kunden bestimmt werden kann. Die Autoren Lusch und Vargo haben diesbezüglich mit ihrem Werk *Evolving to a New Dominant Logic for Marketing* [LuVa04] den Anstoß für eine breite Diskussion gegeben und gründeten damit die sogenannte "Service-Dominant-Logic", worauf innerhalb dieses Kapitels noch detailliert eingegangen wird.

Auch bei der Gestaltung elektronischer Produkte und Dienstleistungen geht es vorerst darum, mit Hilfe eines Geschäftsmodells eine geeignete Kooperationsform zu finden. Solche Kooperationsformen variieren von freien Marktplätzen mit verhandelbaren Waren und Werten über hierarchisch straff organisierte Netzwerke bis zu selbstorganisierten und lose gekoppelten Gemeinschaften. [MeSt05]

Zur Unterstützung des Geschäftsmodells werden Informations- und Kommunikationstechnologien verwendet, um Informationen über Kunden und Marktteilnehmer zu beschaffen und diese zu analysieren, was wieder dazu verwendet wird, neue Produkte und Dienstleistungen zu entwickeln. Außerdem helfen Informationssysteme den Unternehmen, die Entwicklung des Marktes und deren Teilnehmer untersuchen und interpretieren, um wieder strategische Anpassungen bei dem Entwurf und der Produktion von Gütern und Dienstleistungen auf die gegebene Situation machen zu können.

Elektronische Märkte ermöglichen dabei verschiedene, auch neuartige, Ertragsmodelle, die außerhalb des eBusiness (Electronic Business) nicht möglich gewesen wären.

Die Festlegung der Ertragsquellen, mit denen sich das Unternehmen refinanziert, bildet einen zentralen Bestandteil des Geschäftsmodells (Geschäftstätigkeit). Daraus lassen sich verschiedene Ertragsmodelle ableiten: z.B. Advertising Modell, Preismodell für Produkte & Dienstleistungen, Admission Modell (Eintrittsgebühr für zeitlich befristete Nutzung), Subscription Modell (fixe periodische Abonnementsgebühr), Gebühren für Transaktionen. [MeSt05]

Auf diese Themen wird in dieser Arbeit eingegangen, um zu zeigen, wie die aktuellen Erkenntnisse aus der Dienstleistungsforschung Einfluss auf die Entwicklung eines Service-Systems haben können.

2.2.1 Service Science

Service Science wird als ein neues wissenschaftliches Konzept definiert, welches die Lösung von komplexen Problemen einer Dienstleistungswirtschaft durch die Anwendung von transdisziplinären Ansätzen, und zwar durch intensive Zusammenarbeit zwischen Forschung und Serviceorganisationen zum Ziel hat. Services Science als eigenständige Disziplin beinhaltet Forschung sowie akademische Ausbildung für Studenten und Angestellte. Diesbezüglich sind für das Verstehen von Service Science drei Elemente besonders von Bedeutung: die

disziplinübergreifende Forschung, die intensive Zusammenarbeit zwischen Hochschulen und Unternehmen und die Ausbildung innerhalb akademischer Ausbildungseinrichtungen. [SEKL08]

Services Science wird als angewandte Wissenschaft betrachtet, welche zum Ziel hat, sich an der Lösungsfindung von sehr konkreten Serviceproblemen zu beteiligen. Dies wird durch enge Kooperation zwischen der wissenschaftlichen Welt und der Unternehmerwelt ermöglicht. Das bedingt ein starkes Networking zwischen der Forschung an Universitäten mit Forschung und Entwicklung innerhalb Unternehmen.

Dabei sind die Zusammenarbeit bei Forschungsprojekten, eine gemeinsame Definition und Auswahl von Forschungsproblemen sowie eine vereinte Auseinandersetzung mit Forschungsergebnissen notwendig. Nur dadurch kann ein durchgängiger Wissenstransfer zwischen Universitäten und Unternehmen erzielt werden, bei dem alle Teilnehmer davon profitieren. Universitäten und Wissenschaftler würden demnach von neuen Forschungsfragen und Erkenntnissen ebenfalls vielseitig Nutzen ziehen. Serviceunternehmen, die mit wissenschaftlichen Instituten kooperieren, haben die Möglichkeit fundierte Servicekonzepte mit ökonomisch attraktivem Marktpotenzial zu kreieren.

Um die Herausforderungen von Services bewältigen zu können, ist deren zielgerechte Verwaltung essentiell. Service Management kontrolliert, wie Geschäftsleistungen entwickelt und dem Kunden erbracht werden. Sie müssen flexibel ablaufen und zwischen Kunden und Anbieter verhandelbar sein. In den meisten Fällen sind Ressourcen beteiligt, wobei zeitliche Attribute eine große Rolle spielen. Deshalb ist eine Optimierung des Ressourceneinsatzes unbedingt notwendig, um eine effiziente Verwendung von Ressourcen sicherzustellen und um die Kundenzufriedenheit zu steigern. Dienstleistungsgeschäfte werden dabei oft aus verschiedenen Dienstleistungen der unterschiedlichen Unternehmen kumuliert angeboten. Um eine effiziente Konstruktion zu gewährleisten, sollten grundlegende Services und der Informationsgehalt so weit wie möglich standardisiert werden, ohne dabei die Flexibilität einzuschränken. [DoWe08]

In seiner Beschreibung über die Entwicklung der Serviceforschung charakterisiert Grönroos den Service-Management-Ansatz in fünf Basispunkten: [Gron94]

- Eine allumfassende Managementperspektive: Darunter ist das Abweichen vom traditionellen Managementfokus der Kostenreduktion, Wirtschaftswachstum und Arbeitsteilung gemeint, und es werden Marktpartnerschaft, die Effizienz des Unternehmens aus Kundensicht und angestellten-orientierte Anliegen betont.
- Kundenorientierung: Kundenzufriedenheit und Kundenloyalität werden als Ecksteine für ein erfolgreiches Servicemanagement genannt.
- Kundenbezogene Qualitätsorientierung: Qualität wird nicht mehr als Produktionsproblem behandelt, sondern als Problem der Kundensichtweise. Nicht das Unternehmen, sondern der Kunde definiert Qualität.
- Langzeit-Perspektive: Service Marketing und Management trachtet nach langlebigen und wertvollen Beziehungen – anstatt nach kurzfristigen Abschlüssen auf Basis von Kampagnen.
- Ganzheitlicher Ansatz des Managements: Die Barrieren zwischen den traditionellen Geschäftsfunktionen und akademischen Disziplinen müssen beseitigt werden, sodass die Notwendigkeit der Integration von Marketing, Unternehmensressourcen und Humankapital möglich ist.

Die Komplexität der Probleme, welche während der Produktion von Services auftauchen ist so hoch, dass es unzureichend wäre, an diese Probleme nur aus der Perspektive einer einzigen Forschungsdisziplin heranzutreten. Deshalb ist eine disziplins- und fakultätsübergreifende Untersuchung und Kooperation von Serviceproblemen notwendig. Die Integration der Perspektiven von Management und Marketing, als auch von IT-, Ingenieurs- und Sozialwissenschaften ist unbedingt notwendig. Es kann die Generierung von neuem servicespezifischem Wissen aufgrund der Disziplin Kreuzung, der Kombination von verschiedenen speziellen Kompetenzen, Modellen und Methoden erwartet werden. [SEKL08]

Im Gegensatz zu materiellen Produkten, ist der Konsument bei der Definition und beim Konsum von Dienstleistungsgeschäften beteiligt. Um leichter auf spezifische Kundenwünsche eingehen zu können, sind sowohl Service-Schnittstellen, als auch deren spezifischer Wert im Sinne einfacherer Adaptierung flexibler und allgemeiner definiert. Das bedeutet, dass die Verwaltung von Dienstleistungsgeschäften weitaus komplexer ist. [DoWe08]

Besonders Innovation ist für Serviceunternehmen ein zentraler Pfeiler einer nachhaltigen Unternehmensstrategie. Innovation ist mittel- und langfristig eine Grundvoraussetzung für den Unternehmenserfolg. Wissen kann dabei als einzige Quelle des andauernden Wettbewerbsvorteils gesehen werden. Erfolgreiche Innovationsprozesse sind nahezu unmöglich, wenn sich Unternehmen nur auf eigene Quellen im Rahmen der Innovationseingabe und Forschung verlassen. Dies bedingt ein ständiges Networking und den Aufbau verlässlicher Kooperationen und Partnerschaften. [Enge08]

Nicht nur ein neues Forschungsprogramm wird durch Services Science geformt, sondern auch ein Feld der Ausbildung an Institutionen, welche akademische Qualifikationen für eine Beschäftigung in Serviceorganisationen vorbereiten. Service Science bietet Fortbildungsmöglichkeit für Studenten sowie für Personen, die bereits im Sektors tätig sind. Dadurch wird der Weg für einen Beruf in der Serviceindustrie geebnet. Dies ermöglicht auch eine weiterführende Hochschulausbildung, welche die Vorbedingungen für einen Karriereaufstieg abdeckt. Die Entwicklung eines spezifischen Lehrplans, der die transdisziplinäre Natur von Services Science widerspiegelt ist dabei jedoch unumgänglich.

Für eine generelle Theorie von Services haben folgende zehn Bereiche besonders hohe Relevanz: Ökonomik und Rechtswissenschaften, Operations Research, Industrial Engineering, Informatik, Informationswissenschaft, MBA und Managementberatung, Wissensmanagement und Management-Informationssysteme, Organisationstheorie und Organisationales Lernen, Raumplanung sowie Ökosystem- und Naturdienstleistungen, Komplexitätstheorie und Komplexes Adaptives System für soziale Systemforschung. [Spoh08]

Ein Lehrplan sollte somit zumindest aus Modulen mit Inhalten aus den Bereichen Management, Soziologie, Psychologie, Rechtswissenschaft, IT und Ingenieurwissenschaft bestehen. sodass sich Studenten servicebezogenes Wissen aus unterschiedlichen spezialisierten Bereichen zusammen mit den Hauptqualifikationen und Soft Skills (für Serviceberufe besonders wichtig) aneignen. [SEKL08]

Nicht nur im Bereich des internationalen Handels und der Dienstgüterverträgen zwischen Unternehmen, sondern auch im Design von neuen Services, die als fair und nachhaltig betrachtet werden, hat die Verbindung zwischen Rechtswissenschaften und der Ökonomik weitreichende Folgen für Dienstleistungen. Dabei kommt es auch zur Anwendung von Methoden aus der Ökonomik auf rechtliche Probleme.

Operations Research ist eine interdisziplinäre Wissenschaft, die wissenschaftliche Methoden (wie z.B. mathematische Modellierung, Statistiken und Algorithmen) für die Entscheidungsfindung in komplexen echten Problemen anwendet, welche sich mit der Koordination und Ausführung von Operationen in einem Unternehmen befassen. Eine mögliche Verwendung dieser Wissenschaft ist auch der Versuch die bestmögliche Lösung für ein Problem wissenschaftlich zu eruieren, was zur Optimierung bzw. Leistungssteigerung der Organisation führen soll. Grundlegende Instrumente sind dabei Statistiken, Stochastiken, die Optimierung, die Simulation, die Warteschlangen-, die Spiel- und die Graphentheorie.

Methoden und Instrumente des Operations Research werden bei vielen praktischen Problemen mit Serviceabläufen wie Supply Chain Management, Verbesserung der Servicequalität im Netzwerkentwurf, effizientes Kundenbeziehungsmanagement, Personal Management eingesetzt. Operations Research wirft einen Gesamtblick auf ein ganzes System und versucht dieses zu verbessern, anstatt sich auf ein spezifisches Element zu konzentrieren.

Auch Industrial Engineering (IE) betrachtet Techniken aus dem Operations Research als einen bedeutenden Teil seines Werkzeugsatzes, bezieht dabei daraus aber eine eher technische Perspektive. IE ist eine Ingenieursdisziplin, die sich mit der Entwicklung, Verbesserung, Implementierung und Evaluierung von integrierten Systemen von Personen, Wissen, Ausstattung, Energie, Material und Prozessen beschäftigt.

IE hat vor allem zum Ziel die Produktivität zu steigern, die Produktqualität zu erhöhen, Produkte einfacher herstellbar zu machen und Prozesse effizienter zu gestalten. Das Anwendungsfeld des IE ist längst nicht mehr die Industrie alleine, sondern liegt überall dort, wo Leistungserstellungsprozesse vorliegen. Der Begriff wurde ursprünglich im Bereich der Herstellung und Produktion verwendet, hat sich jedoch erweitert und umfasst nun auch die Serviceindustrie u.a. IE hat bereits international an den Universitäten und Fachhochschulen den Einzug geschafft.

Die wachsenden Anforderungen an die Funktionalität von Informationstechnologien sind ohne Konzepte der Informatik nicht bewältigbar, somit müssen diese Konzepte in einer generellen Theorie von Services Science auch enthalten sein. Dienstleistungen auf Basis von Informationstechnologien können ohne die Überprüfung der technischen Realisierbarkeit nicht durchgeführt werden. Abgesehen von den Methoden der Softwareentwicklung bieten speziell Multiagentensysteme eine Basis für die Modellierung von Services. Des Weiteren behandelt Service Computing die technischen Leistungsfähigkeit und Standards im Bereich Web Services und serviceorientierten Architekturen, welche rapide wachsende Interessensgruppen unter IT-Experten entwickeln.

Die Informationswissenschaft ist ebenfalls ein interdisziplinäres Gebiet, welches sich primär mit der Sammlung, Speicherung, Wiederaufindung, Klassifizierung, Manipulation und Verteilung von Information beschäftigt. Die Informationswissenschaft studiert dabei die Anwendung und die Gewohnheit in Organisationen und die Interaktion zwischen Menschen, Organisationen und Informationssystemen. Dabei wird versucht, die Probleme aus Sicht der Beteiligten zu sehen. Ein großer Anteil des Sektors beschäftigt sich mit Informationsdienstleistungen – Tendenz steigend.

Master of Business Administration (MBA) ist ein postgraduales Managementstudium in der Geschäftsadministration, welches Leute aus den verschiedensten akademischen Bereichen anzieht. MBA-Programme beinhalten mannigfaltigste Teilbereiche wie Ökonomik, organisatorisches Verhalten, Marketing, Rechnungswesen, Finanzwesen, Strategie, Operations Management, internationales Business, IT-Management, Management von Innovationen,

Humankapitalmanagement, Risiko- und Versicherungsmanagement, Finanzsteuerung, E-Commerce, Grundsätze der Staatsverwaltung, Strategisches Consulting und Management Consulting.

Managementberatung versucht die Leistung von Unternehmen durch die Analyse bestehender Geschäftsprobleme und die Entwicklung von Zukunftsplänen zu verbessern und beinhaltet eine praktische Unterstützung von Unternehmen. Abgesehen von der generellen Beratung gibt es ebenfalls Spezialisierungsmöglichkeiten in einer Art des Consultings.

MBA-Programme und Managementberatungspraktiken sind somit ebenfalls für eine generelle Theorie von Notwendigkeit, da das professionelle Management von Innovation und Technologie speziell für die Dynamik von Unternehmen relevant ist.

Management-Informationssystem (MIS) ist eine akademische Disziplin, welche die Informationstechnologien auf Geschäftsprobleme anwendet und damit versucht, Führungskräfte bei der Entscheidungsfindung zu unterstützen und dem Controlling hilfreiche Informationen aufzubereiten, indem Unternehmensdaten in einem Datenmodell gesammelt und analysiert werden. Wissensmanagement bezieht sich auf die Art und Weise, wie Organisationen erlangtes Wissen sammeln, verwalten und verwenden. Diese Disziplinen konzentrieren sich vor allem auf die praktische Anwendung der Methoden. Das Tempo des technologischen Wandels und die neuen Methoden, die diese Disziplinen bereitstellen, können eine wichtige Quelle für Wettbewerbsvorteile und ausgezeichnete finanzielle Leistung von Unternehmen sein.

Organisationstheorie und Organisationales Lernen sind verwandte Bereiche der akademischen Studie von Organisationen. Methoden der Ökonomik, Soziologie, Politikwissenschaft, Anthropologie und Psychologie werden für die Untersuchung von Unternehmen angewendet. Verwandte praktische Disziplinen wie z.B. Human-Ressourcen sind ebenfalls einbezogen. Die Organisationstheorie untersucht die Dynamik von Individuen und Gruppen im Rahmen einer Organisation, als auch die Natur der Organisation selbst. Organisationales Lernen ist ein Wissensbereich innerhalb der Organisationstheorie und untersucht, wie Unternehmen lernen und sich umstellen. Auch diese akademischen Studienrichtungen sind für eine generelle Servicetheorie auf Unternehmensebene unerlässlich.

Die Raumplanung ist eine Disziplin, die sich unter anderem mit der physischen, sozialen und wirtschaftlichen Entwicklung von großstädtischen Räumen, Gemeinden und Wohngebieten beschäftigt. Ökosystem- und Naturdienstleistungen werden von der natürlichen Umwelt produziert und können vom Menschen konsumiert werden. Dazu gehört die Bereitstellung von reinem Wasser und sauberer Luft, die Bestäubung von Pflanzen, die Verringerung von umweltbedingten Gefahren sowie Krankheitskontrolle. Naturdienstleistung ist der Überbegriff für die Art und Weise wie die Natur dem Menschen nützt, vor allem der Nutzen, der auf ökonomische Weise betrachtet werden kann. [Spoh08]

Die Komplexitätstheorie ist ebenfalls eine interdisziplinäre Forschungsdisziplin, die versucht Antworten auf grundlegende Fragen von lebenden, anpassungsfähigen und veränderlichen Systemen zu finden. Komplexe adaptive Systeme (CAS) sind Sonderfälle von komplexen Systemen. Sie sind komplex weil sie aus mehreren zusammenhängenden Elementen bestehen und sie sind adaptiv, weil sie ein besonderes Anpassungsvermögen an ihre Umwelt zeigen und die Möglichkeit haben, (aus Erfahrung) zu lernen. Um für eine allgemeine Service-Science-Theorie relevant zu sein, müssen sich CAS auf soziale Systeme fokussieren. [Spoh08]

Einerseits gibt es einen breiten Konsens betreffend der Notwendigkeit einer eigenständigen Services Science Forschung und Ausbildung. Andererseits gibt es verschiedene Gegenströme, welche die Etablierung dieses neuen Feldes erschweren. Beispielsweise stehen individuelle Disziplinen eher traditionell entgegengesetzt zu einer disziplinübergreifenden Forschung und Ausbildung. Gründe dafür können sein, dass transdisziplinäre Forschung geringere Chance bei Publikationen haben könnte oder Forscher im interdisziplinären Bereich schmalere Karrierechance sehen. Auch bei der Kooperation zwischen den Hochschulen und Unternehmen gibt es Hindernisse. Manche Faktoren, wie die mangelnde Serviceforschungs-Tradition und die Unterschätzung deren Relevanz, spielen hierbei eine Rolle. Zusätzlich fürchten Forscher teilweise, dass die Unternehmen zu viel Einfluss auf die Definition der Forschungsfragen und zu viel Kontrolle über den Forschungsprozess haben könnten. Auf der Seite der Unternehmen dämpfen Faktoren, wie Zeitdruck, finanzielle Ziele und mangelnde Ressourcen, oft eine fortlaufende Durchführung von Forschungsprojekten. [SEKL08]

Servicebezogene Forschung ist weitverbreitet, wird jedoch nicht als Services Science wahrgenommen. Eine starke Forderung nach der Etablierung von Services Science scheint das Resultat des wahrgenommenen Forschungsmangels zu sein. Das Fundament für den öffentlichen Ruf nach einer neuen Servicedisziplin ist die Annahme, dass sich die jetzige Forschung vornehmlich mit den Problemen von Industriebetrieben beschäftigt und die spezifischen Probleme von Services vernachlässigt. Diese Annahme wird durch Veröffentlichungen der serviceorientierten Programme an Universitäten und Business-Schulen bestätigt. Auch die wissenschaftlichen Journale aus diesem Gebiet stimmen mit dieser Meinung überein. [Stau08]

Da Serviceforschung nicht nur von Serviceforschern durchgeführt wird, berücksichtigt dieser Standpunkt nur die wissenschaftliche Realität, was zu einer verzerrten Perspektive führen könnte.

Da die weitreichende Mehrheit der Unternehmen heutzutage Servicebetriebe sind und auch Produktionsbetriebe im Servicebereich wachsen, hat sich die allgemeine Serviceforschung in Unternehmen weiterentwickelt. Zusätzlich gibt es auch einige wissenschaftliche Journale, die speziell auf die Serviceindustrie abzielen. Demnach scheint es keinen Mangel an Serviceforschung zu geben. Es existiert auch eine breite Masse an servicebezogener Forschung, die jedoch nicht als Serviceforschung erkennbar ist. Es gibt somit keinen zwingenden Mangel an Forschungsarbeit im Servicebereich, jedoch fehlt womöglich ein allgemeines Verständnis von Serviceforschung, also ein weitläufig anerkanntes Services Science Paradigma. [Stau08]

Die Frage ist jedoch, wie diese Forschung charakterisiert werden kann und welche Themen von der Servicegemeinschaft als relevant bezeichnet werden. Bernd Stauss und seine Mitarbeiter haben eine Schlagwort-Analyse innerhalb der zwei relevantesten servicebezogenen Journalen durchgeführt (*Journal of Service Research* und *The International Journal of Service Industry Management*). Wie auch schon andere zuvor kommen sie in ihrer Auswertung zu einem ähnlichen Ergebnis. Ein Großteil der Beiträge handelt von Aspekten der Kundenzufriedenheit, Servicequalität und Kundenloyalität. Das Ergebnis zeigt klar, dass es einen klaren Fokus auf eine limitierte Anzahl von Themen gibt. [Stau08]

Es gibt Multidisziplinarität (Probleme betreffen mehrere Disziplinen) in der jetzigen Serviceforschung, jedoch ist die interdisziplinäre (Problemlösung durch Beteiligung mehrerer Disziplinen) Forschung selten.

Eine Analyse von Jos Lemmink [Lemm05] zeigt die Beteiligung im *International Journal of Service Industry Management* verschiedener Autoren und zu welcher Abteilung diese gehören. Die

überwiegende Mehrheit war vom Bereich der Marketing- und Businessforschung, gefolgt von der Management- und Konsumentenforschung. Fünfundzwanzig Prozent der Beträge waren Resultate von disziplinübergreifenden Kollaborationen. Eine ähnliche Studie von Bernd Stauss und seinen Mitarbeitern zeigt, dass funktionsübergreifende Kooperationen selten sind. Einige Disziplinen beteiligen sich an der gegenwärtigen Serviceforschung, jedoch dominiert Marketing und Management, die interdisziplinäre Kooperation ist allerdings eher die Ausnahme, als die Regel. [Stau08]

Services Science sollte seine Relevanz und seinen konzeptionellen Kern klarer und präziser abgrenzen. Momentan ist es noch immer ziemlich unklar, wofür Services Science steht. In der Services Science Literatur handelt die meiste Information über die Art der Wissenschaft, die gefordert wird. [Stau08]

Zusammenfassend wird Services Science oft wie folgt beschrieben: [Ches05][ChSp06][Stau08]

- Lösungsorientiert, indem man nicht auf eine Problembeschreibung und Problemerkklärung begrenzt bleibt, jedoch die Entwicklung von Lösungsvorschlägen und tatsächliche Lösungen beabsichtigt bleiben.
- Problemorientiert, indem konkrete Probleme als Ausgangspunkt genommen werden.
- Transdisziplinarität meint, dass Services Science mit komplexen Problemen umgehen muss, welche die Kollaboration von Experten aus verschiedenen Disziplinen erfordert.
- Kollaborativ im Sinne von intensiver Kooperation zwischen Managern von Unternehmen, Forschern und Universitäten, um permanenten gegenseitigen Wissenstransfer zu ermöglichen.
- Team-orientiert, weil die beabsichtigten Effekte nur erreicht werden können, wenn Forscher und Praktiker aus verschiedenen Disziplinen ihre unterschiedlichen Perspektiven durch Teamarbeit integrieren.
- Methodisch facettenreich, sodass ein neu aufkommender Wissenschaftsansatz nicht auf ein einziges methodischen Paradigma verweisen kann, sondern stattdessen den Wettbewerb in verschiedenen Methodischen Traditionen und Leitbildern anregen sollte.

Daraus ergibt sich ein klares Verständnis für den wissenschaftlichen Aspekt, bei dem eine Wissenschaft mit einem expliziten Fokus auf die Lösung von Geschäftsproblemen gefordert und gefördert wird.

2.2.2 Service Dominant Logic

Die sogenannte Service-Dominant-Logic (SDL) repräsentiert das Abweichen von der traditionellen güterdominanten Logik (GDL) des Handels (Marktverhaltens), bei der Güter der Fokus des Handels sind und Dienstleistungen eine spezielle Art von Gütern darstellen. Die neue Logik repräsentiert einen Übergang von der Betonung von "operanden" Ressourcen (normalerweise greifbare und inaktive Ressourcen) zu einer Betonung von "operanten" Ressourcen (dynamische Ressourcen, die mit anderen Ressourcen agieren). [LuVa04]

Als Folge verwirft die SDL die traditionelle Klassifizierung von Gütern und Services. Service sind nicht das Gegenstück zu Gütern. Güter sind Vorrichtungen (Werkzeuge, Verteilungsmechanismen) die als Alternative zur direkten Leistungserbringung dienen. In diesem Sinne repräsentiert die SDL ein invertiertes Bild der GDL. Güter sind eine Spezialfall, oder zumindest eine spezielle Methode, der Leistungserbringung. Somit stellt Service den allgemeinen Fall eines Handelsprozesses dar. Es ist immer Service, was gehandelt und getauscht wird. [LuVa04]

Die SDL sieht angewandte, spezialisierte Fähigkeiten und Wissen als das Hauptaugenmerk von wirtschaftlichem Handeln und einen der fundamentalen Grundlagen, auf welchen die Gesellschaft aufgebaut ist. Demnach ruht die SDL auf der Annahme, dass Individuen und Kollektive durch die Anwendung von spezialisierten Fähigkeiten und Wissen mit Services handeln, um ihr eigenes Wohl zu erweitern, indem sie ihre eigenen Services zu Verfügung stellen, um als Gegenleistung die Services von anderen zu erlangen, die sie ebenfalls benötigen. Wenn Güter in den Handel involviert sind, werden diese als Mechanismen für die Servicebereitstellung gesehen.

Eine zentrale Rolle im Wertschöpfungsprozess ist in der SDL dem Konsumenten zuordnet, da der Service-Wert als eine gemeinsam ausgeführte Funktion durch die Aktionen des(r) Anbieter und dem(r) Konsumenten entsteht. Der Wert selbst wird jedoch immer vom Konsumenten bestimmt, wodurch diese Ansicht entscheidende Auswirkungen auf die Art und Weise hat, wie Unternehmen in den Markt treten. Allerdings sieht die SDL nicht nur der Konsument am Co-Wertschöpfungsprozess beteiligt, denn auch die Partner der Organisation gehören durchaus zum Wertschöpfungs-Netzwerk.

Unter dem Gesichtspunkt der SDL gibt es somit keine sektorale Unterteilung, da alle Tätigkeiten Services sind, Industrie und Landwirtschaft sind nur spezielle Formen davon.

In der Gesellschaft gibt es eine Beschleunigung in der Arbeitsteilung, was dazu führt, dass Personen immer mikrospezialisierter werden. Durch das steigende fachkundige Wissen auf kleineren Gebieten sinken ihre Optionen am Markt. Wenn jemand hochspezialisiert ist (im Gegensatz zu einem Generalisten), gibt es oft keinen effizienten Markt, in dem diese Person ihre speziellen Kompetenzen direkt handeln kann. Im Kontrast sieht SDL die Mikrospezialisierung von Einzelpersonen als Eingabewert, den der Geschäftsführer kombinieren kann, um ein (komplexes) Service zu erstellen, welches von den Leuten gefordert wird (z.B. Computersysteme, Transport, Ausbildung). Organisationen integrieren demnach individuelle Kompetenzen von Personen, um diese in Services transformieren zu können, welche von Menschen am Markt gefordert oder gebraucht werden. [LuVa04]

Vargo und Lusch möchten die SDL auf neun fundamentalen Annahmen ("foundational premises" - FP), konstruieren, die sich wie folgt zusammenfassend charakterisieren lassen: [LuVa04]

- FP1: Die Anwendung von spezialisierten Fähigkeiten und Wissen ist die grundlegende Einheit im Handel.
- FP2: Es findet immer ein Tausch von Service gegen Service statt, auch wenn dies nicht gleich erkenntlich ist beziehungsweise durch verstärkte Arbeitsteilung verdeckt ist.
- FP3: Güter sind Distributionsmechanismen für die Bereitstellung und Erbringung von Services.
- FP4: Wissen ist die fundamentale Quelle von Wettbewerbsvorteilen.
- FP5: Alle Ökonomien sind Service-Ökonomien.
- FP6: Der Kunde ist immer Koproduzent/Koersteller des Leistungserstellungs- bzw. des Wertentstehungsprozesses.
- FP7: Unternehmen können nur Wertvorschläge unterbreiten, jedoch Gütern keinen Wert einbauen.
- FP8: Eine servicezentrierte Sicht ist kundenorientiert und relational.
- FP9: Unternehmen existieren, um mikrospezialisierte Kompetenzen und Fähigkeiten in komplexe Services zu integrieren und transformieren, die am Markt gefordert werden.

Im Gegensatz zu den relativ schwachen normativen Richtlinien der GDL liefert die SDL folgende Richtlinien in Bezug auf Unternehmen: [LuVa04]

- Unternehmen sollten transparent sein und all ihre Informationen symmetrisch in den Handelsprozess eingliedern.
- Unternehmen sollten danach streben Kundenbeziehungen aufzubauen und Langzeitperspektiven einzunehmen. Demnach sollten Firmen soweit wie möglich im Interesse des Kunden handeln, um ein Langzeit-Wohlbefinden des Kunden zu erzielen.
- Unternehmen sollten Investitionen in die Entwicklung von Fachkenntnissen und Fachwissen unterstützen und tätigen, da dies die Urquelle von Wirtschaftswachstum darstellt.

Ein Zusammenfassender Überblick der Unterschiede zwischen der GDL und der SDL:

Goods Dominant Logic	Service Dominant Logic
Güter	Services
Greifbar	Ungreifbar
Operande Ressourcen	Operante Ressourcen
Asymmetrische Information	Symmetrische Information
Propaganda	Konversation
Wertbehaftet	Wertvorschlag
Transaktional	Relational
Profitmaximierung	Finanzieller Feedback

Tabelle 1: Gegenüberstellung - SDL und GDL

Die SDL liefert demnach ein Basisfundament für die Entwicklung einer allgemeinen Theorie. Die SDL wird von Lusch und Vargo als "open Source" betrachtet, wodurch sich die Autoren eine Beteiligung anderer erhoffen, um einen breiten Konsens für dieses neue Paradigma zu ermöglichen.

2.2.3 Service Engineering

Auch die zentrale Forderung des Service Engineering ist die systematische Entwicklung von Dienstleistungen. Dabei beschäftigt sich die Forschungsdisziplin seit Mitte der 90er Jahre mit der zielgerichteten Entwicklung von Dienstleistungen durch Verwendung geeigneter Methoden, Vorgehensweisen und Werkzeuge. Unter Berücksichtigung dienstleistungsspezifischer Eigenschaften wird in diesem Zusammenhang angenommen, dass Dienstleistungen annähernd wie Sachgüter und Software entwickelt werden können. [ScGK06][BuSc06]

Die Disziplin versorgt Unternehmen dabei mit Konzepten zur systematischen Entwicklung von Dienstleistungen, um diese methodisch und zielgerecht entwerfen zu können. Dienstleistungsentwürfe sollen dabei effizient in qualitativ hochwertige marktfähige Leistungen transformiert werden. [BuSc06]

Die Nutzung betriebswirtschaftlicher Konzepte, standardisierter Vorgehensmodelle und einer Konstruktionsmethodik sollen zu einer entscheidenden Verbesserung der Dienstleistungsentwicklung führen. Die Verminderung der Entwicklungskosten, eine reduzierte Entwicklungszeit bis zur Markteinführung und eine stetige Erweiterung der Dienstleistungsqualität helfen als positive Konsequenz die Marktposition des Unternehmens abzusichern und zusätzliche Ertragspotenziale zu generieren. Ebenfalls ist es üblich, Verfahren der Produktmodellierung auf Dienstleistungen anzuwenden. [BuSc06]

Um auf unerwartete Ereignisse im Prozess entsprechend zu reagieren, müssen die jeweiligen Arbeitsschritte überwacht werden. Dies erfordert die Etablierung eines flexiblen Controllings, um zielgerechte Anpassungen bei Problemen durchführen zu können. Bereits in der Entwicklungsphase empfiehlt sich eine kundenorientierte Zusammenarbeit verschiedener Abteilungen. Durch eine frühe Integration von Marketing, Vertrieb, Personalmanagement, IT und Qualitätskontrolle können mögliche Schwachstellen schon vorab ausgemacht werden.

Diese interdisziplinäre Zusammenarbeit erfordert ein gemeinsames Verständnis der zu bewältigenden Herausforderung und eine funktionsgrenzen-überschreitende Kommunikation. [BuSc06]

Gerade der Ausbildung des am Dienstleistungsprozess beteiligten Personals muss ein großer Stellenwert zugerechnet werden. Die Entwicklung wesentlicher Sozial- und Methodenkompetenz, und die Motivation der Mitarbeiter sind absolute Notwendigkeiten, um das Personal für die durchzuführenden Aktivitäten zu qualifizieren. [BuSc06]

Um die Dienstleistungsentwicklung durch Software-Systeme effizient unterstützen zu können, müssen diese flexibel (skalier-, anpass- und erweiterbar) soweit wie möglich entworfen sein, da die Gestaltung der IT-Systeme die Effizienz der Dienstleistungserbringung signifikant beeinflusst. [BuSc06]

Im Gegensatz zur SDL unterscheidet Service Engineering nach wie vor zwischen Produkten und Dienstleistungen.

Scheer, Grieble und Klein definieren dies wie folgt: "Der Leistungs- bzw. Produktbegriff wird in Sach- und Dienstleistungen aufgeteilt. Letztere werden wiederum in Informations- und sonstige Dienstleistungen unterschieden. Eine Leistung kann generell als das Ergebnis eines (Geschäfts-) Prozesses bezeichnet werden. Ein Produkt ist eine Leistung oder eine Gruppe von Leistungen, die von Stellen außerhalb des jeweils betrachteten Fachbereichs (innerhalb oder außerhalb der

Organisation) benötigt werden. Der Begriff Leistung wird ... dem Begriff Produkt gleichgesetzt, wenn dies für Dienstleistungen auch noch etwas ungewöhnlich klingt." [ScGK06]

Sachgüter sind im ursprünglichen Sinn greifbare Waren oder materielle Produkte, die vom Kunden vor dem Kauf betrachtet und berührt werden können. Der Konsument ist im eigentlichen Produktionsprozess nicht eingebunden. Jedoch sind im Produktionsprozess der Sachgüter meistens auch einige immaterielle Objekte wie Arbeitsleistungen, Dienstleistungen, Informationen und Rechte verschiedenster Art involviert. [ScGK06]

Die vorhin bereits erwähnten Autoren Scheer, Grieble und Klein gliedern Dienstleistungen im Artikel *Modellbasiertes Dienstleistungsmanagement* auf Basis einer Literaturrecherche⁴ in vier verschiedene Definitionsansätze, wobei sie eine Abgrenzung in eine enumerative, eine negative, eine institutionelle und eine konstitutive Kategorie durchführen, und zwar folgendermaßen: [ScGK06]

- Enumerative Definitionen:
Durch die Auflistung von Beispielen wird versucht das Wesen von Dienstleistungen näher zu bestimmen.
- Definitionen durch Negativabgrenzung:
Alles was nicht einer Sachleistung zugeordnet werden kann, wird bei dieser Definition als Dienstleistung bezeichnet.
- Definitionen durch institutionelle Abgrenzung:
In diesem Fall handelt es sich um eine sektorale Unterteilung bei der Dienstleistungen ausschließlich im tertiären Sektor einer Volkswirtschaft produziert werden. Unter anderem werden Handel, Verkehr, Versicherungen, Banken und Nachrichtenwesen diesem Sektor zugerechnet.
- Definition durch konstitutive Abgrenzung:
Der Dienstleistungsbegriff wird in diesem Fall aufgrund bestimmter Eigenschaften und auf Basis von konstitutiven Merkmalen spezifiziert. Oft genannte Merkmale von Dienstleistungen sind die Immaterialität und die Integration des externen Faktors in den Leistungserstellungsprozess.

⁴ Literaturrecherche auf Basis von [Cors01],[MeBr00], et.al.

Abgesehen von der Unterscheidung von Dienstleistungseigenschaften, gibt es beim Dienstleistungsbegriff auch eine Differenzierung nach Phasen beziehungsweise Dimensionen. In den meisten Definitionsansätzen wird dabei zwischen der potenzial-, prozess- und ergebnisorientierten Dimension unterschieden. [ScGK06]

- Die potenzialorientierte Dimension:
Die Fähigkeit und Bereitschaft zur Bereitstellung einer Leistung wird in dieser Dimension behandelt. Durch die Kombination interner Potenzial- und Verbrauchsfaktoren (Ressourcen, wie Maschinen, Humanressourcen, Informations- und Kommunikationssysteme), die ein Anbieter zur Verfügung hat, wird die anschließende Erstellung der Leistung ermöglicht. Aufgrund der eingeschränkten Speicher- und Lagermöglichkeit sowie der zeitlichen Abweichungen der Nachfrage an Dienstleistungen entstehen spezielle Anforderungen an die Gestaltung der Potenzialdimension und bedingt eine Kapazitätsverwaltung. [ScGK06][Bull06]
- Die prozessorientierte Dimension:
Die Potenzialfaktoren des Anbieters und des Nachfragers werden in den Prozess der Dienstleistungserbringung eingeführt, wodurch es zur Übertragung der Potenzialdimension auf externe Faktoren (beispielsweise Kunden) kommt. Dienstleistungen werden als Prozesse betrachtet, bei dem es unter einer gewissen Anordnung von Handlungen zur Erstellung eines Produkts kommt, wobei die Integration des Kunden in den Prozess meistens zwingend notwendig ist, um die Dienstleistung überhaupt ausführen zu können. Auch hier bekommt die Simultanität von Dienstleistungen (gleichzeitige Erstellung und Verbrauch) ein Hauptaugenmerk. [ScGK06][Bull06]
- Die ergebnisorientierte Dimension:
Es wird der Zustand nach Vollendung des Dienstleistungsprozesses beschrieben. Dabei muss jedoch bezüglich der Qualität der Ergebnisdimension zwischen dem prozessualen Endergebnis und der eigentlichen Wirkung der Dienstleistung unterschieden werden. Die Auswertung des prozessualen Endergebnisses fällt zeitlich mit dem Abschluss des Produktionsprozesses zusammen, wobei die Wirkung (ursprüngliches Ziel der Dienstleistung und deren Folgen) erst mittel- bis langfristig bewertet wird und die Eigenschaft einer Folge- oder Dauerqualität aufweist. [ScGK06]

Zu betonen ist auch die Feststellung von Bullinger: "Interaktionen zwischen der dienstleistungsanbietenden und der -nachfragenden Wirtschaftseinheit finden sowohl auf der Potenzial-, der Prozess- als auch auf der Ergebnisebene statt." [Bull06]

Im Mittelpunkt des Service Engineering stehen besonders die Verwendung geeigneter Vorgehensmodelle, Methoden und Werkzeuge zur Entwicklung und Gestaltung von Dienstleistungen. [Bull06]

Vorgehensmodelle begleiten schrittweise den Dienstleistungsentwicklungsprozess von der Entwicklung des Entwurfs bis zur Einführung der marktreifen Dienstleistung. Ein Vorgehensmodell kann dabei in verschiedene Phasen unterteilt werden: [Bull06]

Phase	Aktion
Startphase	Formierung von Ideen
Analysephase	Anforderungen analysieren und Ideen bewerten
Konzeptionsphase	Einzel- und Gesamtspezifikationen erstellen
Vorbereitungsphase	Potenzial bereitstellen
Testphase	Gesamtspezifikation testen
Implementierungsphase	Konzept implementieren

Tabelle 2: Vorgehensmodell - Dienstleistungsentwicklungsprozess

Verschiedene Methoden werden wiederum verwendet, um die einzelnen Phasen des Vorgehensmodells zielgerichtet und effizient behandeln zu können. Sie beinhalten für den jeweiligen Anwendungsbereich eine spezifische Vorgehensweise, die bestimmt, welche Tätigkeiten für das Erreichen eines gewünschten Ergebnisses durchzuführen sind. Im Rahmen des Service Engineering eignet sich je nach Phase eine Vielzahl von Methoden: [Bull06]

Phase	Methodenbeispiel
Analysephase	schriftliche und mündliche Kundenbefragungen oder Feedback-Auswertungen zur Ermittlung von Kundenanforderungen
Konzeptionsphase	Kapazitätsplanung zur Konzeption des Ressourcenbedarfs, Modellierungstechniken, Morphologie

Tabelle 3: Vorgehensmodell - Methodenbeispiele

Um den Dienstleistungswicklungsprozess funktional zu unterstützen, können Werkzeuge in Form von Informations- und Kommunikationssystemen verwendet werden. Ein gängiger Anwendungsfall ist der Einsatz von Werkzeugen zur Planung von Geschäftsprozessen, zur Analyse und Simulation von Prozessmodellen, zur Verwaltung von Kundenbeziehungen (CRM) und zur Unterstützung des Projektmanagements. [Bull06]

Um Dienstleistungen langfristig am Markt bereitstellen zu können, ist eine regelmäßige Innovation dieser notwendig. Innovation heißt wörtlich "Erneuerung", womit bei der Anwendung auf Leistungen eine Neuheit, Neuartigkeit oder Erweiterung verbunden wird. Aufgrund diverser Besonderheiten von Dienstleistungen muss auch die Dienstleistungsinnovation speziell behandelt werden.

Reckenfelderbäumer unterscheidet hinsichtlich der Innovationsart die Produkt-, Prozess- (bzw. Verfahrens-) und Potenzialinnovationen. Produktinnovationen sind nach außen auf den Markt gerichtet. Die Prozess- und Potenzialinnovationen versuchen erstrangig eine interne Produktivitätssteigerung. Als Beispiele für Potenzialinnovationen können, neben den Neuerungen im Bereich der Bereitstellungsleistung des Anbieters, auch Sozial- und Strukturinnovationen, die Personalveränderung und Änderungen im Ordnungs- und Beziehungsgefüge der Unternehmung, genannt werden. [Reck06]

Beim Innovationsgrad kann zwischen Basisinnovationen (hoher Innovationsgehalt) und Folge- und Verbesserungsinnovationen (Modifikationen und Variationen bestehender Produkte) unterschieden werden. Innovationen können einerseits als Folgeerscheinung von unternehmerischer Forschung entstehen und andererseits auch vom Markt gefordert werden, weshalb Unternehmen versuchen sollten ihre Innovationsquellen entsprechend auszurichten. [Reck06]

Gewerbliche Schutzrechte, wie z.B. Patente können aufgrund des hohen Immaterialitätsgrads von Dienstleistungen oft nicht auf diesbezügliche Innovationen angewendet werden. Aus diesem Grund müssen Anbieter versuchen anders Wettbewerbsvorteile aufzubauen. Der Hauptgrund für Innovationen ist meist der Aufbau beziehungsweise die Sicherstellung von Wettbewerbsvorteilen gegenüber der Konkurrenz. Ein Wettbewerbsvorteil ist ein komparativer Unterschied, der aus Sicht des Kunden sowohl wahrnehmbar als auch relevant ist. Dabei ist es möglich, dass Wettbewerbsvorteile erst durch die Kundeneinbindung im Innovationsprozess identifiziert werden. [Reck06]

Zusammenfassend liefert Service Engineering somit einen detaillierten Leitfaden zur Entwicklung, Verwaltung und Verbesserung von Dienstleistungen und bietet somit diverse Einsichten, die hinsichtlich des Service-Systems wertvoll sein können.

2.3 Servicetechnologien

Das folgende Kapitel geht vor allem auf aktuelle technologische Entwicklungen ein, die bezüglich des in dieser Diplomarbeit vorgestellten Systems besonders relevant sind. Dabei wird besonders auf die Themen *serviceorientierte Architekturen* (kurz SOA) und *Web Services* eingegangen, die sich zu Schlagwörtern der heutigen IT-Welt entwickelt haben. Diesbezüglich werden vorhandene Spezifikationen vorgestellt, um zu zeigen, worauf es bei der Verwendung von Web Services besonders zu achten ist.

Gerade diese Themen sind zentrale Ankerpunkte dieser Arbeit, da das geplante System dem Entwurfparadigma einer serviceorientierten Architektur entsprechen soll, wobei der Prototyp schlussendlich mittels Web Services umgesetzt wird.

Auch die Themen *Ontologien* beziehungsweise *Semantic Web Services* werden angeschnitten, da sie große Relevanz im Bezug auf automatisiertes Auffinden und dynamisches Binden von Web Services haben. Primär geht es dabei darum, wie Services in maschinenlesbarer Form beschrieben werden können.

Jedoch haben die neuen Konzepte rund um Service-Orientierung nicht nur eine technische Komponente, sondern auch deutliche Konsequenzen für die Organisationen, die auf diese Entwicklung setzen. [Melz08]

Besonders für die IT-Schicht stellt der Versuch, Dienstleistungen halb- oder vollautomatisiert abzubilden, eine bedeutende Herausforderung dar. Um dem entgegenzuwirken, können komplexe Dienste in kleinere Dienste zerteilt werden, um diese leichter beschreib- und evaluierbar zu machen. Außerdem steigert das die Chancen einer Wiederverwendung und ermöglicht, Dienste verteilter durchzuführen.

Service-orientierte Architekturen versprechen eine flexiblere Verwendung physischer Ressourcen sowie des Internets, weshalb sie auch Dienstleistungsgeschäfte auf eine flexiblere Art unterstützen könnten. [Diet06]

Durch die ständig wachsenden Anforderungen an Softwaresysteme steigt auch jeweils die Komplexität der Softwareentwicklung. Bis jetzt konnte man beobachten, dass jedes Mal, wenn die Komplexität nicht mehr beherrschbar war, wurde das Abstraktionsniveau gesteigert⁵. Service-orientierte Architekturen sind dabei das vorläufig letzte Glied in dieser Kette, da sie oft schon als der Nachfolger der objektorientierten Programmierung bezeichnet werden. SOAs erlauben dadurch abstraktere Sichtweise auf sonst sehr komplexe Systeme, wodurch es in vielen Fällen ermöglicht wird, die gestiegene Komplexität zu bewältigen und flexible, robuste, sichere und wiederverwendbare Architekturen zu entwickeln. Außerdem wird die Implementierung einer partnerübergreifenden Integration von IT-Systemen dadurch erleichtert. [Melz08]

⁵ Von Assemblersprachen über prozedurale Sprachen bis hin zur objektorientierten Programmierung.

Um die steigenden Markt- und Kundenanforderungen erfüllen zu können, ist der Einsatz von immer mehr innovativer IT notwendig. Auch die Autoren Lusch, Vargo und Tanniru identifizieren bezüglich Service im Artikel *Value Networks and Learning* sieben grundlegende Gründe, warum IT-Wachstum die Erweiterung von Service-Versorgungsnetzwerken ermöglicht, und zwar: [LuVT09]

- Durch IT-Wachstum konnten immer mehr Waren mit Mikroprozessoren und Intelligenz ausgestattet werden und wurden somit verbesserte Plattformen für die Servicebereitstellung.
- Aufgrund des IT-Wachstums stiegen die Möglichkeiten der Eigenversorgung.
- Somit wuchsen auch die Möglichkeiten zur Unterstützung von anderen.
- Durch die Erweiterung der Kommunikationsmöglichkeiten verringerte sich der Bedarf an Transport.
- Die vermehrten Kommunikationsmöglichkeiten erlaubten somit auch mehr Konsumenten und Lieferanten kennenzulernen, und
- es wurde dadurch möglich, mit Konsumenten und Lieferanten direkt zu interagieren.
- Durch die Möglichkeit günstiger zu kommunizieren, wurde die Koordination zwischen Firmen effizienter und reaktionsfähiger.

Der steigende Einsatz von IT ist jedoch natürlich mit erheblichen Kosten verbunden. Eine strategische Planung bei der Softwareentwicklung ist deshalb angebracht, um diese kosteneffizient durchführen und den laufenden Betrieb sparsam gestalten zu können. Wichtige Faktoren dabei sind, unter anderem, die leichte Anpassungsfähigkeit und die Wiederverwendbarkeit von Softwareartefakten.

2.3.1 Service-orientierte Architekturen

Die grundlegende Voraussetzung und Folge einer Service-orientierten Architektur⁶ ist, dass Applikationen durch Komposition einzelner Services erstellt werden.

Es gibt auch für SOAs eine Fülle unterschiedlicher Definitionen; derzeit existiert jedoch noch keine einheitliche Definition einer SOA. Die Autoren Leyking, Dreifus und Loos haben zum Thema SOA in ihrem Werk *Service-orientierte Architekturen* eine vergleichende Literaturstudie durchgeführt und untersuchen dabei die renommiertesten grundlegenden Werke aus diesem Bereich.

Thomas Erl – einer der aktivsten Autoren zum Thema SOA – definiert die Architektur wie folgt: "SOA is a form of technology architecture that adheres to the principles of service orientation. When realized through the Web services technology platform, SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise". [Erl05]

Ingo Melzer definiert Service-orientierte Architekturen als: "... eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht. ... und sind das abstrakte Konzept einer Software-Architektur, in deren Zentrum das Anbieten, Suchen und Nutzen von Diensten über ein Netzwerk steht. Dienste werden dabei fast ausschließlich von Applikationen oder anderen Diensten genutzt." [Melz08]

In Dieter Masaks Definition ist eine SOA: "... das Modell eines Systems, welches vollständig aus autonomen Services aufgebaut ist, deren Interaktion über dasselbe öffentliche Protokoll abläuft und im Modell stets die drei Rollen Provider, Consumer und Broker vorhanden sind." [Masa07]

Das W3C⁷ wiederum definiert SOA in der aktuellen Version ihrer *Web Services Architecture* [W3C04] als: "A Service Oriented Architecture is a form of distributed systems architecture that is typically characterized by the following properties: Logical view, Message orientation, Description orientation, Granularity, Network orientation, Platform neutral."

Services (bzw. Dienste) stehen in einer SOA somit klar im Mittelpunkt. Ein Service ist in diesem Zusammenhang ein Programm oder eine Softwarekomponente, das lokal oder über ein Netzwerk genutzt werden kann. Um dies zu ermöglichen, müssen Services über eine Schnittstelle zugänglich gemacht werden, wobei nur Zugriffe über diese Schnittstelle zulässig sind. Die Schnittstelle wiederum muss innerhalb einer sog. Service Description (in für andere Services lesbarer Form) beschrieben werden, um möglichen Nutzern zu erklären, wie diese zu verwenden ist. Um Plattformunabhängigkeit und somit eine breite Akzeptanz zu erreichen sollte die Beschreibung unabhängig von der Implementierung lesbar sein. Es konnten sich bereits verschiedene Beschreibungssprachen zu de facto Standards entwickeln (z.B. IDL⁸, WSDL⁹).

⁶ Englisch: Service-oriented Architectures (kurz SOA)

⁷ World Wide Web Consortium: Gremium zur Standardisierung der das World Wide Web betreffenden Techniken.

⁸ Interface Description Language

⁹ Web Services Description Language

Insbesondere im unternehmerischen Umfeld werden jedoch serviceorientierte Prinzipien schon längst nicht mehr auf technische Dienste beschränkt, sondern auf die abstraktere Ebene geschäftsbezogener Services gehoben. [LeDL07]

Bei der Kommunikation zwischen Applikationen handelt es sich jedoch nicht um eine neue Form der RPCs¹⁰, sondern vielmehr um die Möglichkeit, benötigte Funktionalitäten oder Dienste dynamisch zur Laufzeit einzubinden und aufzurufen. Bei einer SOA handelt es sich im Gegensatz zu Techniken wie RPC und RMI¹¹ nicht um eine konkrete Technik, sondern um eine Abstraktion. Die derzeit viel versprechendste Umsetzung dieser Abstraktion sind Web Services. [Melz08]

In seinem Buch *SOA? Serviceorientierung in Business und Software* spezifiziert Dieter Masak mehrere technische Eigenschaften von Services innerhalb einer SOA, welche diese als Framework, aber auch als Designmodell repräsentieren: [Masa07]

- **Lose Koppelung:**
Die lose Koppelung ist eine Grundbedingung von wechselwirkenden Services, da diese plattform- und implementierungsneutral sein müssen. Demnach ist für den Dienstinutzer die Implementierung irrelevant, es muss ihm jedoch die Schnittstelle bekannt sein. Des Weiteren ist die lose Kopplung notwendig, um flexibel den Serviceanbieter wechseln zu können. Die ist nützlich um auf andere Anbieter beziehungsweise Services auszuweichen, falls der Dienst ausfällt, oder dieser aufgrund anderer Umstände getauscht oder verändert werden muss. Eine enge Koppelung würde einen gewünschten Wechsel jedoch vehement erschweren.
- **Nachrichtenbasierte Interaktion:**
Die Kommunikation zwischen einzelnen Services verläuft immer nachrichtenbasiert, was auch einer losen Koppelung zugutekommt. Services sind formell durch den Nachrichtenaustausch zwischen Anbieter- und Nachfrager-Agenten, und nicht durch die Eigenschaften der Agenten selbst bestimmt. Die interne Struktur eines Agenten – beispielsweise seine Implementierung, Plattform, Prozess- und Datenbankstruktur – wird ganz bewusst von einer SOA weg abstrahiert.
- **Dynamische Auffindung:**
Um Services verwenden zu können, müssen potentielle Anwender dieser auch über deren Existenz erfahren können und informiert werden, wie sie diese zu verwenden haben. Deshalb muss ein entsprechendes Modell zum dynamischen Auffinden von Services etabliert werden.
- **Portabilität:**
Services sollten soweit wie möglich plattformunabhängig sein.
- **Implementierungsneutralität:**
Services definieren sich ausschließlich über ihre Qualitäten und ihre eigenen Interfaces, nicht jedoch über ihre jeweilige Implementierung. Ohne diese Prämisse wäre ein Provider- oder Servicewechsel nur schwer möglich.

¹⁰ Remote Procedure Call - ein Protokoll für verteilte Anwendungen zum Aufruf einer entfernten Prozedur

¹¹ Remote Method Invocation von Sun Microsystems – ein Kommunikationsprotokoll und eine Programmierschnittstelle der Programmiersprache Java zum Aufruf einer entfernten Prozedur

- **Autonomie:**
Services müssen selbstständig und ohne Abhängigkeit anderer Services funktionieren.
- **Konfigurierbarkeit:**
Services sollten dynamisch konfigurierbar sein, um auf mögliche Veränderungen reagieren zu können.
- **Späte Bindung:**
Da Services dynamisch aufgerufen werden, findet die eigentliche Bindung erst zur Laufzeit statt.
- **Richtlinienbasiertes Verhalten:**
Bestimmte technische Eigenschaften wie Transaktionssicherheit und Verschlüsselung, sowie Kontext, sollten keine festen Bestandteile der Serviceimplementierung sein, da diese Thematiken auf einer anderen Abstraktionsstufe behandelt werden können und es dem Nutzer dann möglich ist, anwendungsbezogen zu entscheiden, ob diese Eigenschaften gewünscht beziehungsweise gefordert sind, oder nicht.

Auch weitere Serviceeigenschaften innerhalb einer SOA sind noch nennenswert. Sie sind weiters beschreibungsorientiert, da sie in einer maschinenlesbaren Form mittels Metadaten beschrieben werden, wobei nur Informationen, die für den Gebrauch der Services notwendig sind, in der Beschreibung veröffentlicht werden. Die Semantik eines Services sollte entweder direkt oder indirekt durch die Beschreibung dokumentiert sein.

Services werden in SOA oft über ein Netzwerk verwendet; dies ist jedoch keine absolute Notwendigkeit. Die Granularität in einer SOA ist zwar beliebig, wird jedoch häufig in Services nur eine kleine Anzahl von Operationen durchgeführt, bei denen die Nachrichten dafür relativ groß sind.

Nachrichten werden plattformunabhängig in einem standardisierten Format über Schnittstellen übermittelt. Dabei ist XML ein häufig angewendetes Format, das diese Bedingungen erfüllt. Dadurch ist es unbedeutend, welche Hard- oder Software, Programmiersprache oder Betriebssystem die beteiligten Entitäten einsetzen.

SOA ist nicht unbedingt eine neue Idee; eine serviceorientierte Architektur konnte nämlich schon in älteren Technologien, wie beispielsweise CORBA¹², umgesetzt werden. Eigentlich handelt es sich um eine Umsetzung altbekannter Prinzipien aus einer technischen und einer fachlichen Sichtweise. Die einzelnen Glieder der fachlichen Prozesse (Geschäftsprozesse) werden zuerst als fachliche Dienste (Business Services) modelliert, die dann durch technische Dienste in der IT-Schicht umgesetzt werden. [Melz08]

Im Kern entspricht SOA somit einem technischen Managementkonzept, wohinter ein aufbauendes Konzept der Software-Architektur steht. Um effizient auf veränderte geschäftliche Anforderungen reagieren zu können, sollten die Dienste der IT-Infrastruktur an den Geschäftsprozessen des Unternehmens ausgerichtet sein. [OvTu08]

Die aus einer Geschäftslogik stammenden Geschäftsprozesse und -regeln können mittels einer SOA somit dynamisch mittels Services komponierter Anwendungen abgebildet werden. Bei der

¹² Common Object Request Broker Architecture - eine Spezifikation für eine objektorientierte Middleware von der Object Management Group (OMG).

Änderung der Geschäftslogik kann damit flexibel und dynamisch auch die darunter liegende Service-Applikation durch die Veränderung einzelner Services angepasst werden.

Unter anderem können somit aus den Eigenschaften von Services innerhalb einer SOA auch grundlegende Merkmale einer SOA abgeleitet werden. Diesbezüglich gibt es in der Literatur auch unterschiedliche Auffassungen, welche Eigenschaften zwingende Merkmale einer SOA sind.

Einigkeit herrscht bei der losen Kopplung der Dienste, was als eines der wesentlichsten Merkmale einer SOA gilt, welche schon bei den Serviceeigenschaften beschrieben wurden. Kurz wiederholt bedeutet dies, dass Services von Anwendungen oder anderen Services bei Bedarf dynamisch gesucht, gefunden und eingebunden werden können. Die Bindung findet jedoch erst zur Laufzeit statt, was bedeutet, dass zum Zeitpunkt der Übersetzung des Programms noch nicht bekannt sein muss, welcher Dienst zur Laufzeit aufgerufen wird. Die dynamische Bindung ist somit das zweite wesentliche Merkmal einer SOA.

Ein weiteres grundlegendes Merkmal ist das Vorhandensein eines Verzeichnisdienstes, einer sogenannten Registry, bei dem vorhandene Dienste registriert werden können. Innerhalb des Verzeichnisdienstes kann nach registrierten Services gesucht werden, um diese anschließend nutzen zu können. Die Dienste – beziehungsweise deren Schnittstellen - werden im Verzeichnisdienst dabei in einer maschinenlesbaren Beschreibungssprache definiert, um den Suchenden über die Funktion des Dienstes zu informieren. Der potentielle Dienstanutzer muss jedoch die Beschreibung interpretieren können, da sonst keine Bindung zustande kommen könnte. Dies erfordert wiederum die Verwendung von offenen Standards (ist ebenfalls ein wichtiges Merkmal einer SOA), um eine breite Verbreitung und Akzeptanz dieser Architekturen zu ermöglichen.

Auch die effiziente Umsetzung des Prinzips der Wiederverwendung von Services wird in der Literatur oft als wichtiges Merkmal einer SOA geschildert. Dadurch können Ressourcen eingespart werden, da eine sonst notwendige Neuentwicklung verhindert werden kann. Der Grad der Wiederverwendbarkeit nimmt jedoch mit der Größe und Komplexität des Dienstes ab, da dieser dadurch schon an ein spezielleres Problem gerichtet ist. Einfachheit ist deshalb eine oft geforderte Voraussetzung für eine SOA, wobei "zu einfache" Services wiederum zu einer zu hohen Kapselung und dadurch zu einem stark steigenden Verwaltungsaufwand führen können. Mit der Wiederverwendung sind jedoch nicht nur einzelne Dienste, sondern auch aus "Dienstketten" entstandene ganze Systeme gemeint. Es gilt somit von Fall zu Fall ein notwendiges Mittelmaß zu finden, um den Grad der Wiederverwendbarkeit zu optimieren.

Die Prozessorientierung ist ein weiterer Aspekt einer SOA, welche eine funktionale Zerlegung der Anwendungen erleichtert, wodurch einzelne Dienste zu einem gemeinsamen Zweck zusammengestellt werden können, um daraus einen geschäftlichen Mehrwert zu erzielen. Die Modellierung von Geschäftsprozessen ist in Unternehmen gang und gäbe, da sie einen Mehrwert aufgrund des Ansatzes erzielen, damit bereits vorhandene Dienste integriert werden. Ein sogenannter komponierter Dienst ist flexibler gegenüber Änderungen seines Ablaufs, da eine Prozessänderung nicht mehr zu einer neuen Implementierung, sondern zu einer neuen Komposition führt.

Dieter Masak definiert einen Geschäftsprozess im Kontext der Softwareentwicklung als: "...die inhaltlich abgeschlossene, zeitlich-sachlogische Abfolge von Funktionen ... die zur Bearbeitung eines für die Leistungserbringung der Organisation relevanten Objekts erforderlich ist." [Masa07]

Weiters erkennt er mehrere Charakteristiken von Geschäftsprozessen:

- Ein Geschäftsprozess besteht aus einer Reihe von Aktivitäten.
- Für Geschäftsprozesse existiert ein Maß für den Erfolg.
- Jeder Geschäftsprozess enthält Steuer- und Kontrollmechanismen.
- Ein Geschäftsprozess kann formal beschrieben werden.
- Jeder Geschäftsprozess besitzt eine Anzahl von sozialen und physischen Randbedingungen.
- Jeder Geschäftsprozess existiert nur in einem definierten Kontext.
- Jeder Geschäftsprozess hat einen eindeutigen In- und Output.
- Ein Geschäftsprozess hat einen zeitlichen Anfang und ein zeitliches Ende.

Es ist unbedingt notwendig, die Ergebnisse der einzelnen Aktivitäten eines Prozesses zu koordinieren, wobei eine einzelne Aktivität einer konkreten Prozess-Instanz zugeordnet werden muss. Das Ergebnis einer Interaktion ist abhängig vom Zustand, in dem sich der Prozess befindet. Falls ein Vorgang zu einem fehlerhaften Zustand führt, muss es möglich sein eine oder mehrere Aktionen wieder rückgängig zu machen. Um dies zu ermöglichen, müssen Transaktionen definiert werden, auf die später in einem eigenen Kapitel eingegangen wird. [Melz08]

Bei der Komposition von Services zu einem Prozess kann zwischen der Orchestrierung und der Choreographie unterschieden werden.

Die Orchestrierung beschreibt den Workflow (Arbeitsablauf), wie Services interagieren sollen (beinhaltet nämlich die Geschäftslogik und somit den Fluss der Service-Aufrufe und die Reihenfolge der Aktivitäten, inklusive Aufrufparameter), wobei der Prozessfluss von einer Kontrollinstanz überprüft und gesteuert wird. Jeder Dienst hat dabei einen eingeschränkten Sichtbereich.

Im Gegensatz zur Orchestrierung gibt es bei der Choreographie kein zentrales Kontrollorgan. Sie beschreibt die Sequenz von Nachrichten zwischen Diensten mit dem Fokus auf einen dialogorientierten und öffentlichen Austausch von Nachrichten zwischen den Diensten, wobei jeder Dienst seine eigene Aufgabe in der gesamten Komposition beschreibt.

Für einen erfolgreichen Einsatz einer SOA sind Sicherheit und Akzeptanz notwendige Voraussetzungen. Die an einer SOA beteiligten Entitäten können grundsätzlich in drei verschiedene Rollen unterteilt werden: Service-Anbieter, Service-Nutzer und Service-Vermittler. Abbildung1 soll dies verdeutlichen. ([Melz08][Diet06] et al.)

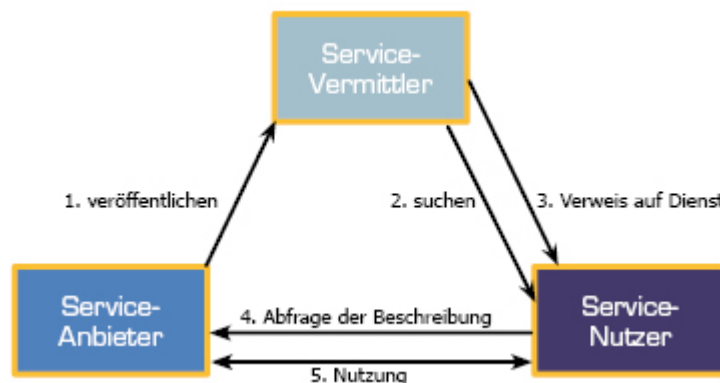


Abbildung 1: Beteiligte einer SOA

Auf einer Plattform des Service-Anbieters läuft zumindest ein Dienst, auf den über eine definierte Schnittstelle zugegriffen werden kann. Der Service-Nutzer greift wiederum über diese Schnittstelle auf den Dienst des Service-Anbieters zu. Die Kommunikation zwischen den beiden läuft dabei über ein Protokoll, das beide verstehen müssen, welches, wie auch die Dokumentation, aus der Servicebeschreibung hervorgeht. Dabei müssen wieder die offenen Standards betont werden, ohne die ein solches Vorhaben massiv beeinträchtigt wäre.

Um Dienste publizieren und finden zu können ist ein Verzeichnis notwendig, welches die Rolle des Service-Vermittlers einnimmt. Dabei muss der Service-Anbieter selbst aktiv werden und seinen Dienst in diesem Verzeichnis registrieren. Das Verzeichnis speichert unter anderem auch die Servicebeschreibung der einzelnen Dienste, um die anfragenden potentiellen Dienstanutzer vorab zu informieren, wie sie auf die Dienste im Verzeichnis zugreifen können.

Um die Verwaltung des Verzeichnisses zu vereinfachen, ist eine Unterteilung in verschiedene Kategorien sinnvoll. Um eine gute Klassifizierung zu ermöglichen, muss jeder Dienst genau einer Kategorie eindeutig zugeordnet werden. Der Verzeichnisdienst muss jedoch nicht von einer einzigen Stelle verwaltet werden, sondern könnte auch ähnlich wie das Domain Name Service in Hierarchien unterteilt werden. Unerheblich ist es jedoch, ob sich Service-Anbieter und -Nutzer bekannt sind.

Nach der erstmaligen Verwendung des Dienstes durch den Nutzer kann bei fortlaufender Nutzung der Zwischenschritt über den Verzeichnisdienst entfallen, was in der Praxis auch durchaus üblich ist, wodurch jedoch die lose Kopplung aufgeweicht wird. Dadurch wird zwar die Laufzeit verbessert, die Flexibilität jedoch eingeschränkt.

Durch Aktionen, wie die Veröffentlichung eines Dienstes, das Suchen von Diensten und deren dazugehörigen Dienstbeschreibungen, die Interaktion mit Diensten und die Bestimmung deren Benutzungsrichtlinien, und abschließend eine erfolgreiche Bindung an einen Dienst ermöglichen somit ein funktionierendes Zusammenspiel zwischen Service-Anbieter, -Nutzer und -Verzeichnis. [Melz08]

Für die Aufrechterhaltung des Servicebetriebs ist meist der Dienstanbieter selbst verantwortlich. Solche Fragen sollten generell in einem Servicevertrag zwischen Anbieter und Nutzer beantwortet werden. Innerhalb des Vertrages werden diverse Bedingungen spezifiziert,

wie das Service benutzt werden soll beziehungsweise welche Rechte und Pflichten die daran Beteiligten eingehen. Wesentliche Bestandteile eines solchen Vertrages können, unter anderem, Sicherheitsbestimmungen, Preise, Zahlungsmodalitäten, Qualitätsrichtlinien, Datensicherungen, Wartungsregelungen, Funktionsbeschränkungen und Verhaltensregeln sein.

Das Thema SOA ist im Bereich der Organisationsstrukturierung in aller Munde und hat seinen Siegeszug im Wesentlichen mit dem Schlagwort Flexibilität angetreten. Eine SOA schränkt sich auf den unteren Ebenen ein, wodurch die Interaktionsmöglichkeiten mit den Diensten klar definiert sind. Ausschließlich verlässlich definierte Schnittstellen ermöglichen den flexiblen Einsatz von Diensten auf höheren Ebenen. [Melz08]

Allerdings kann von der Flexibilität meist erst auf Ebene der Prozessmodellierung profitiert werden. Bei einer sinnvoll implementierten SOA können Prozesse zügig umgestaltet werden, ohne die darunter liegenden Komponenten zu ändern, was wiederum zu agilen Prozessen führt, die mittels standardisierten Elementen realisiert werden.

Mit einer sinnvoll implementierten SOA ist besonders die gewählte Granularität gemeint. Eine zu grobe Granularität schränkt die Möglichkeit der Wiederverwendung stark ein, wobei eine zu feine Granularität zu einem erhöhten Entwicklungs- und Verwaltungsaufwand führen könnte, und die Umsetzung der fachlichen Komponenten erschwert wird.

Innerhalb einer SOA können verschiedene Phasen identifiziert werden, die jeweils mit unterschiedlichen Tätigkeiten verbunden sind, was zu einem Service-Lebenszyklus führt. SOA-Governance betitelt dabei die Entscheidungen, Aktivitäten, Rollen und Verantwortlichkeiten zur Regulierung und Kontrolle einer SOA. Dabei kann zwischen folgenden Phasen unterschieden werden: [Melz08]

- Entwurfsphase mit Servicedefinition, -abgrenzung und -implementierung.
- Laufender Betrieb mit Inbetriebnahme und Geschäftsaktivität.
- Änderungszeitpunkte
- Außerbetriebnahme, wenn Services nicht mehr benötigt werden.

In vielen Unternehmen sind heutzutage noch veraltete Organisationsstrukturen im Einsatz. Vor allem was die IT-Struktur betrifft, haben die unterschiedlichsten Abteilungen die verschiedensten IT-Systeme im Einsatz. Diese Systeme sind oft im Laufe der Zeit mit den Anforderungen gewachsen und repräsentieren oft nicht standardisierte Einzellösungen. Diese Insellösungen haben ihren Zweck in der einzelnen Abteilung sicherlich häufig erfüllt; es kommt jedoch dadurch bei abteilungs- beziehungsweise systemübergreifenden Prozessen zu Schwierigkeiten an den Systemübergängen. Ein Datenaustausch mit anderen Systemen ist bei solch vertikalen Architekturen nur mittels expliziter Schnittstellenentwicklung notwendig. Anpassungen an Prozessen oder besonders eine Neumodellierung des gesamten Prozesses stellen große Herausforderungen dar.

SOA hat sich vor allem für Beratungs-Unternehmen und Werkzeughersteller zu einem Hype entwickelt. Verschiedenste Versprechen werden durch die Umsetzung einer SOA auf die bestehende Organisationsstruktur von Unternehmen gemacht, wobei vor allem Einsparungsmöglichkeiten und Strukturverbesserungen prognostiziert werden.

Dieter Masak identifiziert dabei oft behauptete Vorteile, die aus der Einführung einer SOA resultieren: [Masa07]

- Implementierung großer Enterprise-Systeme:
Mittels SOA und den entsprechenden Managementwerkzeugen können auch große Systeme einfach organisiert werden. Services können leicht ersetzt, eingebunden und extern zugänglich gemacht werden. Im Gegensatz dazu sind mit enger gekoppelten Architekturen derartige Änderungen am System mit einem weitaus höheren Programmier- und Konfigurationsaufwand verbunden.
- Skalierbarkeit:
Da eine SOA nur wenige Annahmen über das Netzwerk trifft, werden dadurch die Abhängigkeiten von anderen Services minimiert.
- Hohe Wiederverwendbarkeit und lose Koppelung:
Die Integration neuer Services wird erleichtert und es können Services auch ohne großen Aufwand in einem anderen Kontext verwendet werden.
- Abstraktion:
Durch die Modellierung von Applikationen in offenen Umgebungen kann eine Abstraktion der Funktionen geschaffen werden.
- Flexibilität:
Das einfache Austauschen und Zusammensetzen der Services ermöglicht es, einfacher und rascher Änderungen am System durchzuführen als bei traditionellen oder objektorientierten Systemen.
- Standardisierung und Unterstützung durch ein breites Spektrum von Anbietern:
Viele Organisationen (z.B. IBM, Microsoft, Sun) beschäftigen sich mit dem Entwickeln von Standards für SOA. Eine breite Unterstützung durch Werkzeuge erleichtert das Erstellen und Veröffentlichen von Services.

Unternehmen wird beispielsweise durch den Einsatz einer SOA in Aussicht gestellt: eine erhöhte Flexibilität zu gewinnen, Anpassung der Geschäftsprozesse zu beschleunigen, die Integration mit internen und externen Partnern zu vereinfachen, kürzere Produktlebenszyklen zu erlangen, Kosten einzusparen, den Return on Investment zu verbessern und die Kontrolle zu erleichtern. [Masa07] [Melz08]

Jedoch ist SOA für jedes Unternehmen individuell zu betrachten. Eine sogenannte Standard-Implementierung einer SOA ist nicht vorhanden, weshalb Unternehmen eine SOA entsprechend den eigenen Bedürfnissen und Bedingungen planen müssen.

Die Einführung einer SOA kann für ein Unternehmen eine Verbesserung in verschiedenen Bereichen erzielen, sodass die daraus entstanden Vorteile die Nachteile überwiegen. Dies muss jedoch nicht der Fall sein. Demnach gilt es, die Chancen, Möglichkeiten und Risiken des Serviceorientierungsparadigmas individuell abzuwägen und zu entscheiden, ob eine Einführung dem Unternehmen als Gesamtheit eher hilft oder schadet. Vorab gemachte Versprechungen sind somit nicht als sinnvoll zu erachten.

2.3.2 Web Services

SOA ist ein abstraktes Modell, bei dem Web Services als eine mögliche technische Instanz angesehen werden können, und sind momentan wohl die am weitesten fortgeschrittene und verbreitete Umsetzung des SOA-Konzepts. Web Services haben grundsätzlich somit auch die zuvor beschriebenen Eigenschaften von Services innerhalb einer SOA.

Jedoch sind Web-Services-Architekturen nicht die erste Umsetzung einer SOA, da auch schon andere Konzepte ähnliche Probleme adressierten und diese zumindest theoretisch auch erfolgreich gelöst haben. Beispiele dafür sind unter anderem COM/DCOM¹³, RMI und CORBA. Von diesen Konzepten ist jedoch CORBA die einzige plattformunabhängige Lösung. Sie ist außerdem an keine Programmiersprache gebunden. Der Kern der CORBA-Spezifikation ist sein Objektmodell, wobei anfangs keine befriedigende Abbildung der Datentypen zu anderen Programmiersprachen existierte und die Spezifikation nicht vollständig war. [Melz08]

Web Services bieten ein grundlegendes, flexibles und allgemeines Gerüst des Nachrichtenaustauschs, welches durch den Einsatz zusätzlicher Spezifikationen erweitert werden kann, um beispielsweise Sicherheitsregeln zu spezifizieren, Transaktionen zu realisieren oder die Komposition von Prozessen zu ermöglichen.

Web Services sind Schnittstellen zu Anwendungen, auf die über ein Netzwerk zugegriffen werden kann, wobei standardisierte Internetprotokolle für die Kommunikation eingesetzt werden. Sie sind dabei nicht an einziges Transportprotokoll gebunden. Es werden beispielsweise meist HTTP oder SMTP als Transportprotokolle eingesetzt. Als Nachrichtenformat wird überwiegend XML verwendet.

Web Services sind eine Technik zur Kommunikation zwischen Maschinen und unterstützen dadurch die Interaktion von Software-Agenten, wobei ein Mensch zwar der Auslöser der Kommunikation sein kann, das Web Service dabei aber nur indirekt nutzt. Deshalb können Web Services auch als RPCs verwendet werden, sind aber mehr als nur ein Nachfolger dieser.

Urban Bettag charakterisiert Web Services weiter durch folgende Merkmale: [Bett01]

- Programmierbar: Sie sind über programmierbare Schnittstellen erreichbar.
- Selbstbeschreibend: Sie werden von Metadaten (sogenannte Daten über Daten) begleitet, die während der Laufzeit von anderen Web Services ausgewertet werden können.
- Kapselung: Sie sind unabhängige, in sich abgeschlossene, gekapselte Anwendungen, die eine genau definierte Aufgabe erfüllen.
- Ortstransparenz: Sie können jederzeit und von jedem Ort aus aktiviert werden, falls Benutzer und Anwendungen entsprechende Zugriffsrechte besitzen.
- Protokolltransparenz: Sie basieren auf Internet-Protokollen. Operationen und Nachrichten können mehrere Protokolle unterstützen, z.B. HTTP oder SMTP¹⁴.

¹³ Das "Component Object Model" bzw. das "Distributed Component Object Model" von Microsoft

¹⁴ Simple Mail Transfer Protocol – Internetprotokoll für den Empfang von E-Mails.

- Komposition: Sie können entweder in weitere Web-Services zerlegt werden oder mehrere wieder verwendbare Basis-Web-Services können zu einem neuen Web-Service zusammengestellt werden.

Die drei wichtigsten Komponenten einer SOA werden in Web Services meistens durch die Basistechnologien SOAP¹⁵, WSDL¹⁶ und UDDI¹⁷ realisiert.

SOAP beschreibt ein Protokoll zum Austausch von Nachrichten und dessen Einbettung in ein Transportprotokoll, wobei XML als Nachrichtenformat eingesetzt wird. WSDL ist eine Metasprache die zur Beschreibung von Web Services dient und basiert ebenfalls auf XML. UDDI stellt die Realisierung eines Verzeichnisdienstes für Web Services dar.

SOAP und WSDL haben sich im Umfeld von Webservices bereits durchgesetzt, was bei UDDI noch nicht klar behauptet werden kann. Wie auch bei SOAP und WSDL gibt es für UDDI Alternativen, wie zum Beispiel die vergleichsweise einfachere Web Services Inspection Language. [Melz08]

Vom Ablauf her muss ein Serviceanbieter zuerst eine Schnittstellenbeschreibung im WSDL-Format erstellen und dieses sogenannte WSDL-Dokument in einem UDDI-Verzeichnisdienst veröffentlichen. UDDI selbst stellt ebenfalls eine SOAP-Schnittstelle zur Verfügung, um das Verzeichnis nach Web Services durchsuchen zu können. Zur Verwaltung der Metadaten – wie z.B. Informationen zur Auffindung – von Web Services spezifiziert UDDI eine standardisierte Verzeichnisstruktur. Wenn sich ein potentieller Dienstinutzer für ein Web Service entschieden hat, fordert dieser die Schnittstellenbeschreibung vom Verzeichnisdienst an, welcher wiederum eine Referenz auf das entsprechende WSDL-Dokument zurückliefert. Durch die WSDL-Beschreibung ist dem Dienstinutzer nun bekannt, wie er mittels SOAP mit der Anwendung kommunizieren kann und ist somit in der Lage das Web Service einzusetzen.

Das World Wide Web Consortium (W3C) definiert die Web-Service-Architektur in ihrem dazugehörigen Dokument [W3C04] und ist auch für die Spezifikation von einigen Web-Service-Spezifikationen hauptverantwortlich. Unter anderem unterliegt dem W3C etwa SOAP und WSDL.

Das W3C definiert beispielsweise in der *Web Services Architecture* wie folgt: "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." [W3C04]

Bei einer entsprechenden Umsetzung ist es somit möglich, Anwendungen und deren Eigenschaften dynamisch zu entdecken. Mittels Web Services können Verbindungen zwischen Anwendungen entsprechend dieser Eigenschaften dynamisch aufgebaut werden. Anschließend können die Anwendungen Nachrichten austauschen, um gemeinsam eine bestimmte Funktion zu realisieren. Unter gewissen Umständen ist dies auch vollautomatisch und ohne menschliche Beteiligung möglich. [Melz08]

¹⁵ SOAP – Anfangs für Simple Object Access Protocol, nun kein Akronym.

¹⁶ WSDL – Web Services Description Language

¹⁷ UDDI – Universal Description, Discovery and Integration

Es gibt auch für die Beschreibung von Geschäftsprozessen für Web Services diverse Standards, wobei vor allem BPEL¹⁸, WSCI¹⁹ und BPML²⁰ nennenswert sind, die es erlauben Web Services zu Geschäftsprozessen zu verknüpfen. Alle drei Standards verwenden das XML-Format. Diese Standards erlauben es Prozesse zu modellieren und aus einzelnen Aufgaben eine komplexe Anwendung zu erstellen. Vor allem der BPEL ist besonders vielversprechend, da er von mehreren großen Konzernen (z.B. Microsoft und IBM) unterstützt wird.

Vorurteile und Gerüchte haben Web Services (z.B. sie benötigen keine Programmierung, sind unsicher und an HTTP gebunden) besonders am Anfang auch verfolgt. So wurden sie anfangs als langsam bezeichnet, da die meisten Standards XML als Format einsetzen und dieses zunächst nur langsam zu parsen war. Deshalb ist die Geschwindigkeit heute noch ein relevantes Thema. [Melz08]

SOAP

SOAP ist die Kommunikationskomponente von Web Services und wird, wie bereits erwähnt, vom W3C spezifiziert. Die aktuelle Version ist die 2007 veröffentlichte SOAP 1.2 und wird im Dokument *SOAP Version 1.2 Part 1: Messaging Framework* [W3C07] genau spezifiziert und hat den Status einer "Recommendation". In der Spezifikation wird der grundlegende Aufbau einer SOAP-Nachricht beschrieben. SOAP ist weder an eine bestimmte Plattform, noch an eine spezielle Programmiersprache gebunden.

In XML-Notation beschreibt SOAP das Nachrichtenformat zur Kommunikation mit Web Services, wobei für die Übermittlung der Nachrichten in ein fast beliebiges Transportprotokoll verwendet werden kann. Ebenfalls bei SOAP werden alle XML-Elemente auch nach Groß- und Kleinschreibung unterschieden (ist somit "case sensitive").

Zu Beginn war SOAP die Abkürzung für "Simple Object Access Protocol". Beim W3C herrschte jedoch Einigkeit, dass das Protokoll nach mehreren Erweiterungen nicht mehr einfach war; deshalb auch die Bezeichnung nicht mehr zutreffend war. Man konnte sich allerdings auf keine Alternative einigen und der Begriff SOAP war schon so weit verbreitet, weshalb beschlossen wurde, dass SOAP kein Akronym ist. [Melz08]

Das W3C beschreibt SOAP in deren Spezifikation grundlegend wie folgt: "SOAP Version 1.2 (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics." [W3C07]

Dabei sind die Einfachheit und Erweiterbarkeit zwei grundlegende Entwurfsziele von SOAP. Man versucht diese zu erreichen, indem Merkmale, die bei anderen verteilten Systemen besonders charakteristisch sind (z.B. Sicherheit, Zuverlässigkeit, Routing, "Message Exchange Patterns" (MEPs), etc.) bewusst vom Nachrichten-Framework wegzulassen. Diese Merkmale können dafür wiederum als Erweiterungen in anderen Spezifikationen definiert werden. [W3C07]

¹⁸ BPEL – WS-Business Process Execution Language

¹⁹ WSCI – WS-Choreography Interface

²⁰ BPML – Business Process Modeling Language

Um die Kommunikation zwischen Sender und Empfänger zu ermöglichen, muss ein Verfahren gewählt werden, das beide einheitlich interpretieren können. Zuerst wählt der Sender einen Kommunikationskanal aus, über den der Empfänger erreichbar ist. Daraufhin verfasst der Sender eine Nachricht in einem für den Empfänger verarbeitbaren Format und serialisiert diese, um sie über den gewählten Kanal versenden zu können. Der Empfänger wiederum muss diese Nachricht im Anschluss an den Transport deserialisieren und kann deren Inhalt danach lesen.

Es wird in keiner SOAP-Spezifikation vorbestimmt, mit welchem Protokoll die Nachricht zu übertragen ist, jedoch wird beschrieben, wie eine SOAP-Nachricht mittels eines Protokolls übertragen werden soll (SOAP Protocol Binding). Das Protokoll muss von den Anwendern entsprechend den Anforderungen und Bedingungen gewählt werden. Die bestehende IT-Infrastruktur könnte diesbezüglich ausschlaggebend sein. [Melz08]

Der Web-Service-Anbieter muss angeben, an welchen Punkt die SOAP-Anfrage gerichtet werden soll. Der Anfragepunkt besteht aus einer URL, die den Server identifiziert und dem dazugehörigen Auftragsnamen (z.B. `http://example.com/SOAPdienst`). Die genaue Umsetzung der Zugriffslogik hängt von der jeweiligen SOAP-Implementierung ab.

Beispiel einer SOAP-Nachricht:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:SOAPtest xmlns:n="http://example.com/SOAPtest ">
      <n:keyID>12345</n:keyID>
      <n:priority>1</n:priority>
    </n:SOAPtest>
  </env:Header>
  <env:Body>
    <m:SOAPmsg xmlns:m="http://example.com/SOAPmsg">
      <m:msg>SOAP Nachricht</m:msg>
    </m:SOAPmsg>
  </env:Body>
</env:Envelope>
```

Tabelle 4: Beispiel einer SOAP-Nachricht

Eine SOAP-Nachricht ist ein XML-Dokument und hat ein SOAP-Envelope als übergeordnetes Wurzelement, welches wiederum aus SOAP-Header und SOAP-Body besteht. Mittels Angabe der URI²¹ wird festgelegt, welche Version der SOAP-Spezifikation in der SOAP-Nachricht verwendet wird. Außerdem wird dadurch der Namensraum im XML-Dokument bestimmt, der an den Präfix *env* gebunden ist.

Das erste Kindelement des SOAP-Envelopes ist der optionale SOAP-Header, der innerhalb einer Nachricht nur einmal vorkommen darf. Der Header ist jedoch in der Spezifikation nicht näher definiert. Er wird beispielsweise von auf SOAP aufbauenden Spezifikationen verwendet, kann aber auch sonst zur Angabe beliebiger Informationen verwendet werden (wie z.B. Sicherheitsinformationen). Alle direkten Kindelemente des SOAP-Headers müssen mit einem Namensraum versehen werden, der jedoch nicht zwingend mit einer URI versehen werden muss.

²¹ Uniform Resource Identifier – Identifikator zur Bezeichnung von Ressourcen (z.B. Webseiten)

Für die Übertragung von SOAP-Nachrichten können auch Zwischenstationen verwendet werden, wobei das Header-Element verwendet wird, um einzelne Zwischenstationen zu adressieren und von diesen bestimmte Aufgaben realisieren zu lassen.

Das zweite Kindelement einer SOAP-Nachricht ist der SOAP-Body, welcher zwingend vorhanden sein muss. Im Body sind die eigentlichen Nutzdaten einer Nachricht enthalten; es muss allerdings auch im XML-Format beschrieben werden. Der Aufbau des XML-Inhalts wird jedoch individuell von den Anwendern bestimmt, wobei wichtig ist, dass die Struktur allen Teilnehmern bekannt ist, um die Daten verarbeiten zu können. Somit lassen sich mit SOAP alle Informationen verschicken, die als XML-Dokument abgebildet werden können.

Die Auswertung der Daten nach der Übertragung der Nachricht ist allerdings nicht Aufgabe von SOAP, sondern der jeweiligen dahinter liegenden Anwendung, die diese entgegennimmt.

Wie bei jeder Kommunikation kann es auch bei SOAP im Laufe der Übertragung zu Fehlern kommen. Um mit diesen entsprechend umgehen zu können, wird in SOAP ein Fehlereintrag spezifiziert (sog. Fault). Wenn es bei Abhandlung der Kommunikation zu einem Fehler kommt, so kann eine SOAP-Nachricht einen Fehlereintrag als einziges Kindelement des SOAP-Body beinhalten, wobei der Namensraum des Fehlereintrags den URI "http://www.w3.org/2002/12/soap-envelope" angeben muss.

In nachfolgender Tabelle wird aufgeschlüsselt, aus welchen Elementen ein Fehlereintrag besteht:

Elementname	Verpflichtend	Beschreibung
Code	Ja	Eine von der SOAP-Spezifikation festgelegte Codierung der Fehlerquelle
Reason	Ja	Textbasierte Beschreibung des aufgetretenen Fehlers
Node	Nein	Beschreibt, an welcher Stelle der SOAP-Kommunikation der Fehler aufgetreten ist
Role	Nein	Beschreibt die Rolle des Node, bei dem der Fehler aufgetreten ist
Detail	Nein	Enthält weitere Informationen zum aufgetretenen Fehler. Der Inhalt des Detailelements kann von der Anwendung frei festgelegt werden.

Tabelle 5: SOAP - Elemente eines Fehlereintrags

Das Element *Code* enthält wiederum die zwei Kindelementen *Value* und *Subcode*. *Value* enthält den Fehlercode der Fehlermeldung und ist verpflichtend.

In der Spezifikation werden die Fehlercodes vorgegeben und sind in der Nachfolgenden Tabelle aufgelistet. *Subcode* hat keine vordefinierten Fehlercodes und ist ein rekursiv aufgebautes, optionales Kindelement von *Code*, und ermöglicht die genauere Beschreibung des Fehlercodes.

SOAP Fault Code	Beschreibung
VersionMismatch	Der Knoten in der SOAP-Kommunikation erwartet eine andere SOAP-Version.
MustUnderstand	Ein Knoten kann ein Pflichtelement eines SOAP-Header-Eintrags nicht auswerten.
DataEncodingUnknown	Es sind Datentypen aufgetreten, die nicht in eine SOAP-Nachricht übersetzt werden können.
Sender	Die SOAP-Nachricht konnte vom Sender nicht verarbeitet werden.

Receiver	Die SOAP-Nachricht konnte vom Empfänger nicht verarbeitet werden.
----------	---

Tabelle 6: SOAP - Fehlercodes

Wer einen tieferen Blick in den SOAP-Standard werfen möchte, findet eine detailgenauere Beschreibung, unter anderem, in der Spezifikation des W3C.

WSDL

Auch die Spezifikation der Web Service Description Language unterliegt der Standardisierung beim W3C und wird vom Konsortium in den Dokumenten *Web Services Description Language (WSDL) Version 2.0: Core Language* [W3C07a] und *Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts* [W3C07b] beschrieben. Auch dieses Dokument hat den Status einer "Recommendation". Die Version 2.0 wurde ursprünglich 1.2 betitelt, doch aufgrund der massiven Änderungen gegenüber dem Vorgänger in 2.0 umbenannt.

Eine plattformunabhängige Beschreibung von Schnittstellen ist ein wesentlicher Bestandteil einer Service-orientierten Architektur, die alle wesentlichen Informationen beinhalten, die für die Verwendung des Services vonnöten sind, wobei kein Wissen über die jeweilige Implementierung dieser erforderlich ist. Dadurch können Anwendungen flexibel aus Diensten unterschiedlicher Kommunikationspartner zusammengesetzt werden. [Melz08]

Das W3C beschreibt WSDL grundlegend in der Spezifikation wie folgt: "Web Services Description Language Version 2.0 (WSDL 2.0) provides a model and an XML format for describing Web services. WSDL 2.0 enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as how and where that functionality is offered." [W3C07a]

Somit kann mithilfe der WSDL sowohl eine abstrakte Beschreibung der Schnittstellen und den dazugehörigen Funktionen von Diensten, als auch eine konkrete Beschreibung der technischen Informationen, die zum Aufruf eines Dienstes benötigt werden, erstellt werden. Auch die WSDL wird im XML-Format abgebildet.

Wie SOAP ist auch WSDL mittels Erweiterungen (z.B. für Operations-Sicherheit, Nachrichtenaustauschmuster, semantische Dienstbeschreibung, etc.) ergänzbar.

Die WSDL trennt zwischen der abstrakten Beschreibung auf der Ebene der Funktionalität und der konkreten Beschreibung auf der Ebene der technischen Details, die bestimmen wo und wie ein Dienst angeboten wird.

Im Komponentenmodell unterscheidet WSDL zwischen den abstrakten Komponenten Operation, Message Exchange Pattern und Interface, und den konkreten Komponenten Binding, Endpoint.

Die Basis und somit das Wurzelement eines WSDL-Dokuments bildet das Description-Element. Dieses wiederum besteht aus den Kindelementen documentation, types, interface, binding und service, wobei die Anordnung der Kindelemente beliebig erfolgen kann.

Das optionale Element documentation bietet dem Anbieter Platz für eine individuelle und textbasierte Beschreibung des Dienstes, der beispielsweise dafür genutzt werden kann, um weitere Angaben für die Nutzung des Dienstes bereitzustellen und Ansprechpartner für einen Kontakt zu nennen.

Das Element `types` dient zur Definition von Datentypen und beschreibt, wie der Dienst Nachrichten sendet und empfängt. Normalerweise werden Nachrichten durch separat deklarierte Typendefinitionen bestimmt. Durch eine zentrale Definition verschiedener Nachrichtentypen wird eine leichte Wiederverwendung in mehreren Operationen oder über mehrere Schnittstellenbeschreibungen hinweg ermöglicht.

Im Element `interface` wird die abstrakte Funktionalität eines Dienstes und somit seine Schnittstelle beschrieben. Eine Schnittstelle besteht aus einer Menge von verschiedenen Operationen. Innerhalb der Schnittstellenbeschreibung kann eine beliebige Anzahl an Fehlerkomponenten mittels dem `fault`-Element erstellt werden. Die Operationen selbst werden wiederum durch das `operation`-Element abgebildet, wo wiederum die definierten Fehlerkomponenten verwendet (durch Verwendung des `outfault`-Elements) werden können. Eine Operation selbst definiert sich aus einer Menge an Nachrichten, die während der Kommunikation zwischen Anbieter und Nutzer ausgetauscht werden. Eingehende Nachrichten werden durch das `input`-Element und ausgehende Nachrichten durch das `output`-Element beschrieben. Ein Nachrichtentauschmuster (sog. Message Exchange Pattern – MEP) regelt noch für jede Operation in welcher Reihenfolge die einzelnen Nachrichten ausgetauscht werden müssen.

Die Nachrichtentauschmuster sind eine vordefinierte Spracherweiterung der WSDL und werden im Zusatzdokument [W3C07b] des W3Cs beschrieben. Sie werden durch das `pattern`-Element an ein Interface gebunden.

Abgesehen von der Nachrichtenreihenfolge kann mit Nachrichtentauschmustern beispielsweise die Häufigkeit von Nachrichten festgelegt oder nach welchen Regeln Fehlercodes generiert werden.

Somit wurden auf abstrakter Ebene die Schnittstellen, Operationen und die dazugehörigen Nachrichten beschrieben.

Über das `binding`-Element wird im WSDL-Dokument festgelegt, wie der Dienst aufgerufen wird. Dazu gehört, unter anderem, die Festlegung auf ein Nachrichtenprotokoll. Außerdem können operationsbezogen noch weitere Regeln für den Transport oder die Kodierung bestimmt werden. Meistens werden Web Services über das Nachrichtenprotokoll SOAP realisiert, was aber aus Sicht von WSDL keine zwingende Voraussetzung ist.

Das `service`-Element im WSDL-Dokument gibt an, wo sich Dienst befindet und ermöglicht für jedes Interface eines Dienstes die Definition von Zugangspunkten, über die auf den Dienst zugegriffen werden kann.

Durch die Mechanismen `include` und `import` wird es ermöglicht, andere Dokumentteile in ein WSDL-Dokumente einzubinden, was zu einer verbesserten Les- und Wartbarkeit dieser führt, und dadurch die Wiederverwendung von Dokumenten ermöglicht wird. Des Weiteren bietet WSDL Mechanismen zur Erweiterung der Sprache, wodurch Elemente und Attribute aus anderen Namensräumen in ein WSDL-Dokument eingebunden werden können. Dadurch wird beispielsweise ermöglicht, kryptografische Mittel zur Absicherung der Kommunikation einzusetzen. [Melz08]

Somit ist es allein auf Informationsbasis einer WSDL-Beschreibung möglich, mit einem Web Service zu kommunizieren, da sowohl der Aufbau der auszutauschenden Nachrichten, als auch die zu kontaktierende Gegenstelle beschrieben werden.

Wie auch schon am Ende der SOAP-Beschreibung sei es nochmals erwähnt, dass die Spezifikationen des W3C einen guten Ausgangspunkt für eine intensivere Betrachtung der Thematik bieten.

Verzeichnisdienste von Web Services

Verzeichnisdienste sind ein wesentlicher Bestandteil einer serviceorientierten Architektur und somit auch von Web Services. Verzeichnisdienste stellen eine zentrale Sammlung von Ressourcendaten dar und werden Benutzern über ein Netzwerk zugänglich gemacht, um innerhalb des Verzeichnisses nach Ressourcen suchen zu können.

Erst durch den Einsatz von Verzeichnisdiensten kann eine lose Kopplung von Diensten in einer SOA erreicht werden, da erst mit ihnen das dynamische Suchen, Finden und Nutzen von Diensten ermöglicht wird.

Ein entsprechender Einsatz eines Verzeichnisdienstes wird vor allem relevant, sobald es um die Verwaltung von einer größeren Anzahl von Diensten geht, da diese sonst nicht handhabbar wären. Ein Verzeichnisdienst sollte dabei über standardisierte Schnittstellen zugänglich sein und eine strukturierte Suche nach einem Dienst ermöglichen. [Melz08]

Die für Web Services vorläufig am meisten versprechenden Ausführungen von Verzeichnisdiensten sind einerseits die *Web Services Inspection Language* (WS-Inspection) und die *Universal Description, Discovery and Integration* (UDDI).

Die WS-Inspection, welche von den Unternehmen IBM und Microsoft entwickelt und spezifiziert [IBMI01] wurde, basiert im Vergleich zu UDDI auf einem weitaus simpleren Konzept zum Auffinden von Web Services. Die Spezifikation des UDDI-Standards unterliegt mittlerweile der *Organization for the Advancement of Structured Information Standards* (OASIS) und ist auf der Webseite dieser abrufbar [OASI04].

Dabei unterscheiden sich die Ansätze beider Standards komplett. Während UDDI versucht mit wenigen zentralisierten Verzeichnissen, die Dienste der Anbieter zu verwalten, arbeitet WS-Inspection mit vielen dezentralisierten Verzeichnissen, bei denen nur die Dienste von einem oder wenigen Anbietern gespeichert werden. [Melz08]

WS-Inspection

Die WS-Inspection wird von ihren Autoren wie folgt beschrieben: "The WS-Inspection specification provides an XML format for assisting in the inspection of a site for available services and a set of rules for how inspection related information should be made available for consumption. A WS-Inspection document provides a means for aggregating references to pre-existing service description documents which have been authored in any number of formats. These inspection documents are then made available at the point-of-offering for the service as well as through references which may be placed within a content medium such as HTML." [IBMI01]

WS-Inspection versucht dabei alle unterschiedlichen Informationsquellen, die für die Nutzung des Dienstes relevant sein könnten, zu verbinden und dies in einer Art darzustellen die sowohl einfach nutzbar als auch erstellbar ist. Die Einfachheit und Erweiterbarkeit sind auch die zwei grundlegenden Entwurfsziele der WS-Inspection, welche komplett dokumentenbasiert abgebildet wird und ebenfalls auf XML basiert. Ein WS-Inspection-Dokument ist im Wesentlichen eine Ansammlung von Verweisen auf Servicebeschreibungs-Dokumente

verschiedener Formate, wobei das daraus entstandene Dokument einfach zu schreiben und zu verwalten sein sollte. [IBMI01]

Das WS-Inspection-Dokument wird vom Anbieter nach seiner Fertigstellung auf seinem Webserver unter vordefiniertem Namen abgelegt. Ein Nutzer kann dieses Dokument dann beispielsweise durch die Nutzung des Internetprotokolls HTTP abrufen und aus dem Dokument eine Liste von Beschreibungen zur Nutzung der Dienste des Anbieters extrahieren.

Um die Entwurfsziele zu erreichen, ist das Datenmodell der WS-Inspection recht simpel gehalten und besteht aus den Datenstrukturen Inspection, Service und Link.

Ein WS-Inspection Dokument hat ein Inspection-Element zur Basis, welches somit das Wurzelement bildet. Das Inspection-Element kann wiederum beliebig viele Service- und Link-Kindenelemente haben. Während Service-Elemente der Definition von Diensten dienen und deren technische Beschreibung beinhalten, verweisen Link-Elemente auf externe Datenquellen (z.B. andere WS-Inspection-Dokumente oder UDDI-Verzeichnisse). Das optionale abstract-Element kann als Kindelement eines Service-Elements verwendet werden, um den Dienst mit Text zu beschreiben. Link-Elemente können auch eingesetzt werden, um Hierarchien von WS-Inspection-Dokumenten zu erstellen, damit ihre Verwaltung noch weiter vereinfacht wird. Es gibt für den WS-Inspection-Standard bereits Erweiterungen für UDDI und WSDL. WS-Inspection-Dokumente können auch dynamisch erzeugt werden (z.B. über ein Skript aus einem lokalen Verzeichnis). [Melz08]

UDDI

Ein UDDI-Verzeichnis ist eine über ein Netzwerk zugängliche Datenbank zur Speicherung von und Suche nach Servicebeschreibungen, und den dahinterstehenden Diensten. Während ein Dienstanutzer ein UDDI-Verzeichnis nutzen kann, um seine Dienste zu veröffentlichen, ist es den sonstigen interessierten Nutzern möglich, dieses nach gewünschten Diensten zu durchsuchen und deren Schnittstellenbeschreibung anschließend als WSDL-Dokument anzufordern. Die Datenstruktur von UDDI-Verzeichnissen ist standardisiert und für die Strukturierung der UDDI-Inhalte wird ebenfalls XML verwendet.

OASIS beschreibt UDDI grundlegend wie folgt: "The focus of Universal Description Discovery & Integration (UDDI) is the definition of a set of services supporting the description and discovery of (1) businesses, organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access those services. Based on a common set of industry standards, including HTTP, XML, XML Schema, and SOAP, UDDI provides an interoperable, foundational infrastructure for a Web services-based software environment for both publicly available services and services only exposed internally within an organization." [OASI04]

So einfach dieser Vorgang aus technischer Sicht klingen mag, gibt es eine Vielzahl von bisher meist nur unzureichend beantworteten Fragen zu klären. Zurzeit steckt die Entwicklung des UDDI-Konzepts noch in den Anfängen.

Dabei sind es nicht die technischen Fragen, die ernsthafte Probleme bereiten.

Die noch immer in den Anfängen steckende Entwicklung des UDDI-Konzepts liegt durchaus nicht einer mangelnden technischen Umsetzung. Ingo Melzer identifiziert vor allem ungelöste Fragen und Probleme aus primär rechtlicher und wirtschaftlicher Sicht: [Melz08]

- Wie ist die Qualität der Web Services?

- Wie wird ein Web Service nach seiner Nutzung abgerechnet?
- Wer trägt die Verantwortung für ein Web Service?

Erst nach Beantwortung dieser Fragen ist es möglich, Web Services (vor allem solche die einem Nutzer noch nicht bekannt sind) gezielt, spontan und dynamisch einzusetzen, da dem Nutzer sonst allein aufgrund der WSDL-Beschreibung nicht genügend Information zur Verfügung stünde.

Das Prinzip von UDDI kann mit der Funktionsweise von Telefonbüchern verglichen werden. Das UDDI-Verzeichnis besteht aus den vier Haupttabellen: White Pages, Yellow Pages, Green Pages und Service Type Registration. [Melz08]

Unternehmen ist es über die White Pages möglich, Informationen über sich selbst und seine Web Services in ein UDDI-Verzeichnis bekanntzugeben, welche ein Dienstanbieter wiederum durchsuchen kann. Die White Pages sind somit mit einem klassischen Telefonbuch vergleichbar.

Es ist jedoch häufig der Fall, dass einem Nutzer kein Unternehmen bekannt ist, sondern nur die Kategorie oder Branche, welches seinen gewünschten Dienst anbietet.

In den Yellow Pages kann ein Nutzer kategoriebezogen das Verzeichnis nach Web Services durchsuchen.

Über die Green Pages kann zu jedem Web Service eine Text-Beschreibung (für "menschliche" Leser) hinzugefügt werden. Wenn der Nutzer also weder Kategorie noch Anbieter kennen sollte, ist es ihm so möglich die Beschreibungen nach seinen Anforderungen zu durchsuchen.

In der Service Type Registration werden Informationen in maschinenlesbarer Form zu den Web Services gespeichert.

UDDI kann somit in zwei unterschiedliche Anwendergruppen unterteilt werden: Menschliche Benutzer, die einen gewünschten Dienst manuell Suchen, und Anwendungen. Deshalb gibt es oft zielgruppenorientierte Zugänge zu einem UDDI-Verzeichnis. Einerseits, beispielsweise, Webanwendungen für menschliche User und andererseits Schnittstellen für Anwendungen, die verschiedene Dienste bereits automatisiert auffinden wollen. [Melz08]

Das Datenmodell von UDDI wird über das UDDI-XML-Schema definiert und stellt die Datenstrukturen businessEntity, businessService, bindingTemplate, tModel, operationalInfo und publisherAssertion zur Verfügung. Weiters stellt UDDI Anwendungsschnittstellen (die sog. UDDI-API) für die Suche, Veröffentlichung und Verwaltung von Verzeichnisdaten zur Verfügung. Diese Schnittstellen sind ebenfalls durch Web Services realisiert und verwenden das SOAP-Protokoll. [Melz08]

Im Datenmodell werden Informationen der Dienstanbieter und technische Beschreibungen der Web Services erfasst, womit die standardisierte Suche, Verwaltung, Kategorisierung und Katalogisierung dieser ermöglicht wird.

Unternehmen oder Organisationen werden durch den businessEntity-Datentyp abgebildet, welcher aus folgenden Elementen besteht: [Melz08]

businessKey	Die businessEntity wird durch das businessKey Attribut eindeutig gekennzeichnet, wobei sein Wert im ganzen Verzeichnis einzigartig ist.
Name	Spezifiziert den Namen der businessEntity
discoveryURLs	Ein optionales Element, das verwendet werden kann, um die Dienste mittels URLs aufzulisten.
Description	Dient zur Beschreibung der businessEntity.
Contacts	Dient zur Beschreibung der Kontaktdaten (z.B. Adresse, Ansprechpartner, etc.) einer businessEntity.
businessServices	Besteht aus einer Liste von businessService-Datenstrukturen, die die angebotenen Dienste repräsentieren.
IdentifierBag	Dient zur Angabe weiterer Daten über das Unternehmen.
categoryBag	Wird zur Kategorisierung von businessEntities verwendet (z.B. Branche, Land, etc.). Die Kategorisierung wird mittels tModels durchgeführt.
Dsig:Signature	Kann eingesetzt werden, um eine businessEntity mit einer digitalen Signatur zu verbinden.

Tabelle 7: UDDI - businessEntity Datentyp

Über den businessService-Datentyp wird eine logische Gruppe von Web Services repräsentiert, wobei da noch keine technischen Informationen über die Dienste hinterlegt sind. Diese Struktur wird dazu verwendet, um eine Menge von Web Services in einer einheitlichen Rubrik zusammenzufügen. Jedes businessService ist dabei ein Kind-Eintrag einer businessEntity und enthält beschreibende Informationen (wie Namen, Beschreibungen und Klassifizierungen), um den Sinn der einzelnen Web Services Rubrik zu definieren. Die Bündelung von Web Services hier zu einer Rubrik kann der Abbildung eines Geschäftsmodells (eines betriebswirtschaftlichen Produkts) entsprechen. Beispielsweise könnten die Web Services "Einloggen", "Ausloggen" und "Suchen" einer privaten Suchmaschine somit logisch zu einer Rubrik zusammengefasst werden. [OASI04]

Die businessService-Datenstruktur besteht aus folgenden Feldern: [Melz08]

serviceKey	Eindeutige Bezeichnung eines businessService-Eintrags.
businessKey	Verweist das businessService auf eine businessEntity.
Name	Spezifiziert den Namen des businessService.
Description	Dient zur Beschreibung des businessService.
bindingTemplates	Dient zur technischen Beschreibungen der im businessService enthaltenen Web Services.
categoryBag	Wird zur Kategorisierung von businessServices verwendet.
Dsig:Signature	Kann eingesetzt werden, um ein businessService mit einer digitalen Signatur zu verbinden.

Tabelle 8: UDDI - businessService Datenstruktur

Der bindingTemplate- und tModel-Datentyp werden in UDDI verwendet, um technische Informationen für den Aufruf von Services abzubilden. Eine bindingTemplate-Struktur repräsentiert ein individuelles Web Service und liefert die technischen Informationen, die von Anwendungen benötigt wird, um sich mit dem Web Service erfolgreich zu verbinden und damit zu kommunizieren. Jedes bindingTemplate ist das Kindelement eines einzelnen businessService-Eintrags und muss weiters einen Zugangspunkt (z.B. eine URL) eines Web Service definieren. [OASI04]

Die bindingTemplate-Datenstruktur enthält folgende Felder: [Melz08]

bindingKey	Eindeutige Bezeichnung eines bindingTemplates-Eintrags.
serviceKey	Verweist den bindingTemplate-Eintrag auf ein businessService.
Description	Dient zur Beschreibung des bindingTemplate.
accessPoint	Spezifiziert die Adresse im Netzwerk, über die der Dienst kontaktiert werden kann. (z.B. URL oder E-Mail-Adresse)
tModelInstanceDetails	Beinhaltet eine Auflistung der im bindingTemplate enthaltenen tModels. Die Summe der tModels bildet die technische Schnittstellenbeschreibung des Web Services.
categoryBag	Wird zur Kategorisierung des bindingTemplate verwendet.
Dsig:Signature	Kann eingesetzt werden, um ein bindingTemplate mit einer digitalen Signatur zu verbinden.

Tabelle 9: UDDI - bindingTemplate Datenstruktur

Ein bindingTemplate kann wiederum mehrere tModels zur weiteren technischen Beschreibung umfassen, die eine eindeutige sowohl durch Mensch als auch Maschine lesbare technische Beschreibung des Dienstes enthalten (wie auch die eigentliche Schnittstellenbeschreibung). [Melz08]

tModels werden in UDDI verwendet, um einzigartige Strukturen zu repräsentieren und bieten einen Aufbau, der die Wiederverwendung und Standardisierung dieser ermöglicht. Dabei wird jede unterschiedliche Spezifikation, Transport, Protokoll oder Namensraum von einem eigenen tModel repräsentiert [OAS104].

Die tModel-Datenstruktur enthält folgende Felder: [Melz08]

tModelKey	Eindeutige Bezeichnung eines tModels.
Deleted	Da tModels in UDDI-Verzeichnissen nicht vollständig gelöscht werden können, dient das Feld, um diese zumindest logisch aus der Datenbank zu entfernen. Ein Löschen ist deshalb nicht möglich, da sonst verschiedenste Verweise auf das tModel nach dessen Löschung ins Leere zeigen würden. Ein als gelöscht markiertes tModel wird im Verzeichnis nicht mehr angezeigt.
Name	Spezifiziert den Namen des tModels in Form einer URI.
Description	Dient zur Beschreibung eines tModels.
overviewDoc	Dient zur optionalen Angabe von Verweisen auf externe Dokumente, um ein Web Service näher zu beschreiben. Beinhaltet auch das Element overviewURL zum Verweis auf die WSDL-Schnittstellenbeschreibung des Web Services mittels Verwendung einer URL.
identifierBag	Dient zur Angabe zusätzlicher Daten, die ein tModel eindeutig kennzeichnen.
categoryBag	Dient der technischen Beschreibung des tModels. Es kann hier auch auf technische Konzepte, beispielsweise Spezifikationen wie XML- oder SOAP-Spezifikationen, verwiesen werden. Somit müssen oft verwendete Konzepte als tModel angelegt und können anschließend einfach per Referenzierung wiederverwendet werden.
Dsig:Signature	Kann eingesetzt werden, um ein tModel mit einer digitalen Signatur zu verbinden.

Tabelle 10: UDDI - tModel Datenstruktur

Um Aufzeichnungen über Veränderungen im UDDI-Verzeichnis zu führen wird die operationallInfo-Struktur angewendet. Sobald ein Datensatz im Verzeichnis verändert wird, wird ein dazugehöriger Datensatz in einer operationallInfo-Datenstruktur erstellt. Somit wird erfasst wer, was, an welcher Stelle, zu welcher Zeit erstellt und verändert hat.

Die publischerAssertion-Struktur wird dazu verwendet, um eine Relation zwischen zwei businessEntity-Strukturen zu erstellen. Dies könnte erwünscht sein, falls verschiedene Tochterunternehmen eines Konzerns auch im Datenmodell auf den Konzern verweisen sollen. [Melz08]

Die UDDI-API wird für die Interaktion zwischen UDDI-Client und UDDI-Server eingesetzt, und verwendet SOAP als Nachrichtenprotokoll, um ebenfalls so weit wie möglich plattformunabhängig zu sein und damit eine Kommunikation mit unterschiedlichsten IT-Systemen zu ermöglichen. Die Nachrichten werden somit im XML-Format über Internetprotokolle zwischen Client und Server übertragen. Die aktuelle UDDI-API-Spezifikation bietet sowohl für Server als auch für Client unterschiedliche Funktionen. Die Server-APIs (UDDI Inquiry, UDDI, Publication, UDDI Security, UDDI Custody Transfer, UDDI Subscription und UDDI Replication) unterstützen Funktionen zur Suche, Veröffentlichung, Verwaltung, Subskription und Replikation von Daten. Die Funktionen für den Client beinhalten einen Abonnement-Dienst und Funktionen für die Prüfung von Referenzen. [Melz08]

Ein branchenspezifischer Marktplatz, ein öffentlicher Verzeichnisdienst oder eine unternehmensübergreifende Dienstkooperation wären mögliche Nutzungsszenarien für einen Verzeichnisdienst wie UDDI [Melz08]. Gerade bei einem Marktplatz wäre es für einen Konsumenten ideal, wenn alle Unternehmen mit ihren Diensten im Verzeichnis enthalten wären, da es ihm dann möglich wäre, ohne Informationsmangel den kompletten Markt einzusehen.

Ein klarer Wunsch für die Zukunft ist, dass es möglich wird, ein ausgefallenes Web Service automatisch durch ein anderes mit selbiger Funktionalität zu ersetzen, damit eine fast nahtlose Weiterverwendung einer aus Diensten komponierten Anwendung möglich sein sollte.

2.3.3 Ontologien & Semantic Webservices

Der Begriff Ontologie stammt ursprünglich aus der theoretischen Philosophie und bezeichnet dort eine eigene Disziplin, wobei heute der Ausdruck "Metaphysik" oft gleichsinnig gebraucht wird. Im Mittelpunkt der Lehre stehen die Grundstrukturen der Realität und somit der Begriff des Seins. Somit werden unter anderem Möglichkeiten und Bedingungen des Seienden und die Systematik der dabei beteiligten Entitäten (wie Gegenstände, Eigenschaften, Prozesse) und ihre strukturellen Beziehungen erfasst. ([Hess02] et al.)

Entscheidend dabei ist, dass die Wirklichkeit, wie wir sie wahrnehmen, dabei durch unsere Wahrnehmungsorgane gefiltert wird. Deshalb können wir uns keinesfalls sicher sein, dass die Welt so beschaffen ist, wie sie von uns wahrgenommen wird. Indem wir bei der Beschreibung des Wahrgenommenen sprachliche Mittel einsetzen, besteht aufgrund der Wahl unserer Begriffe, Worte und Bilder nochmals die Möglichkeit, dass wir uns vom Seienden eigentlich entfernen. [Hess02]

Daraus resultiert, dass jedes Individuum eine eigene Wahrnehmung der Wirklichkeit und darüber hinaus auch noch eine individuelle Interpretation der Sprache hat, weshalb man ebenfalls von einer unterschiedlichen Begriffsinterpretation einzelner Personen ausgehen muss.

Im Kontext der Informatik definiert Tom Gruber Ontologien als: "...a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application." [Grub07]

Das W3C definiert Ontologien in deren Überblickdokument "OWL – Web Ontology Language" folgendermaßen: "OWL is intended to be used when the information contained in documents needs to be processed by applications ... can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology." [W3C04a]

Im Umfeld der Informatik wird der Begriff Ontologie also heute verwendet, wenn "...Automaten Such-, Kommunikations- und Entscheidungsaufgaben in Bezug auf das gespeicherte Wissen übernehmen oder Daten austauschen, die selbst Information darüber enthalten, wie sie zu strukturieren und zu interpretieren sind (sog. Metadaten), so benötigen sie dazu eine Repräsentation der zugrunde liegenden Begriffe und derer Zusammenhänge." [Hess02]

Unter diesem Verständnis wird mittels einer Ontologie somit versucht, mithilfe von standardisierten Begriffen und Beziehungen eine einheitliche Ausdrucksdefinition zu bestimmen. Im Gegensatz zur Philosophie wird in der Informatik – vornehmlich aus Gründen der Abgrenzung – auch die Mehrzahl des Begriffs (also Ontologien) verwendet. Mögliche Formen der Anwendung sind die Kommunikation zwischen Softwaresystemen, automatische Schlussfolgerungen dieser, und die Darstellung und Wiederverwendung von Wissen. [Hess02]

Ontologien werden auch im Bereich des Semantic Web eingesetzt, wo versucht wird, den Inhalt von Dokumenten durch den Einsatz von Metadaten zu beschreiben, um diese für Softwareagenten interpretierbar beziehungsweise lesbar zu machen, oder die Informationen gleich so aufzubereiten, damit eine Verarbeitung durch Maschinen ermöglicht wird. Somit können beispielsweise Softwareagenten und Suchmaschinen eine benötigte Information gezielt und effizient finden und diese mit anderen verbinden, oder auf Basis einer Ontologie – aufgrund

der darin enthaltenen Metadaten und Verknüpfungsregeln – gegenseitig Aufgaben und Ergebnisse austauschen. [HKRS08] [Hess02]

Heutzutage ist das Web vor allem auf menschliche Endbenutzer ausgerichtet, was zur Folge hat, dass ein Großteil des unüberschaubaren Inhalts entsprechend menschlicher Bedürfnisse formatiert ist. Einem Menschen ist es somit in der Regel möglich, die Bedeutung von Informationen auf Webseiten zu erfassen, diese in andere Darstellungsformen zu transformieren und mit auch anderen Informationen in Verbindung zu stellen. Ohne zusätzliche Aufbereitung sind solche Darstellungsformen für Maschinen meist nicht, oder nur geringfügig verständlich. Aufgrund der riesigen Menge an Online-Inhalten wäre es heutzutage nicht mehr möglich, ohne entsprechende Suchmaschinen die gewünschten Informationen rasch zu erhalten. Diese Suchmaschinen durchsuchen den Inhalt momentan, jedoch meistens nicht nach seiner inhaltlichen Bedeutung (also keine semantische Suche), sondern versuchen die angegebenen Suchbegriffe im Text zu finden (Vergleich von Zeichenketten). Die Suche nach Zeichenketten ist allerdings oft nicht sinnvoll, da es in jeder Sprache Begriffe gibt, die mit unterschiedlicher Bedeutung versehen sind und diese nur aus dem jeweiligen Kontext hervor geht. [HKRS08]

Ziel des Semantic Web ist die Schaffung neuartiger Anwendungen und Dienste, bei denen Daten und semantische Sachverhalte von Maschinen selbst interpretiert werden können, und damit die Maschine, die einen Dienst benutzt, auch "weiß", was dieser Dienst für sie erledigt. Dabei werden Daten maschinenlesbar und –interpretierbar gemacht, indem sie mit Struktur (Syntax) und Bedeutung (Semantik) versehen werden, wodurch es Computersystemen erlaubt wird, selbständiger zu handeln und Anfragen von Benutzern besser zu interpretieren. Um dies zu realisieren werden semantische Technologien eingesetzt, welche unter anderem Wissensrepräsentationssprachen für Ontologien sowie Methoden und Werkzeuge zur Erstellung, Wartung und Anwendung von Ontologien beinhalten. [Melz08] [HKRS08]

Auch auf diesem Gebiet ist das W3C tätig und definiert drei aufeinander aufbauende standardisierte Sprachen zum Ausdruck von Beziehungen zwischen Objekten: RDF²², RDFS²³ und OWL²⁴. Diese Sprachen erlauben die Definition und Nutzung von Ontologien, um Sachverhalte aus dem jeweiligen Anwendungsgebiet maschinenverarbeitbar darzustellen. Weiters lässt sich XML einsetzen, um eine Strukturbeschreibung von Daten und Dokumenten zu erstellen und diese zu protokollieren. [Melz08] [Hess02]

Das RDF ist eine formale Sprache zur Beschreibung strukturierter Informationen beziehungsweise Ressourcen und bietet die Möglichkeit, diesen Werte und Verweise auf andere Informationen und Ressourcen zuzuweisen, wodurch die daraus entstehenden semantischen Netze als Graph interpretiert werden können. Anwendungen sind dadurch fähig, Daten im Web auszutauschen, ohne die eigentliche Bedeutung dahinter zu verlieren, womit eine Weiterverarbeitung der enthaltenen Informationen ermöglicht wird. RDF kann somit als grundlegendes Darstellungsformat für die Entwicklung des Semantic Web bezeichnet werden, wobei ein RDF-Dokument ein gerichteter Graph mit einer Menge von Knoten ist, die durch gerichtete Kanten verbunden sind; dabei werden sowohl Knoten als auch Pfeile mit eindeutigen Begriffen bezeichnet. [HKRS08] [Hess02]

²² RDF – Resource Description Framework

²³ RDFS – Resource Description Framework Schema

²⁴ OWL – Web Ontology Language

Semantic Web ermöglicht also eine maschinenlesbare formale Beschreibung von Gegenständen und Beziehungen, die jedoch meistens nur für ein bestimmtes Anwendungsgebiet oder eine beschränkte Personengruppe und somit nicht global gültig ist. Deshalb setzen verschiedene Anwendungsgebiete aufgrund abweichender Anforderungen unterschiedliche Ontologien ein. Aufgrund dessen kann ein Symbol beziehungsweise Inhaltselement nur dann fehlerfrei von Maschinen verarbeitet werden, wenn es eine global eindeutige Bedeutung trägt. [Melz08]

Daraus ergibt sich als grundlegende Frage, wie Semantik in eine maschinenlesbare Darstellung transformiert werden kann; Semantik ist nämlich nur dann übertragbar, wenn alle Teilnehmer die Information gleich interpretieren. Während Missverständnisse bei der zwischenmenschlichen Kommunikation in der Regel durch Rückfragen behoben werden können, ist es Maschinen nur eingeschränkt möglich, Inhalte auf Plausibilität zu prüfen. Menschen ist es meist aufgrund des Kontextes möglich, eine Mehrdeutigkeit des Inhalts auflösen, wogegen solch ein Interpretationsvorgang nicht ohne weiteres von technischen Empfängern einer Nachricht durchgeführt werden kann. [Melz08]

Um ein Ressource identifizierbar zu machen, muss ihr ein eindeutiges Symbol zugeordnet werden, was durch den Einsatz von URIs ermöglicht wird. Im Gegensatz zu einem URN²⁵ für eine Ressource bietet eine URL als Symbol für eine Website zusätzlich noch an, wie diese erreichbar ist. Es ist jedoch nicht entscheidbar, ob zwei verschiedene URIs dieselbe Ressource abbilden, da jeder einer Ressource eine URI zuweisen kann. Mittels OWL ist es jedoch möglich, Äquivalenzen zu bilden und dem Problem entgegenzuwirken. [Melz08]

Wie bereits erwähnt, unterliegt RDF beim W3C der Standardisierung [W3C04b] und bildet die Grundlage für das Semantic Web. Objekte und Konzepte werden durch URIs repräsentiert und können zueinander in Beziehung gesetzt werden sowie zur Erstellung von Ontologien eingesetzt werden. Mittels RDF können Eigenschaften von Ressourcen beschrieben werden, wobei das grundlegende Datenmodell aus Ressourcen, Eigenschaften und Aussagen besteht: [Melz08]

- Ressourcen sind das Basiselement in RDF, die mittels URIs repräsentiert werden, aber nicht zwangsläufig im Web erreichbar sein müssen. URIs können beliebige Konzepte und Objekte darstellen (z.B. E-Mail-Adresse, Bücher, Website).
- Eigenschaften beschreiben die Charakteristika, Relationen und Attribute von Ressourcen und können ebenfalls beschrieben werden.
- Aussagen bestehen aus den Teilen Subjekt, Prädikat und Objekt. Das Subjekt bezeichnet eine Ressource, über die eine Aussage getroffen wird. Das Prädikat repräsentiert eine bestimmte Eigenschaft, wobei das Objekt der Wert dieser Eigenschaft ist. Dieser Objektwert kann entweder eine Ressource oder ein Literal sein, wobei Literale Zeichenketten sind, die womöglich noch anhand angegebener Datentypen dargestellt werden.

Maschinen ist es jedoch allein aufgrund der Beschreibung von Sachverhalten in RDF noch nicht möglich diese zu verstehen, aber die Erstellung eindeutiger und begrenzter Vokabulare zur Beschreibung von Anwendungsgebieten ist mit RDF realisierbar. Beispielsweise könnte solch ein Vokabular in Programmen genutzt werden, um Aussagen über ein potentiell Web Service zu treffen und selbständig Kriterien (z.B. Vertragsbedingungen, Kosten, Sicherheitsbestimmungen) für dieses zu bewerten. [Melz08]

²⁵ URN – Uniform Resource Name

RDF liefert auch zwei gängige Syntaxen zur Serialisierung, Speicherung, Verarbeitung und zum Austausch von Metadaten. Einerseits definiert das W3C in seiner RDF-Syntaxspezifikation [W3C04c] die RDF/XML-Syntax und andererseits die in seiner N3-Spezifikation die Notation 3 [W3C06].

Aussagen werden in N3 als Tripel aus Subjekt, Prädikat und Objekt gebildet und anschließend mit einem Punkt beendet. N3 bietet zusätzlich noch einige Abkürzungsmöglichkeiten. Verschiedene Ausprägungen einer Eigenschaft werden durch Kommas getrennt und verschiedene Eigenschaften eines Subjekts durch Semikolons. Wie der Name schon verrät, wird das XML-Format für die RDF/XML verwendet, wodurch sich die Notation einfach mit anderen XML-Technologien (z.B. Web Services) nutzen lässt. [Melz08]

Wie bereits angesprochen, können Maschinen nicht einfach durch Verwendung von RDF die Bedeutung von Aussagen erkennen.

Für den Austausch und das Verstehen von Informationen ist immer noch ein Vokabular vonnöten, auf das sich die Kommunikationspartner einigen.

Kommunikationspartner müssen sich auf ein Vokabular einigen, um den Austausch und das Verstehen von Information zu ermöglichen, wobei kommunizierende Maschinen sich oft über einen bestimmten Wissensbereich (Domäne) austauschen, der möglichst vollständig im Vokabular definiert werden sollte. Die Objekte und Konzepte einer Domäne werden in Schemas beziehungsweise Ontologien beschrieben. Die Erweiterungen RDFS und OWL stellen ein standardisiertes Vokabular zur Definition von Schemas und Ontologien bereit. [Melz08]

Durch den Einsatz von Schemadefinitionen, wie XSD²⁶ und RDFS, lässt sich die Lesbarkeit von Dokumenten und der Nutzen von XML und RDF noch wesentlich steigern. [Hess02]

Mithilfe in RDFS definierten Begriffen ist es unter anderem erlaubt Generalisierungsbeziehungen zwischen Ressourcen auszudrücken sowie Definitions- und Wertebereiche für Prädikate festzulegen, und es ist somit möglich, die im Vokabular verwendeten Begriffe um terminologisches Wissen oder Schemawissen zu erweitern. [Melz08] [HKRS08]

In OWL werden noch weitere Konstrukte definiert, die ausdrucksstärker sind als RDFS, wobei die Spezifikation die Bedeutung von RDF und RDFS um weitere Konstrukte erweitert, was zu einer Steigerung der Ausdrucksmächtigkeit führt. Diese wird allerdings auch teilweise eingeschränkt, um Entscheidbarkeit zu erzielen. OWL bietet vor allem Elemente zur Beschreibung von Klassenhierarchien und Elemente zur Beschreibung von Eigenschaften. Im Gegensatz zu RDF und RDFS bietet OWL mehrere Möglichkeiten zur Definition von Klassen. Eine Klasse kann einerseits durch Aufzählung ihrer Ausprägungen definiert werden. Weiters wird eine exakte und mengenorientierte Definition von Klassen gewährleistet und zwei Klassen können als disjunkt zueinander definiert werden (d.h. sie haben kein gemeinsames Element). Für Eigenschaften kann mit OWL bestimmt werden, ob diese transitiv, symmetrisch oder funktional sind, wodurch es möglich wird, von bestimmten Aussagen auf andere zu schlussfolgern. [Melz08]

Gerade im Bereich von Web Services wäre es von großem Nutzen, Dienste so zu beschreiben, dass sie automatisch von Maschinen gefunden, verstanden und genutzt werden können. Es gibt bereits Ansätze dafür, um die Funktionalität von Web Services mit Sprachen des Semantic Web zu beschreiben. Ein nennenswerter Vertreter ist, unter anderem, das *Semantic Web Service*

²⁶ XSD – XML Schema Definition

Framework (SWSF), welches aus der *Semantic Web Services Ontology* (WSMO) und der *Semantic Web Services Language* (WSML) besteht. [Melz08]

Diese Entwicklungen sind natürlich auch für das Service-System interessant, da durch eine automatisierte Kompositionsverwaltung noch flexibler und dynamischer auf mögliche Veränderungen der Geschäftslogik oder auf Service-Ausfälle reagiert werden kann.

Auch mit Methoden der künstlichen Intelligenz wird versucht, die kognitiven Aufgaben, die ein Mensch bei der Verarbeitung von Informationen aus dem Web durchführt, bis zu einem gewissen Grade zu übernehmen und die im Web verfügbaren Informationen in formalere, maschinenverarbeitbare Daten zu transformieren. In näherer Zukunft ist allerdings eine zuverlässige Anwendung dieser Methoden sehr fragwürdig. Der Ansatz des Semantic Web ist momentan die vielversprechendere Herangehensweise. [HKRS08]

Somit ist das Vorhaben, dass sich eine aus verschiedenen Web Services komponierte Anwendung beim Ausfall eines komplexen Web Services – welches einen wesentlichen Bestandteil der Anwendung darstellt – über einen Verzeichnisdienst einen adäquaten Ersatz aufgrund einer semantischen Web-Service-Beschreibung sucht und automatisch bindet, vorläufig noch unrealisierbar.

2.4 Methoden zur Qualitätssicherung von Services

Mit Methoden zur Qualitätssicherung soll die Beschaffenheit von Services beurteilt werden. Ziel dabei ist es, dem Konsumenten das Ergebnis einer Leistung bereits vor dessen Konsum absehbarer zu machen und somit das potentielle Risiko eines unerwünschten Resultats zu vermindern. Weiters soll dadurch ein geregelter und kontrollierter Ablauf der Services gewährleistet werden.

Um das Service-System langfristig in Betrieb zu halten und die Nutzung dieses gewährleisten zu können, ist es notwendig, Methoden der Qualitätssicherung einzusetzen. Das Kapitel soll zeigen, welche Möglichkeiten diesbezüglich zur Verfügung stehen.

Ein Anbieter sollte bei Dienstleistungsgeschäften vorab versuchen, dem potenziellen Kunden seine Leistungsfähigkeit unter Beweis zu stellen, wobei der Nachfrager ein konkretes Leistungsergebnis erwartet. Dabei besteht ein gewisses Risiko für den Nachfragenden, da nicht sichergestellt sein kann, ob das tatsächliche Endergebnis der Dienstleistung mit den Erwartungen des Konsumenten übereinstimmt. [ScGK06]

In der Wirtschaft werden etwa seit den achtziger Jahren umfangreiche Qualitätskonzepte eingesetzt. Dabei stellt die Qualitätssicherung Methoden und Verfahren zur Verfügung, um Mängel und Fehler in den Produkten und Leistungen früh zu erkennen und zu vermeiden. Die ständige Analyse und Verbesserung der Geschäftsprozesse ist heutzutage ebenfalls ein wichtiger Bestandteil einer qualitätsorientierten Vorgehensweise. [MeSt05]

Die Kundenzufriedenheit wird vor allem entscheidend durch die Qualität einer Dienstleistung beeinflusst, weshalb sich eine inkonsistente Dienstleistungsqualität negativ auf die Kundenzufriedenheit auswirken könnte und dadurch auch der Erfolg des Dienstleistungsanbieters beeinträchtigt werden könnte. Aufgrund der Heterogenität von Dienstleistungen sowie der Kundenbeteiligung am Dienstleistungsprozess ist eine konstante Dienstleistungsqualität nur schwer zu erzielen, da die Qualität des Leistungserstellungsprozesses und der Ausführung somit vom Personal abhängt, das bei der Durchführung der Dienstleistung beteiligt ist. [PaZB88]

Weiters kommt es durch die Mitwirkung des Konsumenten am Leistungserstellungsprozess zu Unsicherheiten bezüglich des Ablaufs und der Dauer des Hergangs, wovon ebenfalls der Erfolg der Dienstleistung beeinflusst wird. Um eine gleichbleibende Qualität über alle Kunden hinweg zu gewährleisten, sollte beispielsweise das am Prozess beteiligte Personal weitreichend geschult werden oder ein standardisiertes Prozessdesign eingeführt werden. Mithilfe einer ausführlichen Dienstleistungsbeschreibung und geeigneter Kommunikationsmaßnahmen sollen die Anforderungen und Erwartungen dem Kunden gegenüber vermittelt werden, wodurch die Koproduktion des Kunden im Erstellungsprozess verbessert wird. [SiKI09]

Kundenunzufriedenheit kann somit aufgrund mangelhafter Dienstleistungsqualität entstehen, wobei Unzufriedenheit zu einer Abwanderung aktueller Kunden führt. Unzufriedene Kunden könnten wiederum negative Mundpropaganda betreiben, was zu einer Beschädigung der Reputation führen könnte. Die Neukundengewinnung ist jedoch viel kostenaufwändiger als die Bindung von bestehenden Kunden. Die Ausübung von qualitätserhöhenden Maßnahmen muss jedoch nicht unbedingt zu steigenden Kosten führen, da fehlerhafte Leistungen oft sowieso zu Folgekosten führen (z.B. Gewährleistungskosten oder nicht verrechenbare Leistungen aufgrund von Fehlern, etc.), die wiederum die Kosten einer makellosen Leistung übersteigen könnten. [Bruh08]

Parasuraman (et al.) unterscheiden bei der Diskrepanz zwischen der vom Kunden wahrgenommenen Servicequalität und der eigentlichen Servicequalität zwischen vier organisatorischen Lücken, die mit dem Design, dem Marketing und der Durchführung von Dienstleistungen verbunden sind: [ZePM02]

- 1. Lücke: Unterschied zwischen der Kundenerwartung und der Wahrnehmung des Managements von den Kundenerwartungen.
- 2. Lücke: Unterschied zwischen der Wahrnehmung des Managements von den Kundenerwartungen und der Spezifizierung der Servicequalität.
- 3. Lücke: Der Unterschied zwischen der Spezifizierung der Servicequalität und dem Service das wirklich geliefert wurde.
- 4. Lücke: Der Unterschied zwischen Service, wie es geliefert wurde und was den Konsumenten über den Service mitgeteilt wird.

Somit sollte es eine permanente Aufgabe aller am Wertschöpfungsprozess beteiligten Mitarbeiter sein, die Qualitätserwartungen der Kunden soweit wie möglich zu erfüllen. Durch Qualitätsgarantien, genaue Dienstleistungsspezifikationen und vertrauensbildende Maßnahmen kann versucht werden das Kundenvertrauen bezüglich der Qualität zu steigern und dadurch das Kaufrisiko zu reduzieren. Als Hauptaufgabe des Qualitätsmanagements kann somit die Ausrichtung der Unternehmensleistungen auf die jeweiligen Kundenanforderungen bezeichnet werden, wobei die Messung der bestehenden Dienstleistungsqualität als Grundlage eines systematischen Qualitätsmanagements betrachtet werden kann. [Bruh08]

Hiermit stellt sich die Frage: Mit welchen Instrumenten können Qualitätsmessungen sinnvoll durchgeführt werden. Beispielsweise könnten hierfür Methoden der Marktforschung – wie Interviews, Befragungen, Beobachtungen und Experimente – herangezogen werden. Als mögliche Quellen könnten unter anderem Berichte statistischer Ämter, Handelskammern und anderer Unternehmen, Publikationen wissenschaftlicher Institute, Umsatzstatistiken, Reparaturlisten sowie Reklamationen von und Schriftverkehr mit Kunden dienen. Diese könnten mitunter auch durch Webtechnologien (Diskussionsforen, Newsgroups, Online-Produkt- und -Leistungsbewertung, Online-Feedbackformulare, Chaträume, etc.) unterstützt werden.

Um das Vertrauen der Kunden in ein Unternehmen und seine Dienstleistungen zu steigern, ist das Veröffentlichen detaillierter Informationen und Bedingungen sicherlich hilfreich. Abgesehen von Dienstleistungsbeschreibungen, Kontaktinformationen und Geschäftsbedingungen wäre beispielsweise auch die Angabe diverser Unternehmenskennzahlen hilfreich.

Um seine Mitarbeiter auf bestmöglich auf die Dienstleistung vorzubereiten wäre es für die Unternehmen selbst auch förderlich, abgesehen von finanziellen Kennzahlen, auch eine Bewertung des intellektuellen Unternehmenskapitals zu führen.

Auch Bruhn identifiziert in seinem Buch *Qualitätsmanagement für Dienstleistungen* eine Vielzahl von Verfahren zur Messung der Dienstleistungsqualität. Dabei unterscheidet er zwischen kunden- und unternehmensorientierten Messungsmethoden. Bei ersteren wird eine Messung aus Sicht der Kunden vollzogen, wobei zwischen objektiven und subjektiven Messungsmethoden unterschieden werden kann. Die unternehmensorientierten Messungen werden aus Sicht der Manager oder Mitarbeiter durchgeführt. [Bruh08]

2.4.1 Kundenorientierte Messungsmethoden

Objektive Messungsmethoden

Bei objektiven kundenorientierten Messungsmethoden wird die Leistungsqualität eines Unternehmens nicht aufgrund der subjektiven Einschätzung einzelner Kunden beurteilt, sondern es wird versucht, eine breitere Sicht der Kunden einschätzen zu können. Dabei können beispielsweise folgende Verfahren eingesetzt werden: [Bruh08]

- **Expertenbeobachtungen:**
Die Beobachtung des Leistungserstellungsprozesses durch geschulte Experten soll Aufschlüsse über mögliche Mängel und das daraus resultierende Kundenverhalten liefern. [Bruh08]
- **Silent-Shopper-Verfahren:**
Bei sogenannten Silent Shoppers handelt es sich um Testkäufer, die für die im Leistungsprozess beteiligten Mitarbeiter – die darüber nicht informiert sind – eine Dienstleistungssituation zu simulieren, um daraus Rückschlüsse auf Mängel im Dienstleistungsprozess ziehen zu können. [Bruh08]
- **Dienstleistungs- und Warentests:**
Untersuchungen von Dienstleistungen durch die Stiftung Warentest geben ebenfalls einen neutralen Überblick der Dienstleistungsqualität, wobei ein Vergleich zur Konkurrenz enthalten ist, was weitere bereichernde Informationen über die Dienstleistung liefern kann. Durch das hohe Maß an menschlicher Interaktion bei Dienstleistungen wird eine objektive Analyse jedoch erschwert. [Bruh08]

Subjektive Messungsmethoden

Bei subjektiven kundenorientierten Messungsmethoden wird die Qualitätswahrnehmung von Leistungen aus Sicht einzelner Kunden betrachtet. Diese Methodengruppe wird wiederum wie folgt unterteilt: Merkmalsorientierte Ansätze, Ereignisorientierte Ansätze, Problemorientierte Ansätze. [Bruh08]

Merkmalsorientierte Messungsmethoden

Bei merkmalsorientierten Messungsverfahren wird aufgrund der Bewertung einzelner Leistungselemente die Gesamtleistung beurteilt, wobei diese Bewertungen aus Kundensicht meist mittels Kundenbefragungen durchgeführt werden. Beispiele für merkmalsorientierte Messungsmethoden sind unter anderem: [Bruh08]

- Multiattributive Verfahren,
- Dekompositionelle Verfahren,
- Willingness-to-Pay-Ansatz,
- Penalty-Reward-Faktoren-Ansatz.

Ereignisorientierte Messungsmethoden

Bei ereignisorientierten Ansätzen werden die einzelnen Prozesse der Dienstleistung aus Kundensicht bewertet. Dabei werden sogenannte Erzählmethoden (z.B. Storytelling) eingesetzt, bei denen Dienstleistungskunden gebeten werden, ohne konkrete Fragestellung ihre Eindrücke

der Dienstleistung relativ unstrukturiert zu schildern. Beispiele für ereignisorientierte Messungsmethoden sind unter anderem: [Bruh08]

- Sequenzielle Ereignismethode,
- Critical-Incident-Technik,
- Critical-Path-Analyse,
- Root-Cause-Analyse.

Problemorientierte Messungsmethoden

Bei problemorientierten Ansätzen werden mögliche qualitätsrelevante Probleme, die während des Dienstleistungsprozesses auftreten können aus Kundensicht behandelt. Beispiele für problemorientierte Messungsmethoden sind unter anderem: [Bruh08]

- Problem-Detecting-Methode,
- Frequenz-Relevanz-Analyse für Probleme,
- Beschwerdeanalyse und -messung.
- Analyse von Kunde-zu-Kunde-Kommunikation im Internet.

2.4.2 Unternehmensorientierte Messungsmethoden

Managementorientierte Messungsmethoden

Bei managementorientierten Ansätzen werden qualitätsrelevante Aspekte der Dienstleistung aus der Sicht des Managements behandelt. Beispiele für managementorientierte Messungsmethoden sind unter anderem: [Bruh08]

- Benchmarking,
- Fehlermöglichkeits- und -einflussanalyse,
- Fishbone-Ansatz,
- Statistical Process Control.

Mitarbeiterorientierte Messungsmethoden

Bei mitarbeiterorientierten Messungsmethoden wird die externe und interne Qualitätswahrnehmung einzelner Mitarbeiter erhoben. Beispiele für mitarbeiterorientierte Messungsmethoden sind unter anderem: [Bruh08]

- Externe Qualitätsmessung durch Mitarbeiterbefragungen,
- Interne Qualitätsmessungen,
- Betriebliches Vorschlagswesen,
- Poka-Yoke-Verfahren.

Somit stehen Dienstleistungsunternehmen verschiedenste Methoden zur Verfügung, um einerseits die Qualität der Dienstleistungen zu messen und zu vergleichen, und andererseits qualitätsmindernde Fehler und Problemfelder aufzudecken. [Bruh08]

Abseits davon gibt es noch eine Vielzahl anderer Methoden zur Qualitätsmessung von Dienstleistungen und zur Steigerung der Kundenzufriedenheit. Ein der berühmtesten und am häufigsten verwendeten Verfahren ist das in den achtziger Jahren von Parasuraman, Zeithaml und Berry entwickelte SERVQUAL. [PaZB88]

Dabei handelt es sich um eine empirisch abgeleitete Methode, die zur Verbesserung der Dienstleistungsqualität eingesetzt werden kann und die vom Kunden wahrgenommene Leistung der erwarteten Leistung gegenübergestellt wird. Die Messungen werden vorwiegend durch die Befragung ausgewählter Kunden durchgeführt, wobei auch deren Vorstellung einer makellosen Leistung erhoben und in die Analyse ebenfalls einbezogen wird. [PaZB88]

Durch die Kundenbewertungen kann der Abstand zwischen gewünschter Leistung und Ist-Leistung eingeschätzt werden. Außerdem ist es dem Befragten möglich die Wichtigkeit der einzelnen Fragen zu klassifizieren, wodurch sich das Unternehmen auf wesentliche Punkte konzentrieren und die Ressourcen entsprechend einsetzen kann. [PaZB88]

Die Kundenumfragen basieren diesbezüglich auf fünf Dimensionen, welche jene für Kunden wichtigsten Kriterien der Dienstleistungsqualität widerspiegeln. Dieser Ansatz wurde dann weiterentwickelt, denn die Methode wurde anfangs nur in fünf Dimensionen gegliedert. Die insgesamt zehn Dimensionen bilden die grundlegende Struktur der SERVQUAL-Methode, aufgrund der die Kunden befragt werden und Bewertung abliefern: [PaZB88]

- Tangibles – Sachvermögen: Dabei geht es um das äußere Erscheinungsbild, also um die Präsentationsform des Unternehmens. Dazu gehören beispielsweise die Ausrüstung des Personals sowie deren Kleidung und physische Anlagen.
- Reliability – Zuverlässigkeit: Hier wird gefragt, ob es dem Unternehmen gelungen ist den versprochenen Service verlässlich und genau durchzuführen, womit auch eine korrekte und verlässliche Durchführung des Dienstes gegeben war.
- Responsiveness – Reaktionsvermögen: Hiermit ist die Bereitwilligkeit gemeint, um schnell und aktiv auf Kunden zu reagieren, diesen zu helfen und die versprochene Dienstleistung so schnell wie möglich zur Verfügung zu stellen.
- Communication – Kommunikation: An dieser Stelle wird bewertet, ob der Kunde gut informiert und ob auf seine Kommentare eingegangen wird (und das in einer Sprache, die der Kunde versteht).
- Credibility – Glaubwürdigkeit: Hierbei wird gefragt, ob der Kunde das Gefühl hätte, ehrlich vom Dienstleister behandelt worden zu sein und ob dieser somit nach bester Vertrauens- und Glaubwürdigkeit gehandelt hat.
- Security – Sicherheitsgefühl: War der Kunde im Rahmen des Dienstleistungsprozesses frei von jeglicher Gefahr, Risiko und Zweifel und ist es den Angestellten gelungen Vertrauen und Zuversicht zu übermitteln.
- Competence – Kompetenz: Besitzen das Unternehmen und deren Angestellten die erforderlichen Fähigkeiten und das Wissen, die Dienstleistung durchzuführen?
- Courtesy – Höflichkeit: Wurde der Kunde stets höflich, freundlich, mit Respekt und Rücksicht behandelt? Gibt es dadurch eine Kundenfreundlichkeit?

- Understanding/Knowing the customer – Verstehen des Kunden: Haben das Unternehmen und dessen Mitarbeiter bemüht, die Bedürfnisse des Kunden zu identifizieren und auf diese einzugehen? Auch das Einfühlungsvermögen der Mitarbeiter wird in diesem Rahmen bewertet.
- Access – Zugang: Ist es möglich das Unternehmen oder seine Angestellten einfach zu kontaktieren? Erhält der Kunde eine individualisierte Aufmerksamkeit und wird aktiv und schnell auf die Kundenanfrage reagiert?

Die SERVQUAL-Methode kann somit eingesetzt werden, um die Wahrnehmungen der Kunden zu erkennen und auf deren Bedürfnisse einzugehen. Sie kann auch verwendet werden, um die Wahrnehmung der Angestellten zu identifizieren. [PaZB88]

Die SERVQUAL-Methode bezieht sich aufgrund ihres Entwurf-Datums eher auf die Qualitätsoptimierung von klassischen Dienstleistungen (Offline-Dienstleistungen). Jedoch gerade bei Online-Services (z.B. Websites, Verkauf über Webshops, etc.) existiert noch eine Fülle von anderen Merkmalen, die die Qualität maßgeblich beeinflussen, oder deren Gewichtung im Bezug auf die Kundenzufriedenheit ist im Vergleich zur traditionellen Dienstleistung komplett unterschiedlich. Durch den Einsatz von Informationstechnologien kommt es zu neuen Anforderungen auf die sich die Service-Unternehmen ebenso einstellen müssen, wobei es sich zwingend nicht nur um technische Anforderungen handeln muss. In dem Artikel *Service Quality Delivery Through Web Sites*, gehen die Autoren Zeithaml, Parasuraman und Malhotra genau auf diese Problematik ein. [ZePM02]

So ist es für qualitätsorientierte Unternehmen, die Dienste auch Online zur Verfügung stellen, ebenso unbedingt notwendig auf diesbezügliche Anforderungen einzugehen. So führt der Fokus auf Servicequalität womöglich eher zum Erfolg als eine Niedrigpreisstrategie und starke Web-Präsenz. Letztere Strategien können weitverbreitete Probleme der Servicequalität nicht aufwiegen, wenn es Kunden beispielsweise nicht möglich ist, Transaktionen durchzuführen, Kunden-E-Mails nicht rechtzeitig beantwortet werden, die gewünschten Informationen den Kunden nicht zur Verfügung stehen oder die Produkte verspätet oder gar nicht den Kunden geliefert werden. [ZePM02]

Die Forschung in diesem Bereich identifiziert unter anderem folgende Kriterien, die Konsumenten in der Beurteilung von Online-Services angeben: [ZePM02]

- Die Verfügbarkeit von Informationen über das angebotene Produkt oder die Leistung sowie die Wartung und Aktualität von den dazugehörigen Inhalten sind essentiell. Der Zugang zu den Daten muss außerdem leicht und ersichtlich sein.
- Auch die Benutzerfreundlichkeit, Bedienbarkeit und die einfache Nutzung müssen gesichert sein um, Kundenzufriedenheit zu gewährleisten. Themen wie die Funktionalität der Webseite, die Konsistenz des Designs und der Navigation sowie die Interaktionsmöglichkeiten sind entscheidend.
- Sicherheit, Vertraulichkeit und somit auch Datenschutz sind ebenfalls wesentliche Punkte, die berücksichtigt werden müssen. Der Schutz von privaten Daten, der Anonymität und vor Missbrauch müssen sichergestellt sein und dies dem Kunden vermittelt werden, da dieser sonst aus auch diesem Grund abgeneigt sein könnte, den Kauf zu tätigen.

- Auch der grafische Stil der Website hat einen großen Einfluss auf das Kundenverhalten und die -wahrnehmung.
- Wie auch bei traditionellen Dienstleistungen spielt die ordnungsgerechte Durchführung der Leistung und die Verlässlichkeit des Anbieters eine wichtige Rolle im Leistungsprozess.
- Die Aufrechterhaltung des Kundenvertrauens durch breite Kundenunterstützung ist ebenfalls unerlässlich. Dazu gehört, dass Kunden auf verschiedene Möglichkeiten (E-Mail, Chat, Formular, etc.) mit dem Support (oder anderen Ansprechpartnern) Kontakt aufnehmen können und dass dieser auch prompt reagiert. Einige Benutzer ziehen Online-Hilfen und Empfehlungen auf Textbasis dem Support vor, weshalb diese ebenfalls vorhanden sein sollten (z.B. auf Basis einer Knowledge Base).
- Ein robustes und verlässliches System ist ebenso von bedeutend, da sich lange Ladezeiten und Ausfälle ebenfalls negativ auf die Kundenzufriedenheit auswirken.
- Die Verfügbarkeit von Produkten und Leistungen, und deren Preise sollten ebenfalls Online zur Verfügung stehen.
- Kunden favorisieren meist eine personalisierte und individuelle Behandlung, weshalb eine Speicherung und Wiederverwendung von Benutzerinformationen hilfreich sein könnte.
- Auch andere Punkte, wie beispielsweise die Angabe von Gesamtkosten der Bestellung (alle Produkte/Leistungen inklusive Liefergebühren und anderen Zuschlägen), oder auch wie lange ein Unternehmen bereits in der Branche tätig ist, können für einen erfolgreichen Abschluss unverzichtbar sein. Grundsätzlich kann somit gesagt werden, dass das Verschweigen oder Zurückhalten von leistungsbezogenen Informationen negative Auswirkungen auf das Kundenverhalten und die -zufriedenheit haben könnte.

Aufgrund der immer größer werdenden Anzahl an Webinhalten, wird es notwendig auch automatisierte Qualitätskontrollen durchzuführen. Dies erfordert den Einsatz innovativer Internettechnologien, um die Präsentation und den Inhalt von Webseiten automatisch zu analysieren, und aus deren Ergebnissen mögliche Qualitätsverbesserungen zu erzielen. [MeSt05]

Unternehmen konzentrieren sich immer mehr auf ihre Kernkompetenzen, weshalb viele Aufgaben an externe Partner übergeben werden und es somit zum sogenannten Outsourcing kommt. Besonders in Bereichen der Wartung und Reparatur ist es ein Kundenziel, so viel wie möglich zu vereinfachen und Experten zu beauftragen, Entscheidungen zu treffen, welche Aktivitäten durchgeführt werden müssen.[Dorn09] Gerade bei solchen Begebenheiten ist es wichtig, dass die Rechte und Pflichten von Serviceanbieter und -nutzer klar geregelt sind. Dies kann mit sogenannten Service Level Agreements (SLAs – auf Deutsch auch Dienstgütevereinbarung genannt) geregelt werden. [Masa07]

2.4.3 Service Level Agreements

Beim Serviceerstellungsprozess ist es entscheidend, die Verpflichtungen der Beteiligten ordentlich zu planen und zu dokumentieren, wobei alle Beteiligten dabei mit einbezogen werden sollten. Die da geformten Verpflichtungen schaffen, basierend auf den Kundenanforderungen die Grundlage eines Service Level Agreement (SLA). Da sich nicht alle Verpflichtungen adäquat messen und bestimmen lassen, handelt es sich dabei gewiss um eine komplexe Aufgabe. [Masa07]

Ein SLA ist eine verhandelte Vereinbarung zwischen Service-Anbieter und –Nutzer. Dabei kann es sich um einen rechtsgültigen, bindenden formalen Vertrag zwischen Geschäftspartnern, oder aber um einen informellen Vertrag zwischen Geschäftseinheiten eines Unternehmens handeln. Innerhalb eines SLAs werden die betroffenen Services und dessen Eigenschaften (wie Prioritäten, Verantwortungen, Garantien und Gewährleistungen) beschrieben. Jede Serviceeigenschaft könnte dabei einen Wertepiegel (eine Kennzahl und einen dazugehörigen Service-Level, also ein Maß oder Messwert) für ein Service spezifizieren. Diese Wertepiegel können sich dabei sowohl auf die technische (z.B. Ausfallssicherheit von Systemen) als auch betriebswirtschaftliche (z.B. Gewährleistung von telefonischer Erreichbarkeit, Verfügbarkeit von Humanressourcen) Ebene beziehen. So könnte ein SLA beispielsweise Wertepiegel für die Reaktionszeit, Verfügbarkeit, Zahlungsfristen oder Rechnungslegung eines Services definieren. Die Wertepiegel können als Durchschnitts- und/oder als Minimal- und/oder Maximalwerte spezifiziert werden, wodurch der Nutzer vorab informiert wird, in welchem Werterahmen das Service verlaufen kann. [Dorn09]

Unternehmensinterne Verträge sind von Vorteil, da damit die Qualität von einem Service bewertet werden und mit anderen Geschäftseinheiten und Organisationen verglichen werden kann, wodurch eine Werte-Gegenüberstellung zwischen internen und externen Serviceanbietern ermöglicht wird. Außerdem lässt sich dadurch die Verantwortlichkeit transparenter zu gestalten. [Dorn09]

Eine Kennzahl "...bezeichnet eine in numerischer Weise ausgedrückte Information über einen bestimmten quantifizierbaren Tatbestand", wobei ein dazugehöriger Service-Level "...stellt den Wert einer bestimmten, in einem SLA vereinbarten Kennzahl zur Beurteilung der Qualität einer Dienstleistung dar." [Berg07]

Der Betrieb der Services wird somit mittels SLAs reguliert, wobei die Einhaltung des Service-Levels überwacht werden muss, weshalb entsprechende Überwachungsmöglichkeiten definiert und eingerichtet werden müssen. [Masa07]

Es ist auch üblich in SLAs Strafen beziehungsweise Pönale zu vereinbaren, die im Falle einer Nichtübereinstimmung des Services mit dem SLA zum Tragen kommen, wobei sich die Abmachung meist auf die Leistung bezieht, die der Kunde erhält und nicht wie sie vom Anbieter erbracht wurde. [Dorn09]

Abseits der Verpflichtungen sollten in einem SLA auch die Messungsmethoden spezifiziert werden, wobei die Wertepegel und Kennzahlen messbar und auf den Kunden zugeschnitten sein sollten. Laut Masak sollte ein ordentlicher SLA folgende Punkte abdecken und enthalten: [Masa07]

- Eine Spezifikation des zu liefernden Services.
- Die jeweiligen Kennzahlen und Wertepegel des Services.
- Die Rahmenbedingung und Pflichten des Kunden.
- Eskalationsverfahren.
- Überwachungsmöglichkeiten und Berichtswesen.
- Ausnahmen, in welchen Fällen SLAs keine Gültigkeit haben.

Des Weiteren spezifiziert Berger verschiedene typische charakteristische Merkmale von SLAs: [Berg07]

- Ein SLA ist eine formell ausgehandelte und schriftlich dokumentierte Vereinbarung.
- Diese Vereinbarung wird zwischen zwei untereinander unabhängigen Partnern getroffen.
- Ein SLA bezieht sich inhaltlich auf Dienstleistungen als einen Teilbereich der Wirtschaftsgüter.
- Ein SLA beinhaltet Verpflichtungen für den Anbieter, der bestimmte Leistungen erbringen muss, und für den Nutzer, der bestimmte Gegenleistungen erbringen muss.
- SLAs beziehen sich immer auf einen bestimmten Zeitraum.
- SLAs enthalten eine inhaltliche Beschreibung der zu erbringenden Dienstleistungen und Regeln für alle Beteiligten relevante Prozesse der Dienstleistungserbringung.
- In SLAs wird durch die Definition von Kennzahlen und Service-Levels die Qualität der zu erbringenden Dienstleistungen formuliert
- In Abhängigkeit von vereinbarten Service-Levels werden im SLA die Preise für die Dienstleistung geregelt.
- Außerdem werden im SLA Konsequenzen definiert, die im Fall einer Abweichung der vereinbarten Service-Levels zum Tragen kommen.

Berger unterscheidet in Bezug auf Kennzahlen zwischen systembezogenen und nicht systembezogenen Qualitätsdimensionen. Folgende Liste enthält mögliche Kennzahlen mit dazugehörigen Beispielen: [Berg07]

	Kennzahl	Beispiel
Systembezogene Kennzahlen	Verfügbarkeit	Von Anwendungen, Netzwerkkomponenten, einzelnen Clients, Servern
	Leistungsfähigkeit	Antwortzeitverhalten von IT-Komponenten bzw. Anwendungen
	Wiederherstellbarkeit	Wiederherstellungszeit von IT-Komponenten bzw. Anwendungen, Frequenz von Datensicherungen
	Stabilität	Anzahl an Ausfällen pro Zeitraum, mittlere Dauer zwischen 2 Ausfällen
	Sicherheit	Frequenz der Aktualisierung der Virens Scanner
	Betriebszeit	Wann das Service zur Verfügung steht.
	Standardisierungsgrad	Anzahl unterschiedlicher HW/SW-Typen
	Technologie-Innovationsgrad	Dauer bis zum Tausch von Hardware
Nicht systembezogene Kennzahlen	Erreichbarkeit	Dauer bis zur Anrufannahme
	Leistungsfähigkeit	Kundeninformationszeit, Reaktionszeit bei Problemen und Beschwerden, Anzahl der Mitarbeiter pro Anwender, Dauer für die Lösung von Problemen
	Kompetenz	Lösungsraten
	Servicezeit	Servicezeit für Dienstleistungen
	Zuverlässigkeit	Termintreue
	Kommunikationsverhalten	Frequenz von Statusmitteilungen an Nutzer

Tabelle 11: Qualitätsdimensionen - Kennzahlen

Durch den Einsatz von SLAs kann es zu einem differenzierten Modell der Ressourcen-Reservierungen kommen, weshalb Betreiber versuchen, ein ausgeglichenes Verhältnis zwischen Kundenerwartungen und einem effizienten Einsatz der verfügbaren Ressourcen zu erreichen. Zusätzlich könnte es zwischen Anbieter und Nutzer strenge Bedingungen aufgrund von SLAs und flexible Bedingungen entsprechend individuellen Kundenpräferenzen geben, wobei die flexiblen Bedingungen zur Steigerung Kundenzufriedenheit optimiert werden sollten. [Dorn09]

In Bezug auf Web Services gibt es bereits verschiedene Sprachen zur Definition von SLAs. So gibt es beispielsweise die von IBM spezifizierte WSLA²⁷, die ebenfalls auf XML basiert. Mittels WSLA ist es Anbietern und Nutzern von Web Services erlaubt SLAs und die dazugehörigen Kennzahlen und Service-Level zu erstellen und deren Einhaltung zu überwachen. [MeSt05]

²⁷ WSLA – Web Service Level Agreements

2.5 Leistungsbündel und Virtuelle Unternehmen

Einer der größten Vorteile von Servicearchitekturen ist die Möglichkeit der modularen Anwendung, was die Optionen für die Entwicklung neuer Dienstleistungen erweitert. Beispielsweise ist durch die Kombination bestehender Module die Erschaffung einer neuen Dienstleistung möglich. Durch die Standardisierung von Schnittstellen und die Gliederung von Dienstleistungen in Module wird auch die Wieder- und Weiterverwendbarkeit ermöglicht. Dadurch wird eine schnelle Reaktion auf neue Kundenbedürfnisse sichergestellt, da vorhandene und weiter benötigte Module nicht neu entwickelt werden müssen. Die Möglichkeit der Modulkombination kann auch bewusst an Kunden weitergegeben und verkauft werden, wodurch individueller auf Kundenwünsche eingegangen werden kann und der Kunde noch zu einem späteren Zeitpunkt die Möglichkeit hat, die Dienstleistung auf seine Bedürfnisse anzupassen. [BoKr06]

Durch die Bündelung einzelner Module Servicearchitektur ist es einem Unternehmen erlaubt, neue Dienstleistungen aus den verschiedenen Kombinationsmöglichkeiten zu kreieren. Die Modularisierung eignet sich besonders für Branchen, die von den beteiligten Unternehmen eine hohe Innovationsrate erfordern. Einzelne Module könnten wiederrum ausgelagert oder an Kooperationspartner weitergegeben werden. [BoKr06]

Bei der Ausgliederung einzelner Module an externe Partner findet ein Transfer der Verantwortlichkeit von einer Organisation zu einem Anbieter statt. SLAs eignen sich beispielsweise für die Verwaltung dieser neuen Vereinbarung. Einzelne Module könnten auch an Kooperationspartner weitergegeben werden oder es könnten Leistungen beziehungsweise Module anderer Organisationen eingebunden werden, um gemeinsam eine neue Dienstleistung zur Verfügung zu stellen, die man ohne Partner nicht anbieten könnte. Dies führt zu einem sogenannten virtuellen Unternehmen, bei dem die Leistungen meist gebündelt dem Kunden erbracht werden. Zur Verwaltung von Service-Bündelungen eignet sich ein gemischtes Service Level Agreement. [Dorn09]

Bündelung lässt sich allerdings auch hervorragend einsetzen, um mehrere Leistungen eines Unternehmens zu einem einzigen Paket zu schnüren und dieses gesammelt dem Kunden anzubieten. Beispielsweise werden heutzutage, unter anderem, von Telekommunikationsunternehmen oft mehrere Leistungen zusammen angeboten und beworben. Dabei inkludieren die Telekommunikationsunternehmen Festnetz- und Mobilfunk-Telefonie sowie Fernsehen und Breitbandinternet zu einem einzigen gesammelten Tarif (sog. Quadruple Play). Daraus ergeben sich sowohl für Anbieter als auch Nutzer diverse Vorteile. Der Anbieter verspricht sich daraus beispielsweise Kostensenkungen, höhere Marktanteile und mehr Umsatz, wobei dem Kunden ein geringerer Gesamtpreis und ein einzelner Ansprechpartner zugutekommen könnten. Die Idee der Bündelung von Leistungen ist nichts Neues, da beispielsweise schon ein klassisches Mittagmenü (Vor-, Haupt- und Nachspeise) in einem Restaurant nichts Anderes darstellt.

Leistungsbündel setzen sich aus einer Reihe verschiedener Absatzleistungen zusammen. Zwar können Absatzleistungen als Bündel aus Sach- und Dienstleistungen, Rechten, Eigenschaften und Funktionen interpretiert werden, was aber bei einer konsequenten Auslegung dazu führen würde, dass zwischen einem Leistungsbündel und einer Absatzleistung keine klare Differenzierung möglich wäre. Deshalb erfordern gerade Absatzobjekte mit hohem Dienstleistungsanteil eine Differenzierung zwischen einzelnen, eigenständigen Absatzleistungen und ganzen Leistungsbündeln. [RoWo06]

Leistungsbündel entstehen somit aus der Kombination mehrerer Absatzleistungen, die auch isoliert eigenständige Marktfähigkeit aufweisen. [Roth06]

Im Rahmen von Leistungsbündelung gibt es auch Strategien für die Preisbündelung. Roth und Woratschek unterscheiden in diesem Kontext zwischen drei Bündelungsstrategien: [RoWo06]

- Einzelpreisstellung: Es handelt sich streng genommen nicht um eine Bündelungsstrategie, da die kombinierten Absatzleistungen zu eigenständigen Preisen angeboten werden. Die Einzelpreisstellung dient den anderen Bündelungsstrategien lediglich als Referenzpunkt.
- Reine Bündelung: Die Konsumenten können nur das gesamte Leistungsbündel zu einem einheitlichen Preis erwerben.
- Gemischte Bündelung: Die unterschiedlichen Absatzleistungen sind auch einzeln erhältlich.

Leistungsbündel sind in der Praxis häufig in den unterschiedlichsten Branchen und den verschiedensten Ausprägungen im Einsatz, wobei damit die unterschiedlichsten Ziele und Motive verfolgt werden. Roth unterscheidet zwischen vier grundlegenden Formen der Leistungsbündelung, wobei der Individualitätsgrad für die Charakterisierung von Leistungsbündeln ausschlaggebend ist: [Roth06]

- Einfache Leistungsbündel: Die Anzahl der enthaltenen Absatzleistungen ist meist niedrig, wobei deren Kombination fixiert ist und aus standardisierten Absatzleistungen erfolgt. Die Auswahl der einzelnen Absatzleistungen für das Leistungsbündel wird dabei vom Anbieter durchgeführt.
- Individuelle Leistungsbündel: Das Leistungsbündel wird aus der Kombination maßgeschneiderter Absatzleistungen erstellt und somit auf die individuellen Bedürfnisse der Konsumenten abgestimmt, jedoch wird der Leistungsumfang der einzelnen Absatzleistungen vom Anbieter bestimmt.
- Variable Leistungsbündel: Die Konsumenten können eine spezifische Auswahl aus standardisierten Absatzleistungen treffen und somit das Bündel selbst zusammen stellen, wogegen im Vergleich zu den individuellen Leistungsbündeln nicht die einzelnen Absatzleistungen individualisiert werden, sondern ihre spezifische Zusammenstellung.
- Komplexe Leistungsbündel: Es besteht für die Kunden die Möglichkeit, individuelle Absatzleistungen variabel zu komponieren, wodurch die charakteristischen Eigenschaften individueller und variabler Leistungsbündel verbunden werden.

Durch den für den Kunden jeweilig entsprechenden Grad an Variabilität und Individualisierung können bestehende Preisbereitschaften abgeschöpft oder höhere Preisbereitschaften generiert werden. [RoWo06]

Als Grund für Leistungsbündelung können mehrere, teilweise voneinander abhängige, Ziele genannt werden. Beispielsweise sind das: [Roth06]

- Gewinn- und Umsatzsteigerung,
- Kostenreduktion,
- Wettbewerbsziele (z.B. sich durch neuartige Pakete von der Konkurrenz abzuheben und dadurch die eigene Marktstellung verbessern),
- Kundenziele (z.B. Steigerung der Kundenzufriedenheit, Preisreduktion),
- Die Preiswahrnehmung und -bewertung der Konsumenten wird beeinflusst,
- Neugewinnung von Kunden, die vorher nicht zu Kauf bereit waren.

Wie bereits erwähnt, wäre es möglich, dass mehrere Unternehmen eine Partnerschaft eingehen, um gemeinsam aus den einzelnen mit eingebrachten Kompetenzen, eine neuartige Leistung anbieten zu können, wobei es sich auch um eine reine Bündelung handeln könnte.

Unternehmen können unterschiedlich enge Beziehungen oder Austauschformen eingehen, wobei die loseste Form des Austauschs der Markt darstellt, über den Leistungen erworben werden können. Virtuelle Unternehmen nutzen informelle Beziehungen zu potentiellen Kooperationspartnern, die im Rahmen eines Auftrages in das aktive Kooperationsnetz eingebunden werden und damit eine zwischenbetriebliche Kooperation eingehen, wobei die beteiligten Unternehmen zusammenarbeiten, um gemeinsam unternehmerische Ziele zu verfolgen oder eine Leistung zu erstellen. Die daran beteiligten Unternehmen oder einzelnen Entitäten bleiben weiterhin rechtlich selbständig. Als Beispiele für andere Kooperations-Formen können, unter anderem, Joint Ventures oder Outsourcing genannt werden. [Weig07][Soel08]

Bei virtuellen Organisationsformen bestehen teilweise nicht-vertragliche als auch vertraglich geregelte Beziehungen. Dabei kann es auch zur Festlegung von Hierarchien kommen, bei denen nicht alle Partner gleichberechtigt sein müssen. Weigle unterscheidet zwischen drei verschiedenen Kooperationsrichtungen: [Weig07]

- Horizontale Kooperationen: Zusammenarbeit von Unternehmen auf selber Produktionsstufe.
- Vertikale Kooperationen: Gemeinsame Aufgabenerfüllung durch Unternehmen aufeinander folgender Produktionsstufen.
- Komplementäre Kooperationen: Zusammenarbeit von Unternehmen aus verschiedenen Branchen.

Die Ursachen beziehungsweise Motive für die Gründung einer zwischenbetrieblichen Kooperation hängen von der jeweiligen Situation ab und sind allgemein von Unternehmen zu Unternehmen verschieden, wobei das Erlangen von Vorteilen eines und eine Verbesserung der Wettbewerbsposition der kooperierenden Unternehmen als generelles Kooperationsziel betrachtet werden kann. Durch die Bereitstellung der gemeinsamen Ressourcen und eine optimale Auslastung dieser lassen sich Synergieeffekte erzielen (z.B. Kostensenkungen, Änderung von Markt- oder Mobilitätsbarrieren). Es wäre auch vorstellbar, dass erst durch die Kooperation ein Markteintritt erreicht werden kann. [Weig07]

Im Rahmen einer intensiven Literaturrecherche identifiziert Weigle mehrere allgemeine Charakteristiken von virtuellen Organisationen. Einerseits unterteilt er in konstituierende Merkmale: [Weig07]

- Eine virtuelle Organisation besteht aus mehreren rechtlich und wirtschaftlich unabhängigen Beteiligten (z.B. Individuen, Gruppen, Vereinigungen und Unternehmen, wobei es sich sowohl um Konkurrenten, als auch um Lieferanten oder Kunden handeln könnte).
- Die virtuelle Organisation tritt Dritten als eine organisatorische Einheit auf, wobei es fraglich ist, ob sie unter eigenem Namen oder unter dem Namen eines beteiligten Partners auftritt. Unter anderem, aus diesem Grund wäre die Schaffung eines geeigneten rechtlichen Rahmens durchaus sinnvoll.
- Die Beteiligten haben ein gemeinsames Ziel.
- Virtuelle Organisationen konfigurieren sich flexibel und dynamisch zum Nutzen einer Marktchance, da sie während aller Phasen des Lebenszyklus durch ständige Abstimmungs- und Anpassungsprozesse angepasst werden kann und auf äußere Veränderungen schnell reagieren können.
- Virtuelle Organisationsstrukturen haben meist nur eine sehr flache Hierarchie und sind wenig formalisiert. Oft fehlen detaillierte Verträge und Verhandlungen (aus Zeit- und Kostengründen), was zu einem spontanen und selbstorganisatorischen Ablauf führen kann.

Andererseits unterteilt er in kritische Erfolgsfaktoren, dessen Fehlen zu höheren Koordinationskosten und geringeren Spezialisierungsvorteilen führen könnte: [Weig07]

- Die einzelnen Partner der virtuellen Organisation können sich auf ihre Kernkompetenzen konzentrieren, was sich positiv auf alle Beteiligten auswirkt, da die Aufgaben dadurch meist effizienter und besser ausgeführt werden können, als von nur einem der Beteiligten alleine.
- Um den vergleichsweise hohen Koordinationsaufwand in der verteilten Leistungserstellung bewältigen zu können ist eine adäquate und intensive Nutzung von Informations- und Kommunikationstechnologien unbedingt notwendig, sodass alle Beteiligten immer rechtzeitig die Informationen erhalten, die sie zur Ausführung ihrer Aufgaben benötigen.
- Die vorhandenen Ressourcen aller beteiligten sollten soweit wie möglich allen Partnern zur Verfügung stehen, um Synergien der einzelnen Kompetenzen zu ermöglichen.
- Ohne faires Verhalten und gegenseitiges Vertrauen aller Beteiligten ist die Organisationsform nicht vorteilhaft nutzbar.

Außerdem identifiziert er noch weitere Merkmale, die in der Literatur gegensätzlich beschrieben werden: [Weig07]

- Die an der virtuellen Organisation beteiligten Partner sind voneinander geographisch getrennt. Dies könnte bei entsprechender Distanz wesentlich von Vor- oder Nachteil sein (Zeitverschiebung, Kultur, Lohnniveau, etc.).
- In einigen Definitionen ist oftmals Kurzfristigkeit als temporäres Merkmal der virtuellen Organisation beschrieben, wonach sich virtuelle Organisationen nach einer Problemlösung wieder auflösen oder neu konfigurieren. Die endgültige Beendigung einer funktionierenden Kooperation ist dabei im Falle eines erfolgreichen gemeinschaftlichen Projekts eher unrealistisch (gemeinschaftliche Kultur, Sprache, Werte, Normen und aufgebautes Vertrauen). Weigle betont stattdessen die Flexibilität von virtuellen Organisationen, welche einen raschen Auf-, Um- und auch Abbau der zwischenbetrieblichen Beziehungen ermöglicht.
- Während manche Definitionen eine lose vertragliche Bindung beschreiben, wird in anderen eine umfassende rechtliche Absicherung empfohlen. Detaillierte vertragliche Vereinbarungen sind jedoch zeit- und kostenintensiv und schränken die geforderte Flexibilität einer virtuellen Organisation ein, weshalb wohl individuell von Fall zu Fall entschieden werden muss, ob eine detaillierte rechtliche Abmachung notwendig ist oder nicht.
- In diversen Konzepten ist ein zentraler Koordinator für das virtuelle Unternehmen vorgesehen, während andere das Prinzip der Selbstorganisation betonen.

Die Nutzung virtueller Unternehmen bietet allen Beteiligten einigen Chancen, beinhaltet aber auch diverse Risiken, die nicht vernachlässigt werden dürfen.

Die im Netzwerk angeschlossenen Partner könnten beispielsweise dadurch eine verbesserte Kundenorientierung erzielen und die verfügbaren Kompetenzen erweitern. Auch die erhöhte Verfügbarkeit neuer Technologien, ein erleichterter Zutritt zu neuen Märkten, ein Zugewinn an Wissen und die Ausnutzung von Synergieeffekten könnten positive Effekte sein. Des Weiteren wird das Risiko von allen Beteiligten gemeinsam getragen und somit aufgeteilt. Klares Risiko für die Beteiligten ist die Unsicherheit auf juristischer Ebene, die möglichst minimiert werden sollte. Es ist auch möglich, dass es zu einem unfreiwilligen Wissens-Transfer kommt. Da im virtuellen Unternehmen auch Konkurrenten vertreten sind, gilt es vorher gründlich abzuwägen, ob die Vorteile der Partnerschaft die Nachteile überwiegen. Eine zu große Vernachlässigung der restlichen geschäftlichen Tätigkeit muss ebenfalls verhindert werden. [Weig07]

Für das Netzwerk sollten, unter anderem, die Chance auf eine erhöhte Reaktionsfähigkeit, mögliche Kosteneinsparungen, weiteres Potenzial aus Synergien, kürzere Projektlaufzeiten und die flexiblere Nutzung internationaler Ressourcen genannt werden. Außerdem wird durch die Kooperation womöglich eine gewisse Unternehmensgröße erreicht, ohne die einige Aktionen gar nicht möglich gewesen wären. Klare Nachteile für das Netzwerk sind die wahrscheinlich reduzierte Möglichkeiten für persönliche Treffen und synchrone Kommunikation. Außerdem könnte es durch zu hohe Komplexität zu Wartezeiten und zu redundanten Ausführungen kommen. Konflikte zwischen beteiligten Partnern sind womöglich nur schwer zu Regeln und könnten das ganze Vorhaben gefährden, wenn dadurch ein unersetzbarer Partner verloren geht. Auch ein opportunistisches und eigensinniges Verhalten einzelner Akteure könnte nachteilige Folgen haben. [Weig07]

Kunden könnten von einer solchen, für sie individuellen Leistung und einer auftragsspezifischen Konfiguration, profitieren. Die virtuelle Organisation ermöglicht für den Kunden eine schnellere Problemlösung und liefert eher bessere und angepasste Produkte. Des Weiteren kann der Kunde verschiedenste Leistungen von einem einzigen Ansprechpartner entgegen nehmen und muss nicht verschiedene Unternehmen dafür kontaktieren. Ein mögliches Risiko für den Kunden sind die aufgrund der dynamischen Bildung oft fehlenden Referenzen. [Weig07]

Die in dem virtuellen Unternehmen involvierten Mitarbeiter und Angestellten profitieren von einer abwechslungsreicheren Tätigkeit und erhalten verbesserte Qualifizierungsmöglichkeiten. Jedoch können Motivationsprobleme, Sicherheits- und Sozialdefizite mögliche Risiken darstellen. [Weig07]

Gerade in Bezug auf Web Services ist die Kooperation zwischen verschiedenen Anbietern von Vorteil. So können sie eigene Web Services mit bestehenden anderer Anbieter kombinieren und somit bündeln, um damit das eigene Angebot schnell und flexibel zu erweitern.

3. Systementwurf und Planung des Prototyps

Dieses Kapitel beschreibt die Entwurfsphase des prototypischen Systems sowie welche Überlegungen und Punkte bei der Entwicklung solch einer Plattform noch bedacht werden müssen, um solch ein System erfolgreich umsetzen und implementieren zu können. Dabei wird dargestellt, wie das System aufgebaut sein soll und wie es schlussendlich eingesetzt werden könnte.

Anhand der Thematik "Konferenz- bzw. Veranstaltungsplanung" soll gezeigt werden, wie das Service-System geplant und verwendet werden kann, um die dafür benötigten Ressourcen (z.B. Gebäude und Räume) bereit zu stellen (z.B. durch Buchungen oder Reservierungen). Zudem wird in diesem Kapitel noch dargestellt, wie im dazu Vergleich der klassische Vorgang aussieht, oder welche anderen Technologien und Umsetzungsmöglichkeiten dafür heutzutage in Frage kommen würden.

Die Forderung an solch ein System ist unter anderem sowohl für Ressourcenanbieter als auch Ressourcennachfrager einen klaren Vorteil gegenüber dem klassischen Buchungsvorgang auszuarbeiten, die Geschäftsmöglichkeiten zu erweitern und eine Geschäftsautomatisierung einzuführen. Demzufolge sollen Geschäftsprozesse, die an der Buchung von Ressourcen beteiligt sind, durch das neue System verbessert, vereinfacht und beschleunigt werden.

Durch die Systemarchitektur wird nicht nur den Anbietern eine flinke Anpassung derer Geschäftsmodelle und -logiken ermöglicht, sondern sie bietet auch Platz für neue Ideen und Modelle, mittels denen die bestehenden Angebote der Anbieter erweitert, erneuert oder weiter spezifiziert werden können.

Der Einsatz von Informationstechnologie zur Unterstützung der Geschäftslogik war bis dato auch bereits möglich, jedoch unterstützt eine SOA (die mittels Web Services umgesetzt wird) in diesem Fall unter anderem noch die Wiederverwendung von Software und die leichte Erweiterung von Funktionalitäten.

Dadurch soll das System eine gute Abstimmung von Geschäftslogik und Technologie ermöglichen, um eine Aktualisierung und Anpassung der Anwendungen an eine veränderte Geschäftslogik zu gestatten. Bereits in traditionellen Anwendungen war eine gute Abbildung einer Geschäftslogik mittels IT möglich, jedoch waren diese Lösungen oft unflexibel, besonders wenn die Geschäftslogik verändert werden musste (aufgrund neuer Marktumstände, Erweiterung der Tätigkeiten, neue Vorgehensweisen, etc.). Der Grad, in dem Geschäftsanforderungen erfüllt werden, ist deshalb oft davon abhängig, wie genau die Geschäftslogik durch die Lösungslogik ausgedrückt und automatisiert wird.

So sollen beispielsweise Logiken zum Ertragsmanagement formuliert werden können, um eine Preis- und Kapazitätssteuerung durchzuführen. Weiters könnten vorab Kontingente bestimmt werden, um Rabatte und Preisnachlässe zu definieren. Die Kontingente könnten wiederum beispielsweise aufgrund von Auslastung, Nachfrage oder Stornoquoten automatisch angepasst werden.

Außerdem wird erörtert, weshalb sich das Service-System (aber auch ähnliche Implementierungen) zur Erweiterung bzw. Neuerstellung von Geschäftsmodellen eignet, und welche betriebswirtschaftlichen Chancen sich aufgrund der Verwendung des Service-Systems zur Unterstützung bzw. Ersetzung bestehender Geschäftsprozesse ergeben.

Das System umfasst eine serviceorientierte Lösungslogik, die mittels Komposition einzelner Services umgesetzt wird. Dahinter steht ein Serviceinventar (Repository), das alle verfügbaren Services des Systems erfasst.

Heutzutage ist Online-Handel Gang und Gebe, bei dem Ressourcen aller Art, wie beim traditionellen Offline-Handel, angeboten und vertrieben werden. Da Web Services dynamisch gefunden und gebunden werden können, ist es somit auch möglich eine echte Weltressource mittels eines Web Service aufzudecken.

Eine Organisation, die eine oder mehrere dieser Ressourcen besitzt, kann diese über das Internet im Rahmen des Service-Systems anbieten. Dieses muss im Hintergrund die Verfügbarkeit der Ressource und die Zahlungsfähigkeit des Konsumenten überprüfen und den Preis dieser Ressource entsprechend den Kundenbedürfnissen kalkulieren. Bei erfolgreichem Abschluss handelt es sich um ein Dienstleistungsgeschäft, welches kollaborativ zwischen Anbieter und Konsument definiert ist.

Der "Rational Unified Process" (RUP - siehe dazu [ZuGK04]) bietet ein geläufiges Vorgehensmodell zur systematischen, strategischen und geplanten Softwareentwicklung, welches auch für ein Projekt wie dieses notwendig ist und eingesetzt wurde.

3.1 Der klassische Buchungsvorgang

In diesem Abschnitt wird der typische Vorgang einer Konferenz-Organisation betrachtet. Der übliche Organisationsvorgang kann dabei in mehrere Hauptphasen gegliedert werden:

- Die Finanzielle Planung
- Die Disposition der benötigten Ressourcen (Gebäude, Räume, Hotels)
- Die Kommunikation mit den Teilnehmern, Sponsoren und der Medien
- Die Selektion von Publikationen und dazugehörigen Rednern
- Die Planung des Qualitätsmanagements

In dieser Arbeit wird hauptsächlich der Dispositionsprozess betrachtet, der sich mit der Organisation der für die geplante Konferenz benötigten Ressourcen beschäftigt.

Den Vorgang der Ressourcenorganisation ist wiederum in mehrere Tätigkeiten unterteilt. Zuerst muss der Ressourcen-Suchende diverse Anbieter ausfindig machen, die Ressourcen zur Verfügung stellen. Die können beispielsweise online über diverse Suchdienste oder per "Gelbe Seiten" (oder andere Verzeichnisse, die Kontaktdaten von Seminar- und Kongressorganisation beinhalten) ausfindig gemacht werden.

Nach erfolgreicher Ermittlung der Anbieter, gilt es deren Angebot zu eruieren; also welche Ressourcen sie anbieten und mit welchen Eigenschaften diese versehen sind. Die Ermittlung dieser Details und Ausstattungsmerkmale findet meist telefonisch, per Mail, über Unternehmensseiten im Internet oder eventuell mittels Prospekten statt, wobei dies für jeden Anbieter einzeln durchzuführen ist.

Nachdem die dem Bedarf und Wünschen entsprechenden Objekte bzw. Räumlichkeiten gefunden wurden, muss überprüft werden, ob diese an geforderten Terminen verfügbar sind. Hier muss berücksichtigt werden, dass häufig mehrere Räume und Gebäude gebucht werden müssen, um die Konferenz halten zu können. Diese stehen jedoch in direkter Abhängigkeit

zueinander, da sie beispielsweise nicht allzu weit voneinander entfernt sein sollten und eine gewisse Infrastruktur (z.B. Hotels als Unterkunft für die Teilnehmer, öffentliche Verkehrsmittel, etc.) in der näheren Umgebung vorhanden sein sollte.

Zusätzlich müssen auch natürlich noch Preise, Zahlungskonditionen und vertragliche Bedingungen berücksichtigt werden.

Wenn alle Forderungen von den etwaig gefundenen Objekten gedeckt wurden können diese anschließend reserviert beziehungsweise gebucht werden.

Bei der Raum- und Gebäudeausstattung können beispielsweise folgende verschiedene – wie sie in geläufigen Beschreibung zu finden sind – Ausstattungsmerkmale erwähnt werden:

Ausstattungsmerkmale	
Nutzfläche / Gesamtfläche	Anzahl der Zimmer/Räume
Leinwand	Beamer
Internetzugang	WLAN
Flipchart	Fernsehgerät
DVD-Player	Mikrophon / Lautsprechersystem
Verkabelung	Sanitäranlagen
Support-Mitarbeiter vor Ort	Art der Einrichtung / Möbel (z.B. Sessel- und Tischanzahl, Fußböden, Fenstertypen und Verdunkelungsmöglichkeiten)
Catering	sonstige Einrichtungen vor Ort

Tabelle 12: Räumlichkeiten - Ausstattungsmerkmale

Folgende Angaben zu der Umgebung sind meist angeführt:

Umgebungsmerkmale	
Attraktionen in der Umgebung	Andere Objekte in der Nähe
Lage (in der Stadt)	Öffentliche Verkehrsmittel in der Nähe
Parkmöglichkeiten	Anfahrtsbeschreibung

Tabelle 13: Räumlichkeiten – Umgebungsmerkmale

Es gab natürlich auch schon vor serviceorientierten Architekturen Umsetzungen (z.B. RPCs, CORBA, andere Schnittstellenmodelle), die zur Unterstützung des Buchungsprozesses dienten. Aufgrund fehlender Eigenschaften (z.B. lose Kopplung, Prozessorientiertheit, Portabilität, späte Bindung, etc.) war jedoch die Umsetzung mit diesen Architekturen problematisch. Um Anbieter in ein etwaiges System zu integrieren, mussten entsprechend derer Systeme Schnittstellen entwickelt werden, wobei keine Plattformunabhängigkeit oder Implementierungsneutralität gewährleistet werden konnte. In vielen Fällen war auch kein ständiger Zugriff auf die verteilten Funktionen möglich, weshalb Daten oft in gewissen Zeitabständen synchronisiert wurden. Dadurch war der Datenbestand jedoch nicht immer aktuell, weshalb bei der Buchung keine Garantie abgegeben werden konnte.

3.2 Vorteile aufgrund des Service-Systems

Wie bereits erwähnt, das Service-System kann für alle daran Beteiligten Vorteile gegenüber dem klassischen Buchungsvorgang bieten. Einerseits ist sowohl für Ressourcensuchende, als auch für Anbieter eine Aufwandsenkung möglich. Andererseits bietet das System Platz für neue Berufs- und Geschäftsmöglichkeiten.

3.2.1 Kundenvorteile

Je höher die Anbieterbeteiligung einer entsprechenden Kategorie ist, umso weiter wächst der Marktplatz, auf den die Kunden Zugriff haben. Ein globalisierter Markt eröffnet den Suchenden eine höhere Auswahl und der Marktzugang wird erleichtert. Weiters profitiert der Kunde von einer standardisierten Abwicklung und direkten Preisverhandlungen. Durch standardisierte Kataloge kann der Informationsgehalt für die Kunden erhöht werden.

Durch einen vergrößerten Markt kann der Informationsstand der Konsumenten gesteigert werden, wodurch die Ausnutzung und Bildung lokaler Monopolstellung nicht mehr so leicht möglich ist. Anbieter unterliegen in Folge einem größeren Preiskampf, weshalb sie sich durch gewisse Qualitätsmerkmale von der Konkurrenz abheben müssen, was wiederum dem Kunden langfristig zugutekommt, da dadurch eine schnellere Produkt- und Leistungsentwicklung stattfinden könnte.

Die Prozesse werden somit für die Kunden transparenter. Informationen, die dem Kunden früher nicht ohne weiteres bekannt waren, sind dadurch direkt zugänglich.

Durch das Service-System können Kunden Ressourcen verschiedener Anbieter kombinieren, um ein gewünschtes Portfolio zu erhalten. Das Service-System könnte dabei als virtuelle Organisation auftreten, bei der sich die beteiligten rechtlich unabhängigen Anbieter virtuell für den Zeitraum der Leistungsausübung zu einem gemeinsamen Geschäftsverbund zusammenschließen. Die virtuelle Organisation tritt jedoch gegenüber dem Kunden wie ein einheitliches Unternehmen auf.

3.2.2 Vorteile für Anbieter

Auch die Anbieter profitieren von dem globalisierten Markt. Es werden dadurch für den Anbieter Kunden erreichbar, was sonst nur mit vermehrtem Aufwand realisierbar gewesen ist. Dadurch ergibt sich für Anbieter eine Ersparnis bei Werbekosten, da deren Angebote direkt im System gefunden werden. Angebote werden somit leichter von potenzieller Kundschaft gefunden.

Aufgrund einer Vereinfachung komplexer Prozesse und eine teilweise Geschäftsautomatisierung ist beispielsweise auch eine Einsparung bei Betriebs- und Personalkosten möglich.

Für Anbieter soll der Zugang zum Service-System möglichst einfach gewährleistet werden können. So soll es auch Ressourcenanbietern, die keine eigenen Services zu ihrem Angebot haben, möglich sein, vom System zu profitieren. Einerseits wäre es möglich, den Ressourcen-Datenbestand des Anbieters direkt im Service-System zu führen. Andererseits kann über standardisierte Services auf den Datenbestand des Anbieters direkt zugegriffen werden. In beiden Fällen besteht Einsparungspotenzial bei der IT-Infrastruktur des Anbieters. Er muss nicht selbst eine komplexe Infrastruktur aufbauen und warten, sondern kann das Service-System für die Umsetzung seiner Geschäftslogik verwenden.

Aufgrund einer möglichst weitreichenden Geschäftsautomatisierung im Service-System sollen Tätigkeiten und Entscheidungen, die früher manuell von Menschen durchgeführt werden mussten nun soweit wie möglich automatisiert ablaufen. Dies ermöglicht einen beschleunigten Abwicklungsprozess, wodurch auf Kundenwünsche schneller reagiert werden kann.

Das System bietet eine gute inhärente Interoperabilität, was die Zusammenarbeit unabhängiger und heterogener Systeme ermöglicht. Somit können Informationen effizient verarbeitet und dem Benutzer zur Verfügung gestellt werden, wobei die einzelnen Services nicht gesondert aneinander angepasst werden müssen. Das System vereinfacht und fördert dadurch auch die Zusammenarbeit verschiedener Anbieter und liefert eine gute Basis zum Austausch von Benachrichtigungen.

Durch diese Systemarchitektur können Organisationen agiler auf Veränderungen eingehen und Geschäft und Technologie gut aufeinander abstimmen. Aufgrund der vereinfachten Wiederverwendung und der Einsatz von externen Services ergibt sich für die einzelne Organisation (den einzelnen Anbieter) meist eine geringere IT-Belastung.

Abhängig von der Betriebsart des Service-Systems sind für Anbieter noch weitere Vorteile etablierbar. Das Service-System kann als Online-Marktplatz in mancher Hinsicht beispielsweise mit dem *Apple Appstore*²⁸ verglichen werden. Dieser stellt ebenfalls einen zentralen Marktplatz dar, wo Software-Entwickler ihre Applikationen (z.B. für mobile Geräte wie das *Apple iPhone*²⁹) über diesen anbieten und vertreiben können. Daraus ergeben sich für die Anbieter folgende Vorteile:

- Der Preis der Software kann selbst gewählt werden.
- Der Entwickler erhält eine monatliche Abrechnung und Überweisung der verkauften Software. Er muss sich somit selbst weder um den Verkauf, den Zahlungsvorgang, noch um die Abrechnung kümmern.
- Es entfallen keine Kosten für unterschiedliche Zahlungsmethoden (wie z.B. Kreditkarten).
- Es gibt keine monatliche Gebühr für die Verwendung des Marktplatzes.
- Es werden auch keine Marketing-Gebühren eingehoben.
- 30% des Verkaufspreises werden jedoch als Gebühr von Apple eingehoben.

Bei der Buchung von Konferenzen kann es öfters der Fall sein, dass mehr als eine Ressource benötigt wird, um die Bedürfnisse abzudecken. So passiert es regelmäßig, dass ein Anbieter alleine nicht alle Ressourcen zur Verfügung stellen kann, die für die jeweilige Veranstaltung benötigt werden. Es herrscht jedoch die Tendenz, dass Konsumenten versuchen, Ressourcen nur von einem einzigen Anbieter in Anspruch zu nehmen, damit der Aufwand zur verringert wird. Einerseits ist der direkte Preis-, Ausstattungs- und Qualitätsvergleich nur erschwert möglich, sowie ist die Kontaktaufnahme mit vielen anderen Anbieter bzw. überhaupt deren Ausforschung und Einsicht in deren Angebot mit erheblichem Aufwand verbunden. Andererseits ist der Wunsch nach einem einzigen Ansprech- und Vertragspartner vorderrangig. So geben Konsumenten unter Umständen mehr aus, um die Ressourcen von einem alleinigen Anbieter zu erlangen und einen einzigen Vertragspartner zu haben.

²⁸ Apple App Store – Software zum Download von Applikationen. [Appl10]

²⁹ Apple iPhone – Smartphone. [Appl10a]

Diese Tendenz könnte sich wandeln, wenn die Suche und Buchung über mehrere Anbieter hinweg größtenteils automatisiert durchgeführt werden kann und sich der Aufwand des Ressourcensuchenden dadurch massiv verringert. Beziehungsweise können mehrere Anbieter eines Portfolios als virtuelles Unternehmen auftreten und ebenfalls gegenüber dem Kunden als einziger Vertragspartner auftreten. Dadurch könnten auch kleinere Anbieter ihre Umsätze steigern, da sie öfters berücksichtigt werden.

Auch auf Entwicklungsebene lassen sich einige Vorteile bestimmen. Beispielsweise können User Interfaces neu gestaltet werden, ohne die dahinter liegende Logik und Services dabei verändern und anpassen zu müssen. Da es sich um ein Online-System handelt, ist bei einer Änderung keine neue Kompilierung der Anwendung und keine Softwareverteilung an die Nutzer notwendig. Auf Wunsch kann jedoch ein Software-Client für die Verwendung von Kompositionen und Services entwickelt werden, denn man ist nicht zwangsweise an Webanwendungen gebunden. Aufgrund der Plattformunabhängigkeit von Web Services kann auch beliebig Programmiersprache und Betriebssystem gewechselt werden.

3.3 Glossar

Folgendes Glossar soll dazu dienen, um häufig verwendete Begriffe im Kontext des Service-Systems eindeutig zu beschreiben.

User / Benutzer	Ein Anwender des Systems, der dieses verwendet um Ressourcen von Anbietern zu buchen, um diese anschließend entsprechend der Buchung zu gebrauchen.
Anbieter	Eine Organisation, ein Unternehmen oder eine Person, die Dienste oder Ressourcen zur Verfügung stellt und an das System angebunden hat.
Systemverwaltung	Die Rolle im System, die mit dem Betrieb, der Kontrolle und der Wartung des Systems vertraut ist.
Service / Dienst	Im Kontext des Systems ist ein Service bzw. Dienst ein Softwareartefakt, also ein technischer Anwendungsprozess meistens in Form eines Web Services, der mit gewissen Eingabewerten zu einem Ergebnis führt. Es ist über ein Netzwerk verfügbar, wurde mittels eines Softwaresystems implementiert und kann diverse Operationen ausführen, wodurch eine Serviceanfrage bedient wird.
Buchung	Eine vom Benutzer durchgeführte Bestellung von einer oder mehreren Ressourcen eines oder unterschiedlicher Anbieter. Die Ressourcen stehen dem Benutzer für den in der Buchung festgelegten Zeitraum zur Verfügung. Es resultiert somit in einem Vertrag mit dahinterliegenden Konditionen.
Reservierung	Eine vom Benutzer durchgeführte Aktion, um eine zukünftig beabsichtigte Buchung in Aussicht zu stellen, wobei es nur zur Festlegung von Ressourcen und den benötigten Zeitraum derer kommt, jedoch nicht zu einer fixierten Bestellung. Diese Funktion ist sinnvoll, wenn beispielsweise der Bedarf oder der benötigte Zeitraum noch nicht definitiv geklärt sind.
Ressource	Ein Gut oder eine Leistung jeglicher Art, die von einem Anbieter zur Verfügung gestellt wird und einer Kategorie zugeordnet werden kann.
Kategorie	Dient zur Aufgliederung der Ressourcen in verschiedene Bereiche bzw. Branchen und spezifiziert, mit welchen Eigenschaften Ressourcen einer Kategorie beschrieben werden müssen.
Service-System / Plattform	Das System bzw. die Plattform umfasst alle Funktionalitäten, Dienste und Datenstrukturen, die darin eingegliedert sind.

Tabelle 14: Service-System - Glossar

3.4 Entwurfdetails

3.4.1 Grundlagen

Das Service-System speichert und verwaltet Anbieter- und Providerdaten, die in einem Verzeichnis (Datenbank) eingetragen werden. Zusätzlich werden zu den Anbietern Services und dessen Funktionen hinterlegt, über die auf Ressourcendaten der jeweiligen Anbieter zugegriffen werden kann. Darüber hinaus können unabhängige Provider auch weitere Services eintragen, die unterschiedlichste Funktionen bieten können, um den Nutzen und die Möglichkeiten zu erweitern.

Diese können mehrfach und an verschiedenen Orten eingesetzt werden, und mit anderen Services zu Kompositionen verbunden werden, um eine gemeinsame Aufgabe zu verwirklichen.

Web Services bieten sich für das System deshalb an, weil sie eine sprachenunabhängige und namhafte Infrastruktur für die Integration heterogener Komponenten zur Verfügung stellen. Durch ihre neutralen XML-basierten Standards gelingt es, verschiedene Komponenten zu einem einzigen und kohärenten System zu verbinden, egal ob die Komponenten in unterschiedlichen Programmiersprachen entwickelt wurden oder auf verschiedenen Plattformen laufen.

Durch die lose Kopplung von Web Services kann das System flexibel und dynamisch seine gewünschte Funktionalität anpassen und erweitern. Dadurch wird ermöglicht, dass Modelle durch die zusätzliche Einbindung oder durch den Austausch von Services verändert werden können. Daraus ergeben sich sowohl aus technischer als auch aus betriebswirtschaftlicher Sicht Vorteile. Eine Komposition von Services kann beispielsweise angepasst werden, um ein ausgefallenes Service zu ersetzen oder ein Geschäftsmodell zu erweitern. Durch eine lose Kopplung kann eine geringe Abhängigkeit zwischen Komponenten gewährleistet werden. So kann für eine bestimmte Aufgabe immer wieder dasselbe Service verwendet werden; es könnte aber auch durch ein anderes ersetzt werden. Wenn allerdings ein Service ausfällt, welches auf den Datenbestand eines bestimmten Anbieters zugreift, ist es eher unwahrscheinlich einen Ersatz dafür zu finden (wenn das Service nicht redundant abgebildet ist).

Ein Service kann durch eine oder mehrere Servicebeschreibungen charakterisiert werden, welche in einer Service-Registry (dem Verzeichnis oder Repository) gespeichert werden. Die unterschiedlichen Servicebeschreibungen können unterschiedliche Details über das Service liefern. Dies ist hilfreich, wenn der Serviceanbieter den unterschiedlichen Nutzern verschiedene Einblicke über ein Service liefern möchte. Dadurch wird ermöglicht, Personen unterschiedlichster Befähigung Informationen zu den Services zu vermitteln.

Um eine bestimmte Geschäftslogik umzusetzen werden Services zu einer Komposition verbunden, damit ein gewünschtes Resultat zu erzielt wird. Solch eine Komposition kann sowohl vorab zusammengestellt als auch bei der Laufzeit erzielt werden. Auch wenn die Servicekomposition im Vorhinein entworfen wurde, wird die Selektion der einzelnen Services und deren konsequente Bindung zur Komposition entweder bei der Entwicklung, bei dessen Einsatz oder dynamisch zur Laufzeit durchgeführt. Im Fall der Konferenzorganisation besteht eine geläufige Komposition voraussichtlich zumindest aus Services zur Suche und Buchung von Räumlichkeiten.

Ressourcenanbietern soll es möglich sein bestehende Dienste in das Service-System zu integrieren oder neue Dienste dafür zu entwickeln, wobei im Prototyp nur Letzteres berücksichtigt wird. Über diese Dienste wird einerseits der Datenbestand der Ressourcen der

Anbieter zugänglich, wodurch es den Service-System-Benutzern erlaubt ist deren Ressourcen zu durchsuchen, zu reservieren und zu buchen. Andererseits können zusätzliche Dienste diese Funktionalitäten erweitern, um den Benutzern weitere und detailreichere Informationen und Funktionen zu den Ressourcen zu liefern.

Web Services stellen die Bindeglieder zwischen den Anbieterdaten und dem Service-System dar. Autorisierungskomponenten überprüfen dabei, ob der Zugriff und die Verwendung der jeweiligen Web Services gestattet ist. Sie können in unterschiedlichsten Kompositionen eingesetzt werden und sind in vielen Fällen wieder verwendbar, da sie eine autonome Funktion darstellen, die für zukünftige Kompositionen noch immer wertvoll sein könnte.

Über grafische Oberflächen (z.B. sogenannte Web-Forms) tätigen die Benutzer Eingaben, welche an die jeweiligen Services der Komposition weiter gegeben werden und somit Anfragen auf die Ressourcendaten der beteiligten Anbieter gestellt werden. Aufgrund der Anfrage liefern die Web Services entsprechende Ergebnisse zurück, welche wiederum innerhalb einer grafischen Oberfläche dargestellt werden.

In einer Komposition muss also auch festgelegt werden, welche grafischen Oberflächen verwendet werden, um Eingaben und Ausgaben zu ermöglichen. Die graphischen Interfaces können somit genauso flexibel gewechselt werden, wie die Services, die sie miteinander verbinden. Kompositionen benötigen auch einen Koordinator (Controller), der die Services entsprechend der gewünschten Logik aufruft, überwacht, ihnen die entsprechenden Parameter übergibt und somit für die Kompositionslogik verantwortlich ist und dadurch die Logik des Geschäftsprozesses umsetzt. Diese Rolle wird ebenfalls vom Service-System übernommen. Die in einem Datenmodell erfasste und vom Koordinator durchgeführte und überwachte Komposition stellt somit das Bindeglied zwischen einem Geschäftsmodell und der SOA dar.

Der Kompositions-Koordinator entscheidet aufgrund der dahinter liegenden Logik während der Ausführung, welche Servicefunktion wann und mit welchen Parametern aufgerufen wird. Dabei werden beispielsweise die Rückgabewerte eines Services überprüft und entsprechend auf den Wert dieser die folgenden Aktionen gesetzt.

Hier ist gutes Scheduling wichtig, um einen effizienten Ablaufplan zu erstellen und begrenzte Ressourcen gut zu verteilen. So sollten möglichst viele Prozesse in möglichst kurzer Zeit verarbeitet werden, die Ressourcen möglichst weit auslasten, um sie für künftige Anfragen frei zu bekommen.

Für die Erstellung eines Services benötigt man die Prozesse und dessen Subfunktionen, die damit umgesetzt werden sollen. Außerdem müssen der Informationsfluss, die benötigten Daten und andere Ressourcen, die den Prozess beeinflussen, bekannt sein. Zum Zeitpunkt des Entwurfs eines Services ist es meist nicht möglich zukünftige Entwicklungen in dessen Umfeld einzuplanen oder eine mögliche Anpassungsnotwendigkeit vorauszusehen. Man kann sich jedoch aufgrund bestimmter Vorkehrung gut darauf vorbereiten, um Dienständerungen mit möglichst geringem Aufwand durchführen zu können.

Bei der Gestaltung der Services ist auch auf die Wahl der passenden Granularität zu achten. Wenn Services zu umfangreich gestaltet werden, dann sinkt die Möglichkeit der Wiederverwendung. Bei zu simpler Gestaltung steigt jedoch automatisch der Verwaltungsaufwand. Deshalb ist bereits in der Planungsphase der Geschäftslogik auf technischer Ebene eine sinnvolle Unterteilung der Prozessschritte notwendig.

Betrachtungsgegenstand innerhalb des Prototyps sind erstrangig Ressourcen zur Organisation von Konferenzen, Symposien, Tagungen oder Veranstaltungen. Ziel ist es jedoch auch das System so zu entwickeln, sodass ohne großen Aufwand auch möglichst jede andere Ressourcenkategorie generisch eingesetzt und erweitert werden kann.

Jede Branche hat allerdings generalisierte und allgemein anerkannte Vorgehensweisen, Praktiken und Ziele, die im System berücksichtigbar sein sollen und mit möglichst wenig Aufwand integriert werden können. Um spezielle Forderungen zu erfüllen, ist ein Entwurfsprinzip erforderlich, welches Richtlinien enthält, um die Lösungslogik auf eine bestimmte Art und Weise im Hinblick auf bestimmte Ziele zu gestalten.

Zur weiteren Veranschaulichung des Systemkonzepts soll folgende Graphik dienen:

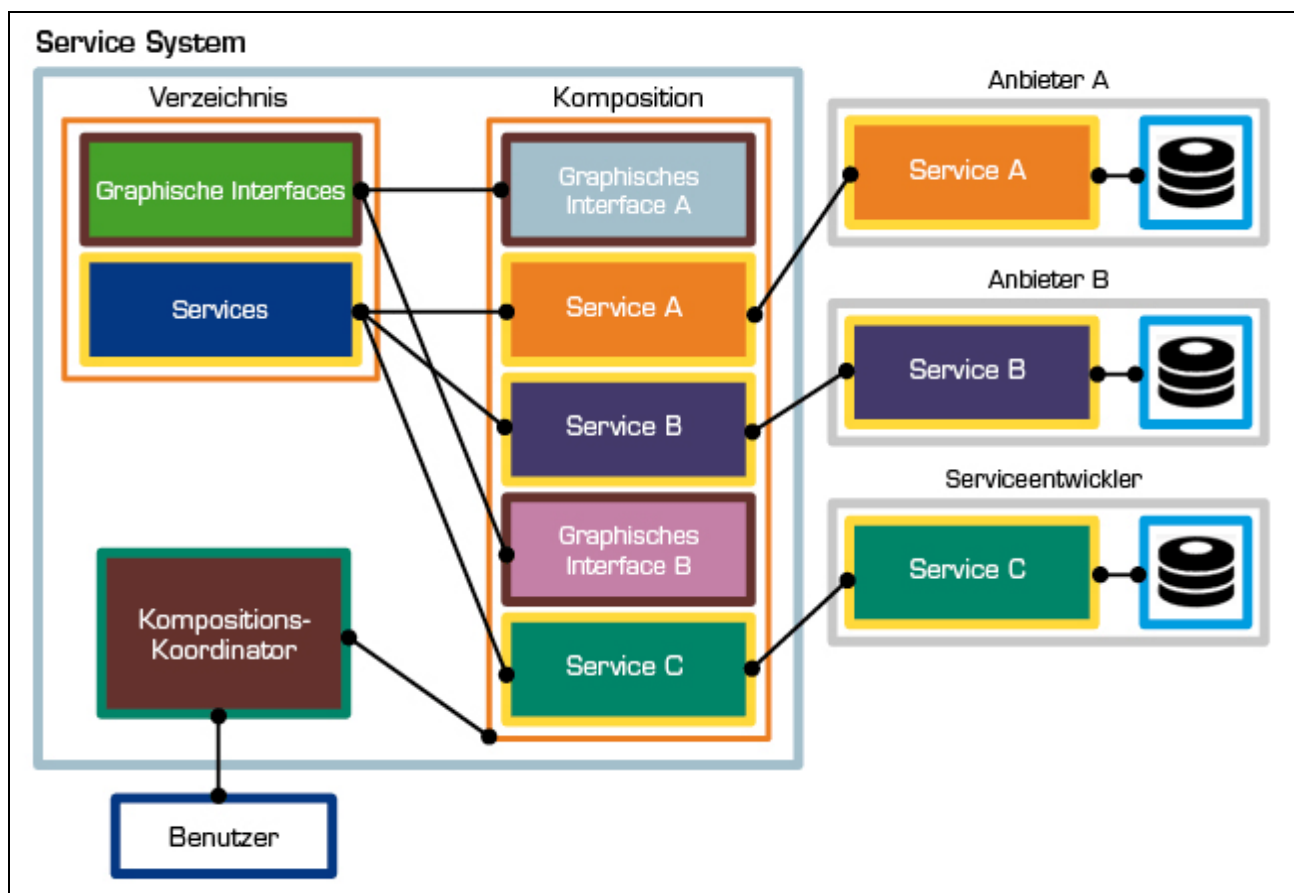


Abbildung 2: Systemkonzept

3.4.2 Anforderungen an das System

Zusammengefasst soll der Prototyp des Service-Systems folgende Anforderungen erfüllen und umsetzen:

- Anbieter können ihre Infrastruktur um standardisierte Dienste erweitern, um ihre Ressourcen im System verfügbar zu machen.
- Es können Dienste entworfen werden und diese ins System integriert werden. Diese Dienste sollen dazu dienen eigene Ressourcen am Markt anzubieten oder sonstige Funktionalitäten für Kompositionen anzubieten.
- Graphische Oberflächen können entworfen werden, um den Benutzern eine Schnittstelle zu den Web Services zu bieten

- Services und graphische Oberflächen können zu Kompositionen zusammengefasst werden. Somit soll die Integration verschiedener Geschäftslogiken ermöglicht werden.
- Mittels einer exemplarischen Komposition kann auf Ressourcen verschiedener Beispiel-Anbieter zugegriffen werden. Kompositionen können über das Service-System also ausgeführt werden und müssen von diesem koordiniert werden.
- Das Service-System bietet Web Services, mit denen nach Diensten und Kompositionen gesucht werden kann.

3.4.3 Service-Beschreibungen

Allgemeines

Um Services dynamisch, spezifisch und flexibel einsetzen zu können, muss deren Auffindbarkeit gewährleistet sein. Nach Services kann im Service-System-Verzeichnis (Repository) gesucht werden. Die Schwierigkeit dabei ist jedoch automatisch zu entscheiden, ob ein gewisses Service für die benötigte Funktionalität verwendbar ist oder nicht. Das heißt es müssen Servicedetails in maschinenlesbarer Form vorliegen, weshalb es notwendig ist, Methoden zur Servicebeschreibung zu verwenden und zu entwickeln.

Um eine möglichst gute Auffindbarkeit gewährleisten und eine korrekte Interpretierbarkeit bieten zu können, müssen Metainformationen zur Servicequalität bei einem Service hinterlegt werden, wobei sich aus diesen Informationen entsprechend Details von Funktionalität und Servicequalität auslesen lassen müssen. Unter anderem empfiehlt sich deshalb die Einführung eines standardisierten Vokabulars, um eine effiziente Stichwortsuche etablieren zu können.

Neben der WSDL- und XML-Schema-Definitionen können zusätzlich für Menschen lesbare Kommentare angehängt werden. Weiters können optionale oder ignorierbare Policy-Zusicherungen hinterlegt werden, um Vorgaben und Kompatibilitätsoptionen zu präsentieren. [Erl08]

Grundsätzlich sollte man vorab versuchen, dem potenziellen Service-Nutzer so viele Informationen und Hinweise (z.B. Metadaten über Funktionalität und Servicequalität) bezüglich der zugrunde liegenden Logik, des Verhaltens und Beschränkungen des Services zu liefern.

So sollten bei einem Service sowohl funktionale Metainformation in menschen- und maschinenlesbarer Form, sowie auch Metainformationen über die Servicequalität in technischer (SLAs) und nichttechnischer Form hinterlegt sein.

Serviceprofile

Alle Informationen, die zu einem Service abgelegt werden, können als Serviceprofil zusammengefasst werden. Es handelt sich dabei also um eine detaillierte Beschreibung des Services aus verschiedenen Blickwinkeln, zu welcher noch mit SLAs und Links zu WSDL und anderen Definitionen hinterlegt werden können. Folgende Informationen sollten dabei hinterlegt sein: [Erl08]

Serviceprofile	
Service name	Name des Services.
Zweck	Sowohl eine kurze Beschreibung des Services und sein Zweck in einem Satz, als auch eine ausführliche Darstellung und Erklärung von Kontext und Funktionsgrenzen des Service mit so vielen Details wie notwendig.
Anforderungen an die Servicequalität	Dieses Feld nimmt die voraussichtlichen Anforderungen an die Servicequalität sowie Merkmale und Beschränkungen auf, die den Service als Ganzes betreffen. Dazu gehören Sicherheit, Performance, Verfügbarkeit und Geschäftsanforderungen.
Fähigkeiten	Sowohl Fähigkeiten die vorhanden, in Entwicklung oder geplant sind sollten dokumentiert werden. (Informationen dazu werden noch gesondert dokumentiert)
Schlüsselwörter	ein oder mehrere Schlüsselwörter, die einem offiziellen Vokabular entstammen.
Kontakt Daten	Des Inhabers, Zuständiger, Autoren und Personen, die bezüglich der Dokumentation erreicht werden können.
Version	Versionsnummer
Status	Der Entwicklungsstatus des Services mit Standardbegriffen ausgedrückt, die für die Entwicklungsphasen des Projekts stehen, wie z.B. Analyse, Vertragsentwurf, Entwicklung oder Produktion. Voraussichtlicher Erstellungstermin, wenn nicht in Produktion.
Weitere Details	Geschäfts- als auch Technologieaspekte in für Menschen lesbarer Form.

Tabelle 15: Serviceprofile - Datenstruktur

Metadaten von Servicefähigkeiten:

Servicefähigkeiten	
Operationsname	Name der Operation, also der Fähigkeit.
Zweck	Knappe Beschreibung des Zwecks und funktionalen Kontextes der Fähigkeit.
Logikbeschreibung	Eine schrittweise Beschreibung der Logik, die diese Fähigkeit ausführt. Diese kann durch Algorithmen, sog. Workflow-Diagramme oder sogar ganze Geschäftsprozessdefinitionen ergänzt werden - je nachdem, welches Stadium die Fähigkeitsdefinition erreicht hat.
Ein/Ausgabe	Diese beiden Felder definieren die zulässigen Eingabe- und/oder Ausgabewerte einer Fähigkeit und die damit verbundenen Beschränkungen.
Schlüsselwörter	ein oder mehrere Schlüsselwörter, die einem offiziellen Vokabular entstammen.
Kontaktdaten	Des Inhabers, Zuständiger, Autoren und Personen, die bezüglich der Dokumentation erreicht werden können.
Version	Versionsnummer
Status	Der Entwicklungsstatus der Operation.
Qualität	Anforderungen an die Servicequalität.

Tabelle 16: Servicefähigkeiten - Datenstruktur

Wichtig ist, dass Informationen zu Services nach einem standardisierten Muster erfasst werden, um sicherzustellen, dass dieselben Arten von Daten erfasst werden. Womöglich auch in unterschiedlichen "Schreibstilen", sodass Personen unterschiedlicher Qualifikation sich ein Bild davon machen können (z.B. Techniker).

3.5 Anwendungsfälle

Der geplante Prototyp kann in mehrere Anwendungsfälle gegliedert werden.

3.5.1 Überblick

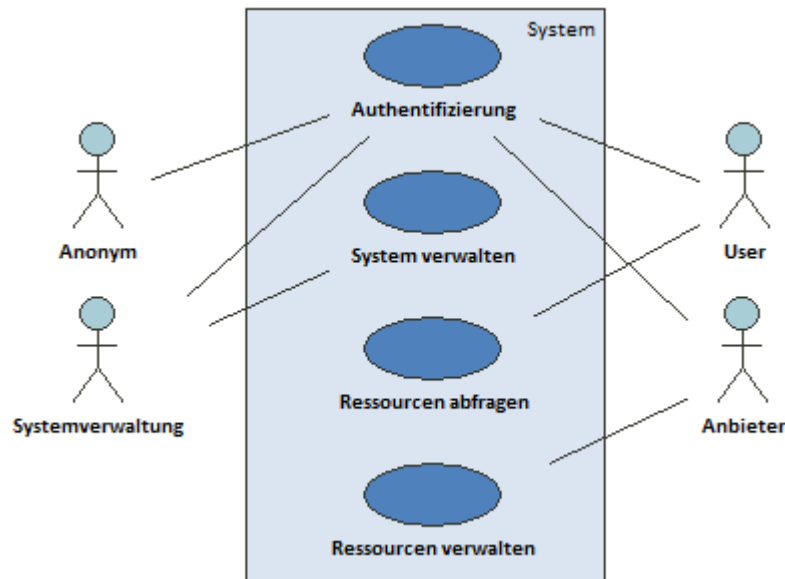


Abbildung 3: Anwendungsfälle - Überblick

3.5.2 Authentifizierung

Um im System Aktionen durchführen zu können ist eine Authentifizierung notwendig. Nach dem Login nimmt der Benutzer - ein noch anonymer Benutzer - eine Rolle im System ein.

Dies können sein:

- User: Ein Standard-Benutzer, der das System verwenden will, um Ressourcen zu buchen.
- Anbieter: Ein Benutzer, der Ressourcen zur Verfügung stellt.
- Systemverwaltung: Ein Benutzer, der das System verwalten kann und darf.

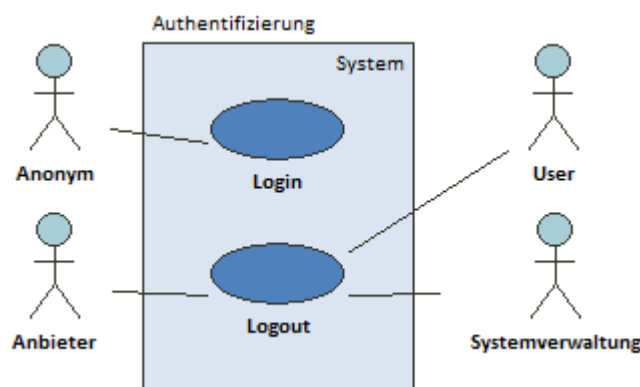


Abbildung 4: Anwendungsfall - Authentifizierung

3.5.3 System verwalten

Die Benutzerrolle Systemverwaltung ist verantwortlich für die Verwaltung von Benutzern, Kompositionen, Kategorien und dem Systemanschluss.

Kategorien dienen zur Klassifikation von Services und bestimmen somit, zu welcher Branche ein Anbieter-Service gehört. "Raumvermietung" ist im Fall des Prototyps die primär verwendete Kategorie. Verschiedene Services können zu Kompositionen kombiniert werden, was ebenfalls von der Systemverwaltung durchgeführt wird.

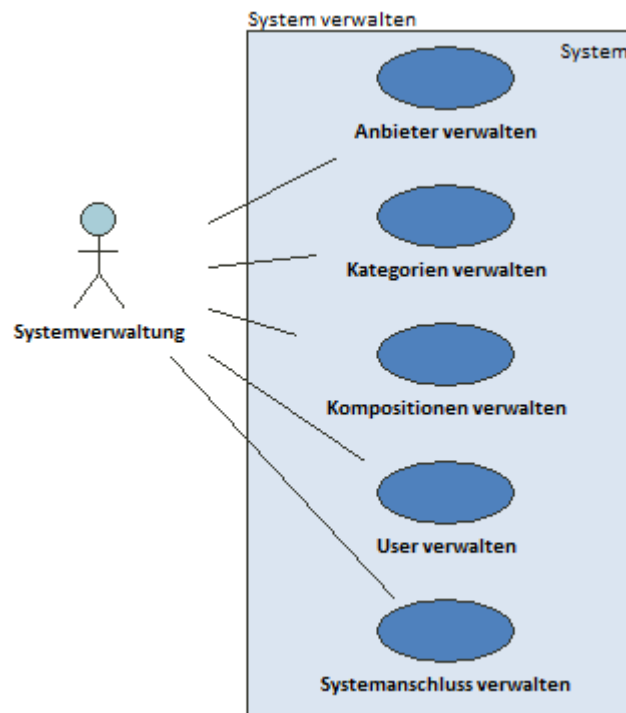


Abbildung 5: Anwendungsfall - System verwalten

Nachdem sich Benutzer im Service-System registriert haben, müssen sie von der Systemverwaltung zuerst freigeschalten werden. Danach wird es Anbietern ermöglicht eigene Services in das System zu integrieren, die ebenfalls von der Systemverwaltung genehmigt werden müssen. Anbieter können natürlich auch wieder das System verlassen oder von der Systemverwaltung entfernt werden. Diese Aufgaben werden unter dem Begriff Systemanschluss zusammengefasst.

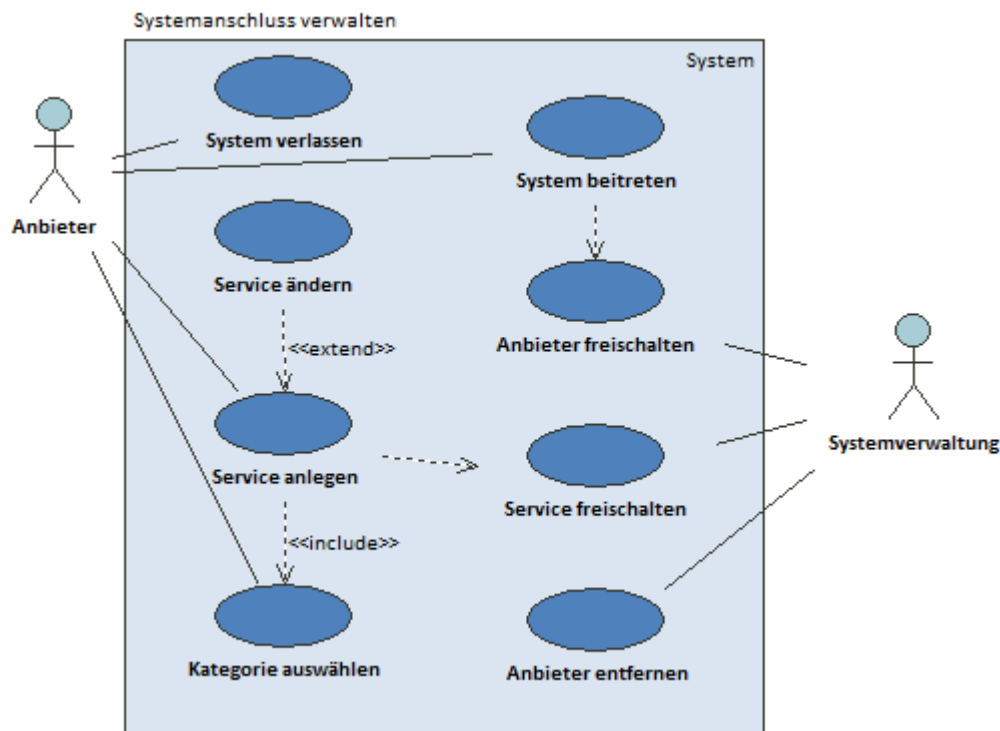


Abbildung 6: Anwendungsfall - Systemanschluss verwalten

3.5.4 Ressourcen reservieren und buchen

Bei den Anwendungsfällen "Ressource reservieren" bzw. "Ressource buchen" handelt es sich bereits um die Verwendung einer Komposition, die im Prototyp exemplarisch gezeigt wird. Dabei werden mehrere Anbieter "simuliert" und diesen Services zugeteilt, um auf deren Ressourcen-Datenbestand zuzugreifen.

Der Benutzer kann über eine Weboberfläche eine Suche per Eingabe von Suchkriterien abschicken. Nachdem kann er, entsprechend der aufgrund der eingegebenen Suchkriterien zurückgelieferten Ergebnisliste, eine oder mehrere Ressourcen - in diesem Fall Räumlichkeiten - auswählen.

Bei der Suche werden schon Zeiträume spezifiziert, in denen die gewünschten Ressourcen vorhanden sein sollen. Die ausgewählten Ressourcen können schlussendlich reserviert werden. Reservierungen müssen natürlich auch angepasst und storniert werden können.

Anbieter können die Reservierungen derer betroffenen Ressourcen natürlich einsehen und verwalten. Zusätzlich können Anbieter entscheiden, ob eine Reservierung von einer bestimmten Ressource überhaupt möglich sein soll oder ob diese ausschließlich buchbar ist. Anbieter sind grundsätzlich eher daran interessiert die Anzahl und Dauer von Reservierungen so gering wie möglich zu halten, um diese für potenzielle Buchungen frei zu haben. So wäre es auch denkbar Geschäftslogiken zu integrieren, in denen die Gültigkeitsdauer von Reservierungen spezifiziert werden kann.

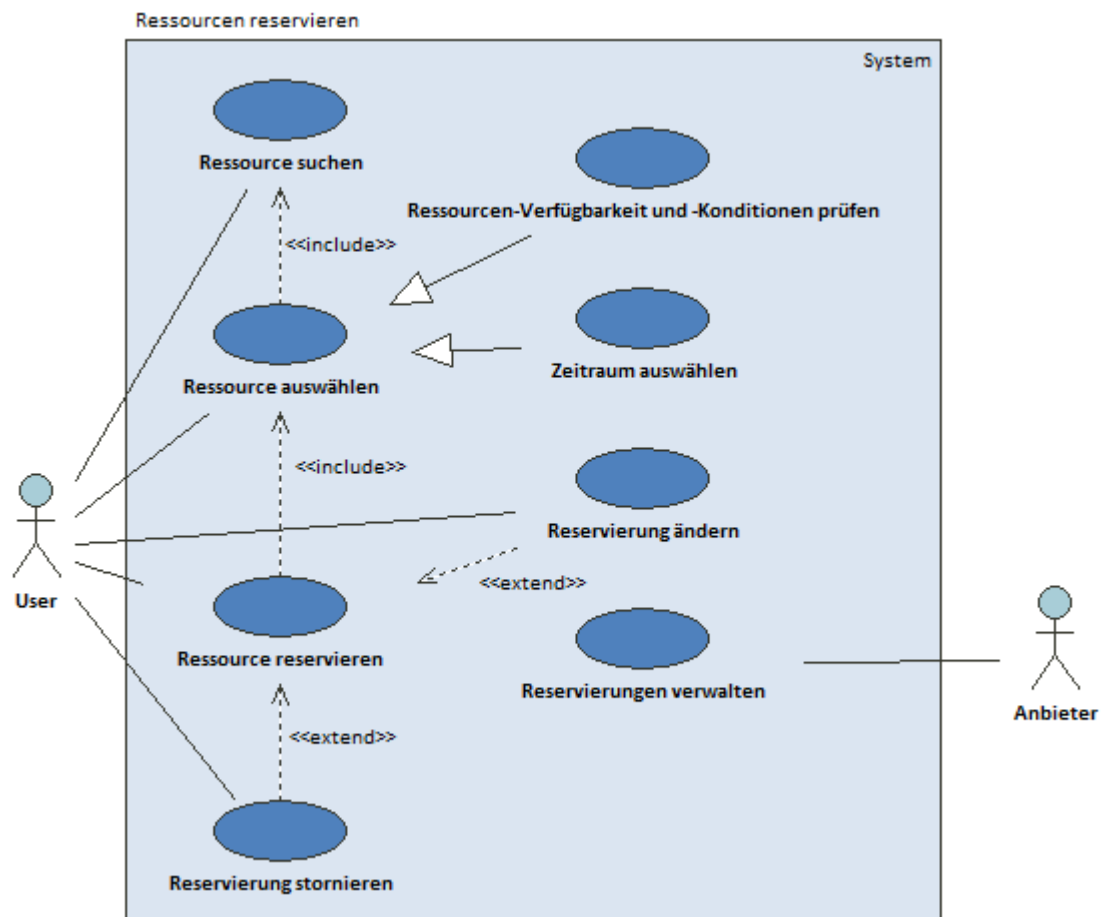


Abbildung 7: Anwendungsfall - Ressourcen reservieren

Der Anwendungsfall "Ressource buchen" ist beinahe identisch mit dem Anwendungsfall "Ressource reservieren" jedoch können Benutzer zusätzlich eine bestehende Reservierung in eine verbindliche Buchung umwandeln.

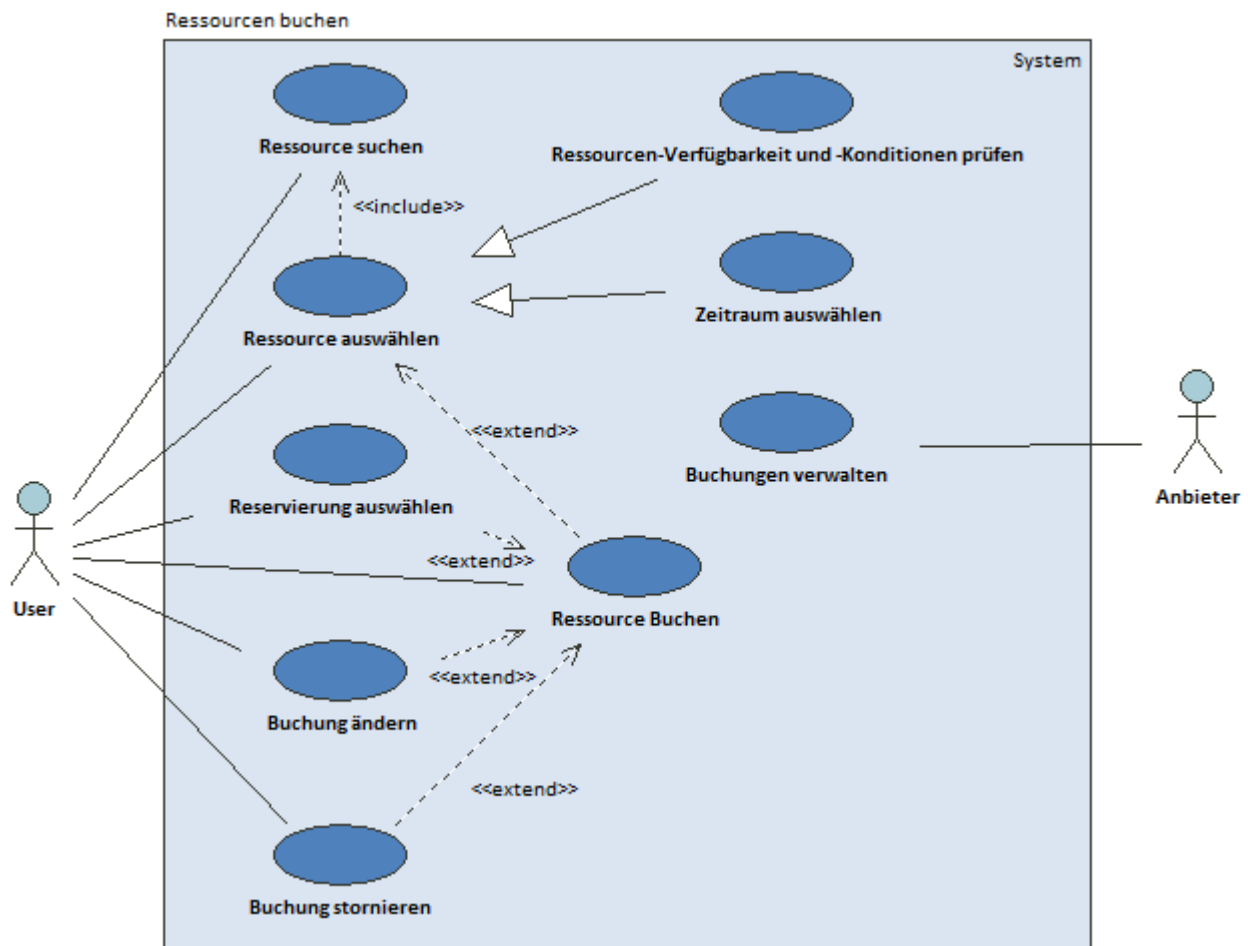


Abbildung 8: Anwendungsfall - Ressourcen buchen

Eine detaillierte Beschreibung des Anwendungsfalls "Ressource buchen":

Anwendungsfall:	Ressource buchen
Akteure:	User
Auslöser:	"Buchung durchführen"
Beschreibung:	1. Ressource durch die Eingabe von Kriterien suchen 2. Ressource für einen Zeitraum auswählen 3. Ressource buchen
Alternativen:	1a. Keine Ergebnisse aufgrund der Suchkriterien 2a. Ressource ist im gewünschten Zeitraum nicht verfügbar
Erweiterungen:	2a. Es kann eine bereits bestehende Reservierung ausgewählt und diese gebucht werden. 3a. Die Schritte 1 und 2 können vorher beliebig oft wiederholt werden. Somit ist die Buchung mehrerer Ressourcen gleichzeitig möglich.
Vorbedingungen:	Um die Buchung abzuschließen, muss der User eingeloggt sein.
Nachbedingungen Erfolg:	Die Buchung ist im System erfasst und wird an das System des Anbieters übermittelt. Benutzer wird über Buchungsinhalt nochmals informiert.
Nachbedingungen Fehlschlag:	Daten bleiben unverändert. Benutzer erhält Nachricht über Fehlschlag.

Tabelle 17: Anwendungsfallbeschreibung - Ressource buchen

3.6 Datenmodell

Um die unterschiedlichen Anwendungsfälle umsetzen zu können, ist ein adäquates Datenmodell notwendig, welches in diesem Kapitel beschrieben wird.

Dieses kann in sechs Gruppen gegliedert werden:

- Userverwaltung
- Serviceverwaltung
- Raumverwaltung
- Kompositionsverwaltung
- Reservierungsverwaltung
- Buchungsverwaltung

3.6.1 Userverwaltung

In der Userverwaltung werden Daten zu den einzelnen Entitäten gesammelt, wobei die Rolle bestimmt, welche Berechtigung die Benutzer im System einnehmen.

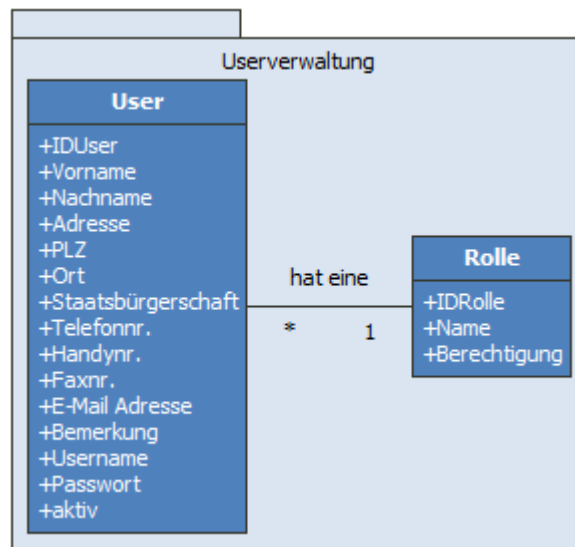


Abbildung 9: Datenmodell - Userverwaltung

3.6.2 Serviceverwaltung

Die Serviceverwaltung spiegelt das Verzeichnis des Service-Systems wieder. Services haben einen Zugangspunkt, über den sie erreicht werden können. Das Serviceprofil könnte einerseits als Ganzes hier in der Relation Service abgelegt werden, oder alternativ ebenfalls verlinkt werden. Ein Service gehört zu einer Kategorie, die wiederum eine oder mehrere definierte Eigenschaftsfelder haben kann. Bei der Kategorie "Raumvermietung" könnten das beispielsweise die Nutzfläche oder die Anschrift sein. Ein Service enthält eine oder mehrere Funktionen, wobei ein Funktionsprofil hinterlegt ist, worin die Fähigkeiten der einzelnen Funktionen beschrieben werden.

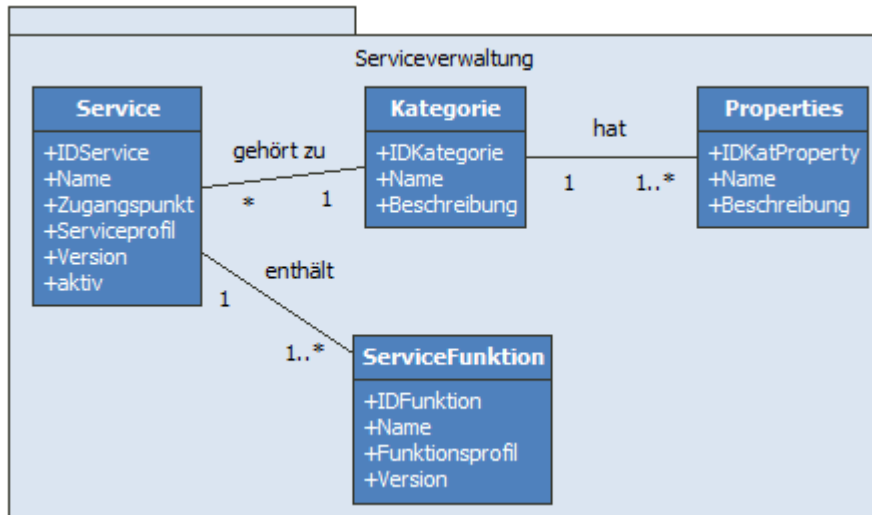


Abbildung 10: Datenmodell - Serviceverwaltung

3.6.3 Raumverwaltung

In der Räumlichkeit-Relation werden die Ressourcen der Anbieter erfasst. Eine Räumlichkeit hat wiederum mehrere Eigenschaftswerte, die sie genauer spezifizieren. Zum einen kann sie verschiedenartig typisiert werden; unterschiedliche Raumtypen könnten beispielsweise Konferenzräume, Besprechungsräume, Interviewräume, Schulungsräume oder ganze Gebäude sein.

Bei der Erfassung der Eigenschaftswerte der Räumlichkeiten bieten sich unterschiedliche Varianten an. Einerseits können in einer Relation (siehe dazu Abbildung11) verschiedene Typen von Details, also von Eigenschaften, definiert werden. Das können - wie bereits im Kapitel zum "klassischen Buchungsvorgang" erwähnt - unterschiedliche Ausstattungsmerkmale sein; wie beispielsweise Umgebungsdaten oder Daten zu Rauminstallationen. In der Relation "Raumdetails" werden den Ausstattungsdetails Werte zugeordnet.

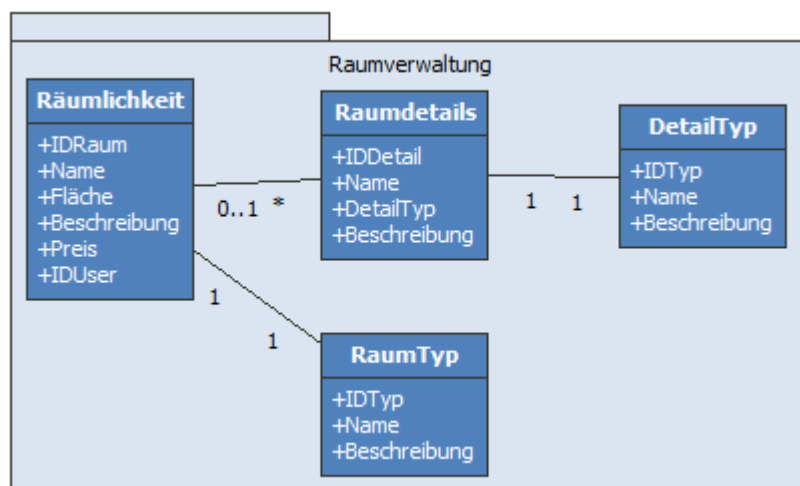


Abbildung 11: Datenmodell - Raumverwaltung

Alle Ausstattungsmerkmale direkt in der Relation "Räumlichkeit" zu erfassen wäre eine andere Variante. In diesem Fall können jedoch nicht flexibel die einzelnen Eigenschaften angepasst werden (siehe dazu Abbildung12). In dieser Abbildung sind auch die Preise anders modelliert worden. In Abbildung "Raumverwaltung" ist der Preis der Räumlichkeit noch direkt in der "Räumlichkeit"-Relation hinterlegt. Dieser kann zwar auch problemlos geändert werden und zu jeder Buchung wird der Preis sowieso separat gespeichert, jedoch lässt sich mit diesem Modell alleine nicht direkt eine Verkaufs-Logik darstellen. Mittels der Relation "Raumpreise" in

Abbildung "Raumverwaltung2" können zu einer Räumlichkeit verschiedene Preise definiert werden. Der Preisbereich definiert, in welchem Fall welcher Preis gültig ist. So können Standardpreise hinterlegt werden, aber auch Rabatte und ähnliche Preiskontingente definiert werden.

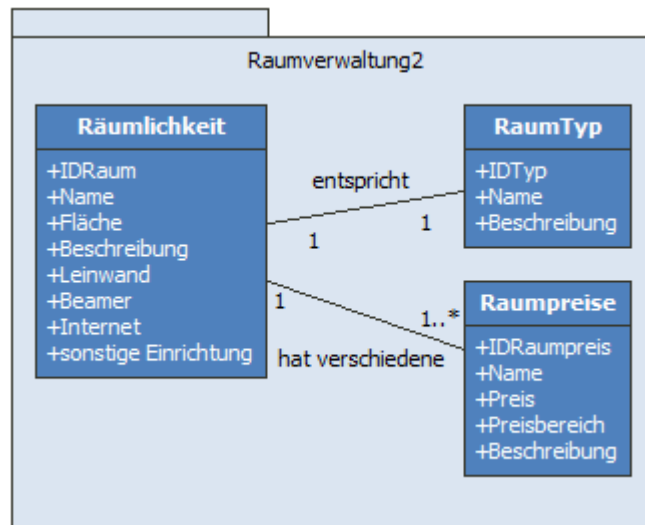


Abbildung 12: Datenmodell - Raumverwaltung2

Eine weitere Möglichkeit der Ressourcenabbildung im Datenmodell soll Abbildung "Ressourcenverwaltung" zeigen. Es ist das generischste Modell, da es prinzipiell für alle Kategorien anwendbar ist. Es können Ressourcen angelegt werden, die ein oder mehrere Eigenschaftswerte haben kann. Die möglichen Eigenschaftswerte werden in der Relation "Properties" definiert. Die Verwaltung dieses Modells ist allerdings auch am aufwendigsten.

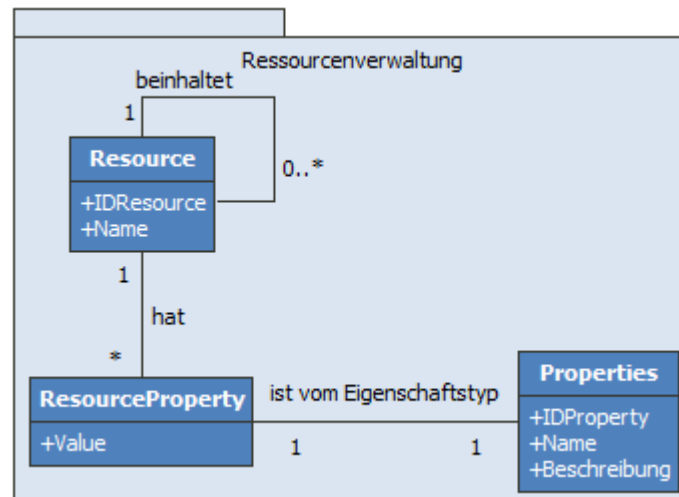


Abbildung 13: Datenmodell - Ressourcenverwaltung

Bei allen Varianten muss beachtet werden, dass die Ressourcendaten der Anbieter vor der Suche nicht in der Systemdatenbank gespeichert sind, sondern über Web Services erst eingeholt werden. Für Bestellungen und Reservierungen müssen diese allerdings im System hinterlegt sein, um auf diese verweisen zu können, ohne abermals auf die Anbieterdaten zugreifen zu müssen. Dementsprechend müssen sie bei einer Reservierung oder Buchung in der Systemdatenbank abgelegt werden, um die Buchung ordnungsgemäß speichern zu können und ein Verweis auf die Ressourcen möglich ist.

3.6.4 Kompositionsverwaltung

In der Relation *Komposition* werden zunächst der Name, der Ersteller und die Beschreibung der Komposition erfasst. Zu den *Kompositionsdetails* werden die in der Komposition enthaltenen Service-Funktionen erfasst und vermerkt, welches Service diese startet. Über den *Kompositionspfad* wird festgelegt, welches Service nach Ablauf des Vorgängers ausgeführt werden soll. Dies wird über die Angabe von Wertebereichen erreicht, wobei bei der Kompositions-Koordination ein Rückgabewert mit dem Wertebereich verglichen wird.

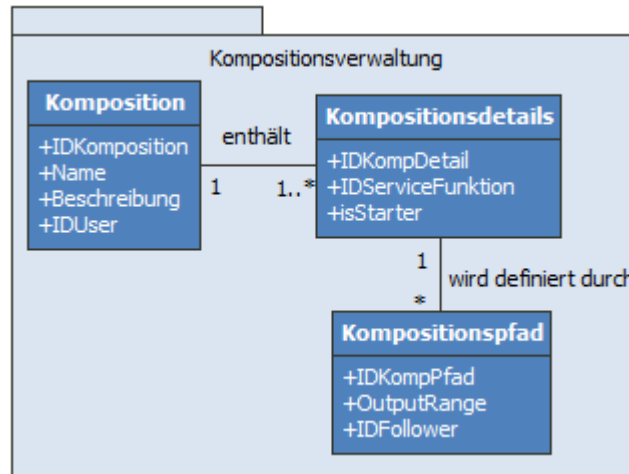


Abbildung 14: Datenmodell - Kompositionsverwaltung

Der Kompositionspfad kann in diesem Fall nur bedingt eingesetzt werden. Für simple Datentypen funktioniert das Modell problemlos, jedoch wenn Funktionen komplexe Datentypen zurückliefern, ist es schwer, diese in einer einfachen Abfrage auf gewisse Kriterien zu prüfen. In der Relation *Kompositionspfad* dient das Feld *OutputRange* als Überprüfungs-kriterium. Dies könnte beispielsweise ein Standardparameter sein, der angibt, dass unabhängig vom Rückgabewert eine bestimmte Funktion aufgerufen werden soll. Es könnte aber auch ein fixer Wert oder eine Wertebereich angegeben werden.

Wünschenswert wäre es auch, wenn auch Funktionen nicht nur einzeln als Kompositionsschritt hinzugefügt werden können, sondern auch in Gruppen, die dann gleichzeitig abgearbeitet werden. Dies bietet sich beispielsweise für Such- und Buchungs-Funktionen an. So müssen Anbieter nicht seriell abgearbeitet werden sondern können parallel kontaktiert werden, um die Laufzeit zu beschleunigen.

3.6.5 Reservierungs- und Buchungsverwaltung

Eine Reservierung oder Buchung kann jeweils eindeutig einem Benutzer, der diese durchgeführt hat zugeordnet werden. Dahinter stehen noch ein oder mehrere Räumlichkeiten und die dazugehörigen Preise und Zeiträume. Zeiträume definieren wann und wie lange die Räumlichkeit vermietet wird. Die Zeiträume werden dabei entsprechend dem standardisierten iCalendar-Format gespeichert. iCalendar ist ein Standard zum Austausch von Kalenderinformationen und wird in vielen gängigen Applikationen verwendet. So wird es Benutzern erleichtert Buchungs-Termine in andere Applikationen zu übertragen.

Ein Beispiel für einen iCalendar-Termin:

```

BEGIN:VCALENDAR
BEGIN:VEVENT
SUMMARY:Kurzinformation
DESCRIPTION:Beschreibung des Termins
DTSTART: 20100101T100000
DTEND:20100101T1140000
DTSTAMP:20100101T140000
END:VEVENT
END:VCALENDAR
    
```

Tabelle 18: Buchung - iCalendar-Termin

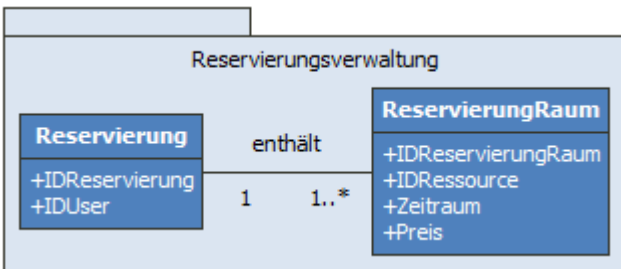


Abbildung 15: Datenmodell - Reservierungsverwaltung

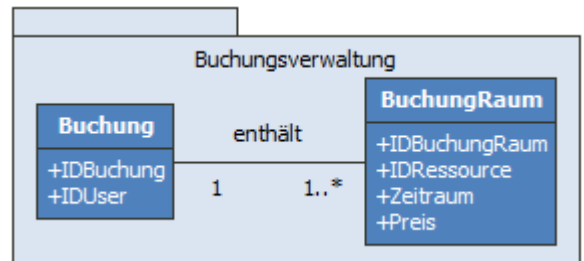


Abbildung 16: Datenmodell - Buchungsverwaltung

3.6.6 Gesamtmodell

In folgender Abbildung ist das Zusammenspiel der zuvor einzeln beschriebenen Module erkennbar.

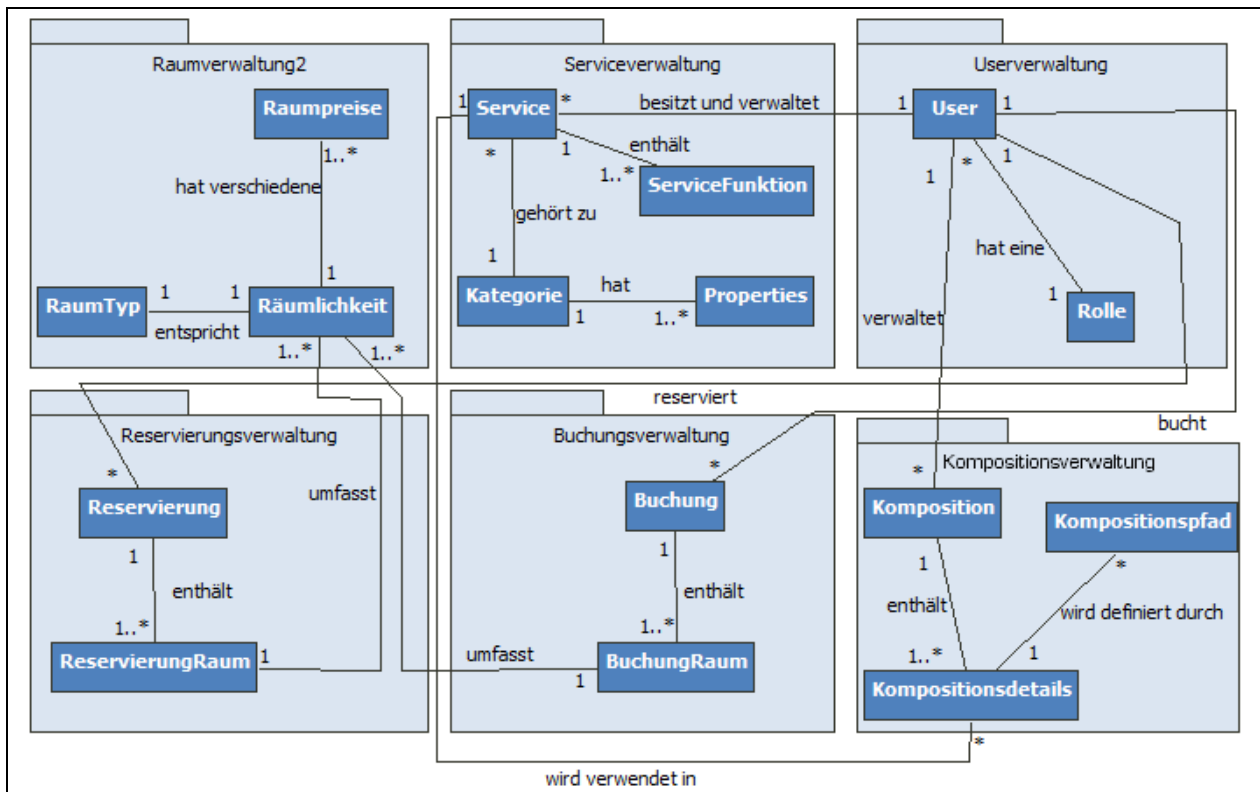


Abbildung 17: Datenmodell - Gesamtmodell

3.7 Abläufe im System

Anhand den simpel dargestellten Flussdiagrammen soll exemplarisch gezeigt werden, wie der Buchungsvorgang bei einer simplen Komposition ablaufen soll. Dabei werden die zentralen Prozesse des Prototyps dargestellt.

Während des Suchprozesses wird zuerst eine Kategorie bzw. eine Komposition ausgewählt. Durch diese Wahl wird bestimmt, nach welcher Art von Ressourcen bzw. in welcher Branche gesucht werden soll. Anschließend kann der Benutzer gewünschte Suchkriterien angeben, um die spätere Ergebnisliste einzuschränken. Mit der Aktion "Suchabfrage abschicken" wird die Suchfunktion aufgerufen und der Suchprozess somit abgeschlossen.

Der Prozess der Reservierung wird mit dem Suchprozess eingeleitet. Dieser liefert eine Ergebnisliste, aus der der Benutzer anschließend gesuchte Ressourcen auswählen kann. Bei der Auswahl muss er einen Termin bzw. die Dauer der Reservierung angeben. Danach kann er entscheiden, ob er eine weitere Ressource auswählen oder mit der Aktion "Reservierung abschicken" die Reservierung abschließen möchte.



Abbildung 18: Systemabläufe - Suchvorgang



Abbildung 19: Systemabläufe - Reservierungsvorgang

Der Buchungsprozess ist dem der Reservierung sehr ähnlich. Zu Beginn des Prozesses kann der Benutzer eine bereits bestehende Reservierung auswählen, um Ressourcen auszuwählen. Alternativ kann abermals der Suchprozess verwendet werden, um Ressourcen zu finden. Diese können anschließend wieder ausgewählt und ein Termin dafür bestimmt werden. Danach muss abermals entschieden werden, ob eine weitere Ressource benötigt wird. Die Buchung wird mit der Aktion "Buchung abschicken" abgeschlossen.

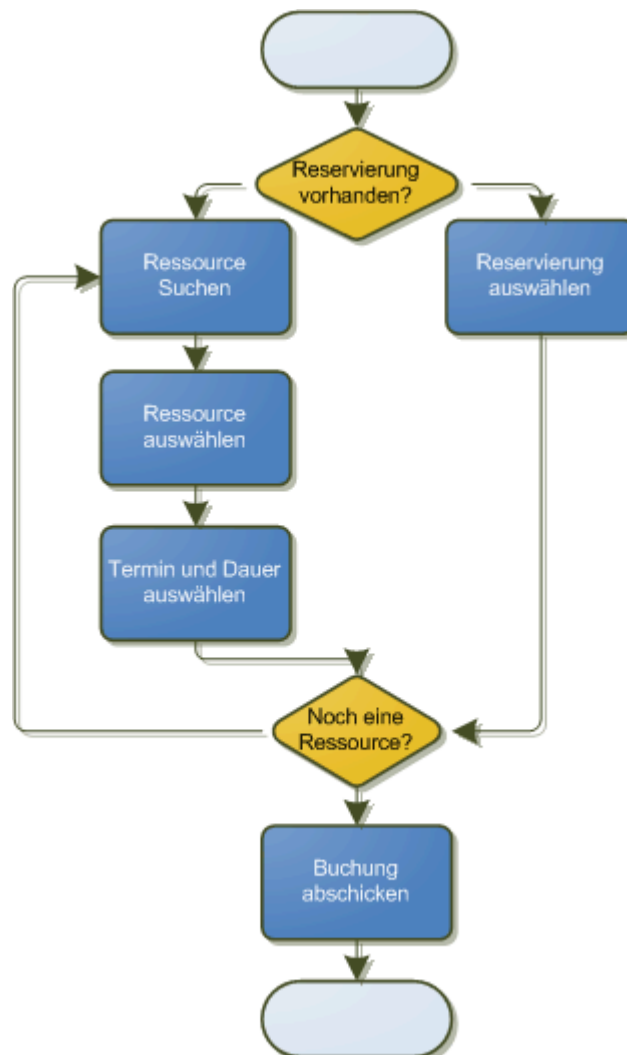


Abbildung 20: Systemabläufe - Buchungsvorgang

4. Systembeschreibung

In diesem Kapitel wird der im Rahmen der Diplomarbeit entwickelte Prototyp beschrieben und es wird dargestellt, wie dieser umgesetzt wurde. Es wird auch auf alternative Lösungswege eingegangen, die nachfolgend in der Evaluierung detailliert behandelt werden.

4.1 Softwarearchitektur

4.1.1 Entwicklungsumgebung

Der Prototyp des Service-Systems wurde mit der Skriptsprache PHP³⁰ umgesetzt, wobei die aktuelle Version 5 verwendet wurde. PHP ist eine Open-Source-Software, dessen Syntax den Programmiersprachen C bzw. Java ähnelt und sie wird primär für die Umsetzung dynamischer Webseiten eingesetzt wird.

Hauptentscheidungskriterien für die Wahl der Softwarearchitektur waren einerseits bereits gute persönliche Kenntnisse sowie Erfahrung in der verwendeten Software und die Verwendung von nicht-proprietärer Software. Darüber hinaus sollte möglichst schnell eine brauchbare Testumgebung geschaffen und einfache Anpassungen am User Interface möglichst rasch durchgeführt werden können. Außerdem war es der Wunsch, das Service-System als Webanwendung im Browser ohne zusätzlichen Client verwenden zu können. Die Ausgabe eines Skriptsprachen-Interpreters ist meist ein HTML-Dokument, wodurch diese Forderung abgedeckt ist.

Desweiteren bietet PHP eine Fülle an zusätzlichen Programmbibliotheken, die den Funktionsrahmen erweitern und ständig weiter entwickelt werden. Es ist eine der am häufigsten eingesetzten Skriptsprachen im Web. Es unterstützt viele verschiedene Datenbanktypen und ist unter anderem deshalb so beliebt, da es relativ einfach erlernbar ist. Außerdem steht online eine Fülle von Lernmaterial zur Verfügung. Da es sich dabei um eine serverseitige Ausführung handelt und nur ein Client in Form eines Browsers notwendig ist, können somit auch Inkompatibilitäten weitgehend ausgeschlossen werden.

Die Wahl dieser Softwarearchitektur birgt teilweise auch Nachteile in sich, die allerdings mit dem Einsatz zusätzlicher Lösungen problemlos beseitigt werden könnten. Da beispielsweise für jeden HTTP-Request der Webserver eine neue Instanz des PHP-Interpreters anlegen muss, kann die Verwendung von PHP als CGI-Lösung die Laufzeitgeschwindigkeit negativ beeinflussen. Auch der Ressourcenverbrauch anderer Skriptsprachen, wie beispielsweise Perl, wird teilweise geringer eingestuft. In der nächsten Version von PHP wären zusätzliche Funktionen zur Generierung von WSDL-Dateien wünschenswert, welche in anderen Implementierungen bereits integriert sind. Weiters sind noch nicht alle aktuellen Standards (z.B. WSDL 2.0) unterstützt.

Alternative Architekturen für die Umsetzung wären beispielsweise:

- Java Platform, Enterprise Edition (JEE)
- Microsoft .NET
- Apache AXIS

³⁰ PHP – Akronym für "PHP: Hypertext Preprocessor"; Skriptsprache. [PHP10]

Ein PHP-Dokument ist auf einem Webserver gespeichert und kann von einem Nutzer online über einen Browser abgerufen werden. Zum Zeitpunkt der Anfrage beauftragt der Webserver den PHP-Interpreter mit der Verarbeitung des Dokuments entsprechend eventueller Eingaben. Der Interpreter verarbeitet den im PHP-Dokument enthaltenen Code und liefert die Ergebnisse an den Webserver zurück, der wiederum eine entsprechende Antwort an den Browser des Nutzers schickt. Meist liefert der Webserver dabei ein HTML-Dokument zurück, es können jedoch mit PHP aber auch andere Dateitypen, wie beispielsweise Bilder, generiert werden. Der PHP-Quelltext bleibt also am Server und ist für den Client, genauso wie die andere dahinter stehende Architektur nicht, genau ersichtlich.

Für die korrekte Ausführung von PHP wird abseits des Interpreters also ein Webserver benötigt, wobei hier die Entscheidung für den meist verwendeten und plattformunabhängigen Apache³¹ gefallen ist.

4.1.2 Datenbank

Zur Umsetzung des relationalen Datenmodells wurde MySQL³² – ebenfalls in der Version 5 – als Datenbank gewählt, wobei die Open-Source-Variante eingesetzt wurde. Um Abfragen und Anpassungen an der Datenbank durchzuführen, wurde die Datenbanksprache SQL verwendet. Die Datenbankzugriffe wurden dabei über die in PHP integrierten Bibliotheken zur Kommunikation mit MySQL-Datenbanken durchgeführt. Mit der Funktion `mysql_connect` wird eine Verbindung zu einem MySQL-Server geöffnet, wobei Benutzername, Passwort und der entsprechende Hostname angegeben werden müssen. Mit der Funktion `mysql_select_db` wird dann die gewünschte Datenbank am Server ausgewählt, die zu diesem Zeitpunkt schon existieren muss. Mittels der `mysql_query`-Funktion wird eine SQL-Anfrage an die Datenbank gesendet. Falls die Anfrage Ergebnisse zurückliefert (meist bei einer SELECT-Anfrage), können diese mit der Funktion `mysql_fetch_object` einem Objekt zugewiesen und mit diesem dann weiterverarbeitet werden.

4.1.3 Web Services mit PHP

In PHP 5 ist bereits nativ eine SOAP-Erweiterung in Form einer Bibliothek integriert, die Klassen und Methoden für den Einsatz von Web Services bereitstellt. Mithilfe der Funktion `phpinfo` kann eruiert werden, ob diese Erweiterung aktiviert ist und somit geklärt werden, ob auf dem Webserver der Einsatz von Web Services möglich ist. Die fertigen Bibliotheken erleichtern den Entwicklern den Einsatz von SOAP, da sie sich nicht mehr selbst um die Generierung der XML-Nachrichten zur Web Service Kommunikation kümmern müssen.

In PHP gibt es zwei verschiedene Möglichkeiten Web Services bereitzustellen. So kann ein sogenannter SOAP-Server mit oder ohne WSDL-Beschreibung angelegt werden. Innerhalb des Prototypen wurde zu jedem Web Service eine WSDL-Beschreibung erstellt.

Für die Erstellung eines Web Services muss zuerst eine Funktion oder Klasse erstellt werden, die eine gewisse Logik umsetzt. Im Fall des Service-Systems handelt es sich dabei meist um Datenbankzugriffe in Form von Abfragen oder Anpassungen. Innerhalb der PHP-Datei, die zum Aufruf des Web Services dienen soll, muss ein Objekt vom Typ `SoapServer` angelegt werden, wobei der Pfad zu einer WSDL-Datei übergeben wird.

³¹ Apache Webserver – der meistbenutzte Webserver im Internet [ASFo10]

³² MySQL – relationales Datenbankverwaltungssystem [Orac10]

Die SOAP-Bibliothek innerhalb von PHP 5 unterstützt momentan nur die "WSDL 1.1"-Spezifikation. Eine WSDL-Datei muss ein entsprechend der Spezifikation definiertes Format haben und auf dem Webserver im gewünschten Verzeichnis abgelegt werden.

Durch die Methode *addFunction* kann dem *SoapServer*-Objekt eine Funktion hinzugefügt werden, die dann von einem Client aufgerufen werden kann. Anstatt der Verwendung von *addFunction* für einzelne Funktionen kann die Methode *setClass* des *SoapServer*-Objekts verwendet werden, um alle Funktionen einer Klasse dem Web Service freizugeben.

Das *SoapServer*-Objekt wird mittels der Methode *handle* zur Behandlung von Client-Anfragen beauftragt.

Der Zugriff auf ein Web Service wird mittels einer Client-Komponente realisiert. Dabei wird ein Objekt des Typs *SoapClient* verwendet, dem die entsprechende URI auf die WSDL-Datei übergeben wird. Mit Hilfe des Client-Objekts kann dann eine Operation des Web Services ausgeführt werden, wobei das Endergebnis, sofern es eines geben sollte, einer Variable zugewiesen werden kann.

Der Entwickler muss sich dabei keine Gedanken um die Erstellung und den Versand der SOAP-Nachrichten machen, da SOAP-Bibliothek die Generierung der Nachricht im korrekten XML-Format durchführt und entsprechend der im WSDL-Dokument definierten Kodierung und Protokoll versendet.

Die Client-Komponente in Form des *SoapClient*-Objekts bietet noch mehrere nützliche Funktionen, die den Umgang mit Web Services erleichtern. Beispielsweise liefert die Methode *getFunctions* alle in einem WSDL-Dokument definierten Funktionen und die Methode *getTypes* eine Liste aller in der WSDL-Datei vorkommenden Datentypen. Diese Methoden bieten sich unter anderem an, um eine Logik zur automatischen Aufdeckung von Web Services zu unterstützen.

4.1.4 Testumgebung des Prototyps

Es können fertige Installations-Pakete verwendet werden, um schnell eine entsprechende Umgebung aufzustellen. Eine solche freie Zusammenstellung ist beispielsweise XAMPP³³, die für die unterschiedlichsten Betriebssysteme zur Verfügung steht. XAMPP ermöglicht eine schnelle und einfache Installation und Konfiguration und beinhaltet den Webservers Apache, die Datenbank MySQL, die Skriptsprachen PHP und Pearl, den FTP-Server FileZilla Server und den Mailserver Mercury. Zusätzlich werden noch einige Webanwendungen mit installiert, um die verschiedenen Installationen konfigurieren zu können. Somit können schnell verschiedene Installationen für Testserver und -umgebungen durchgeführt werden. Für einen produktiven Einsatz eignet es sich allerdings nicht ohne weiteres.

³³ XAMPP – Zusammenstellung freier Software von apachefriends.org [ApFr10]

4.2 Umsetzung des Prototyps

Das folgende Kapitel beschäftigt sich mit der technischen Umsetzung des Prototyps und seiner unmittelbaren Umgebung.

4.2.1 Systemfunktionen

Das Service-System bietet unterschiedliche Services an, um dieses zu verwalten und benutzen zu können. Dabei kann zwischen Services zur Systemverwaltung – systeminternen Services – und Services innerhalb von Kompositionen unterschieden werden.

Systeminterne Services:

- Services zur Authentifizierung.
- Services zur Benutzerverwaltung.
- Services zur Verwaltung des Verzeichnisses.
- Services zur Kompositionsverwaltung.

Systeminterne Services dienen zur Verwaltung und Wartung des Service-Systems und stehen über dieses zur Verfügung. Die Servicebeschreibungen sind ebenfalls im Serviceverzeichnis abgelegt, weiters sind die dazugehörigen WSDL-Dokumente am Server des Service-Systems abgelegt. Die Services selbst werden ebenfalls auf diesem Server ausgeführt, wobei diese auch zusätzlich auf einem anderen Ort redundant angeboten werden. Dies wäre allerdings erst dann sinnvoll, wenn auch die Daten redundant abgelegt werden.

Services innerhalb Kompositionen:

- Services zur Suche und Auffindung von Ressourcen.
- Services zur Buchung von Ressourcen.

Services innerhalb von Kompositionen werden im Normalfall nicht vom Service-System selbst zur Verfügung gestellt. Dabei handelt es sich um Services, die von Anbietern zur Verfügung gestellt werden, um auf deren Datenbestand zuzugreifen.

4.2.2 Das Anbietersystem

Um die Funktion des Service-Systems veranschaulichen zu können, war die Entwicklung eines Anbietersystems notwendig. In diesem Anbietersystem wurden Datenstrukturen angelegt, um Räumlichkeiten und Buchung ablegen zu können. Das Anbietersystem beinhaltet Web Services, um diese Daten durchsuchen und anschließend Buchungen ausführen zu können. Es wird angenommen, dass etwa solch ein System und Datenbestand (natürlich weitaus komplexer) bei einem Anbieter von Räumlichkeiten zum Einsatz kommt.

In folgender Abbildung ist das dazugehörige ER-Diagramm abgebildet.

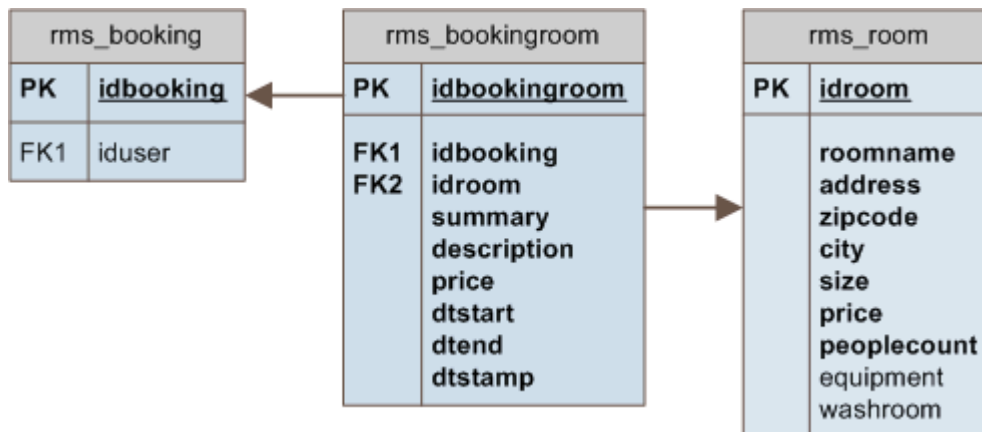


Abbildung 21: ER-Diagramm - Anbietersystem

In der Relation *rms_room* werden die Stammdaten von Räumlichkeiten erfasst. Es muss ein Name der Räumlichkeit, eine Anschrift, die Fläche, der Preis und die Personenzahl eingegeben werden. Eine Beschreibung von Einrichtung und Sanitäreinrichtungen sind keine Pflichtangaben. Wie bereits im Entwurf beschrieben, gibt es noch andere Daten, die für Räumlichkeiten erfasst werden können oder sogar sollten, um diese besser voneinander differenzieren zu können. Beispielsweise könnten noch weitere Stammdaten erfasst, in einer dahinter liegenden Relation noch Raumtypen verwaltet und die Preise anders modelliert werden.

Die Datenstrukturen solcher Stammdaten wurden im Prototyp absichtlich einfach gehalten, um die Konzentration auf das Wesentliche – nämlich die Kommunikation zwischen Service- und Anbietersystem – zu lenken.

Die Tabelle *rms_booking* dient zur Speicherung von Buchungen, wobei eine eindeutige Buchungsnummer (*idbooking*) und der Kunden, der die Buchung ausgeführt hat, erfasst werden. Über den Fremdschlüssel *iduser* wird dabei auf die Userrelation verlinkt, wobei diese im Service-System gespeichert ist. Ein Web Service des Service-Systems dient zur Abfrage der Userinformationen, womit mit Angabe der User-ID die Kundendaten abgerufen werden können.

In einem Produktivsystem würde es durchaus Sinn machen für die Kundenstammdaten eine eigene Kundenrelation anzulegen, um diese einheitlich verwalten zu können. Dabei sollten beispielsweise Firmen- und weitere Kontaktdaten abgelegt werden. In den meisten professionellen Shop-Systemen können auch verschiedene Kundenadressen hinterlegt werden, um beispielsweise zwischen Rechnungs- und Lieferadresse unterschieden zu können. Eine Erfassung von Daten für den späteren Zahlungsvorgang ist natürlich auch essentiell.

Mithilfe der *rms_bookingroom*-Relation werden einzelne Räumlichkeiten zu einer Buchung zugeordnet. So ist es möglich, Buchungen abzuschließen, die mehr als eine Räumlichkeit beinhalten. Über das Feld *idbooking* wird auf die *rms_booking*-Relation verwiesen. Mit *dtstart* und *dtend* werden Start- und Endzeitpunkt einer Buchung erfasst. Das Feld *dtstamp* spiegelt den Zeitpunkt des Buchungsabschlusses wieder. Der Preis der Buchung – also die Mietkosten des Objekts, die für den Kunden anfallen – werden auch hier hinterlegt. In einem Echtssystem müssten natürlich Rechnungsdaten erfasst werden, um genau Preise, Steuern, Rabatte und ähnliches aufschlüsseln zu können. Die Felder *summary* und *description* dienen dazu, um kurze bzw. lange Detailinformationen zur Buchung hinterlegen zu können.

Auch beim Anbietersystem wurde PHP eingesetzt, um auf die gerade eben beschriebenen Datenstrukturen – die in einer MySQL-Datenbank angelegt wurden – zuzugreifen. Es wurden

dabei die in der Softwarearchitektur beschriebenen Funktionen eingesetzt. HTML-Webformulare wurden eingesetzt, um Eingaben zu realisieren. In einem PHP-Skript gibt es unterschiedliche Varianten, um auf Formulardaten zuzugreifen. Im Prototypen wurden die superglobalen Array `$_POST` für den Zugriff auf Formulardaten und `$_GET` für die Abfrage von sogenannten Querystrings (werden über die URL mitgeliefert) eingesetzt. So wurden PHP-Seiten erstellt, um Räume oder auch Buchungen anzulegen und zu verwalten.

Die für das Services-System jedoch wirklich relevanten Anbieter-Komponenten sind deren Web Services. So wurde einerseits ein Web Service erstellt, um Anbieterdaten – sprich deren verfügbaren Räumlichkeiten – zu durchsuchen. Dazu wurde eine PHP-Klasse erstellt, die eine Suchfunktion beinhaltet. Der Suchfunktion müssen mehrere Parameter übergeben werden, die der Übergabe der Suchkriterien dienen. So müssen Parameter für eine Postleitzahl, einen Ort, eine minimale und maximale Flächenangabe, ein minimaler und maximaler Preis sowie eine minimale und maximale Personenanzahl übergeben werden. Wenn bei Postleitzahl und Ort ein leerer String oder bei den anderen Parameter ein Maximalwert von 0 übergeben wird, dann werden diese Parameter in der Suchabfrage nicht berücksichtigt. Wenn den Räumlichkeitsdaten neue Felder hinzugefügt werden, sollten diese natürlich auch darauffolgend in der Suchfunktion berücksichtigt werden.

Die übergebenen Parameter werden daraufhin verarbeitet und daraus ein SQL-Befehl generiert, der dazu verwendet wird, eine Abfrage an die Datenbank zu schicken. Die entsprechend der Kriterien zurückgelieferte Antwort des Datenbankservers wird verarbeitet und in ein zweidimensionales Array überspielt. Dadurch werden die Daten serialisiert, was notwendig ist, um diese mittels SOAP sinnvoll weiterverarbeiten zu können. In der ersten Dimension des Arrays wird die Nummer und in der zweiten die Datenfelder des Datensatzes geschrieben.

Daraufhin wurde zu dieser Funktion eine WSDL-Datei (Abbildung) erstellt, um diese zu beschreiben und daraufhin als Web Service zugänglich zu machen. Im Types-Element wird ein komplexer Datentyp definiert, der dann für die Antwort des Web Services verwendet wird. In den messages-Elementen wird eine abstrakte Definitionen der zu übertragenen Nachrichten erstellt. Die Nachrichtendefinition *rms_doSearchRequest* enthält die Parameter für den Aufruf der Suchoperation. Das message-Element *rms_doSearchResponse* verwendet den zuvor definierten komplexen Datentyp und stellt die Rückmeldung der Suchabfrage dar, also die Suchergebnisse.

Im portType-Element *rms_searchPortType* wird definiert, welche der zuvor beschriebenen Nachrichten als Eingangs- beziehungsweise Ausgangsnachricht verwendet werden. Dem binding-Element *rms_searchBinding* wird der darüber definierte Port-Typ zugewiesen und es legt fest, welches Protokoll und welche Kodierung verwendet werden soll.

Schlussendlich wird im service-Element *rms_search* noch ein Port spezifiziert, in dem festgelegt ist, welches Binding verwendet werden soll und über welche Kommunikationsschnittstelle das Web Service aufgerufen werden soll. Um dies durchführen zu können, habe ich auf meinem Testserver die Web Services angelegt.

```

<?xml version ='1.0' encoding ='UTF-8' ?>
<definitions name='rms_search'
  xmlns:tns=' http://datest.sirka.at/raumverwaltung/rms_search.wsdl '
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
  xmlns:wSDL='http://schemas.xmlsoap.org/wsdl/'
  xmlns='http://schemas.xmlsoap.org/wsdl/'>
<types>
  <schema targetNamespace="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="rms_searchreturn">
      <complexContent>
        <restriction base="SOAP-ENC:Array">
          <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="xsd:string[][]"/>
        </restriction>
      </complexContent>
    </complexType>
  </schema>
</types>
<message name='rms_doSearchRequest'>
  <part name='zipcode' type='xsd:string'/>
  <part name='city' type='xsd:string'/>
  <part name='sizemin' type='xsd:integer'/>
  <part name='sizemax' type='xsd:integer'/>
  <part name='pricemin' type='xsd:integer'/>
  <part name='pricemax' type='xsd:integer'/>
  <part name='peoplemin' type='xsd:integer'/>
  <part name='peoplemax' type='xsd:integer'/>
</message>
<message name='rms_doSearchResponse'>
  <part name='searcharray' type='xsd1:rms_searchreturn'/>
</message>
<portType name='rms_searchPortType'>
  <operation name='rms_doSearch'>
    <input message='tns:rms_doSearchRequest'/>
    <output message='tns:rms_doSearchResponse'/>
  </operation>
</portType>
<binding name='rms_searchBinding' type='tns:rms_searchPortType'>
  <soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http'/>
  <operation name='rms_doSearch'>
    <soap:operation soapAction=' http://anbieter1.sirka.at /rms_doSearch'/>
    <input>
      <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
    </input>
    <output>
      <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
    </output>
  </operation>
</binding>
<service name='rms_search'>
  <port name='rms_searchPort' binding='rms_searchBinding'>
    <soap:address location=' http://anbieter1.sirka.at /rmssearch.php'/>
  </port>
</service>
</definitions>

```

Tabelle 19: WSDL - Beschreibung des Web Services zur Suche

Über die im WSDL-Dokument angegebene Kommunikationsschnittstelle kann auf das Web Service zugegriffen werden. In dieser PHP-Datei wurde ein *SoapServer*-Objekt mit Verweis auf die WSDL-Datei angelegt und mit der Funktion *setClass* die Such-Klasse und somit auch die Suchfunktion dem Web Service hinzugefügt. Durch den Funktionsaufruf *handle* wird der SOAP-Nachrichtenaustausch initiiert.

Das Anbietersystem stellt natürlich auch für die Buchung ein Web Service zur Verfügung, um die zuvor gefundenen Räumlichkeiten auch anschließend buchen zu können. Um dies durchführen zu können wurde, Klasse *rms_booking* erstellt, die zwei Funktionen enthält. Die erste (*rms_addBooking*) dient dazu, einen Datensatz in der Relation *rms_booking* anzulegen. Der Funktion müssen somit die Parameter Vorname, Nachname, PLZ, Ort, Email und Telefon übergeben werden. Als Rückgabewert liefert die Funktion, die neue eindeutige Buchungsnummer (*idbuchung*). Weiters beinhaltet die Klasse noch die Funktion *rms_addBookingDetail*, die wiederum zur Erstellung von Datensätzen in der Tabelle *rms_bookingroom* dienen soll. Dieser wiederum müssen die entsprechende Buchungsnummer, die Raumnummer und die Termindaten übergeben werden, worauf diese in die Datenbank geschrieben werden. Auch diese Funktion liefert die neue ID als Rückgabewert zurück.

Um auch diese Funktionen als Web Service nutzen zu können, wurde dafür ein WSDL-Dokument entwickelt. Dieses ähnelt der davor vorgestellten WSDL-Beschreibung weitreichend, mit dem Unterschied, dass dieses Web Service zwei Operationen beinhaltet.

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='rms_addbooking'
  xmlns:tns='http://anbieter1.sirka.at/rms_addbooking.wsdl'
  xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'
  xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
  xmlns:wSDL='http://schemas.xmlsoap.org/wsdl/'
  xmlns='http://schemas.xmlsoap.org/wsdl/'>
  <message name='rms_AddBookingRequest'>
    <part name='iduser' type='xsd:integer' />
  </message>
  <message name='rms_AddBookingResponse'>
    <part name='addbookingreturn' type='xsd:string' />
  </message>
  <message name='rms_AddBookingDetailRequest'>
    <part name='idbooking' type='xsd:integer' />
    <part name='idroom' type='xsd:integer' />
    <part name='summary' type='xsd:string' />
    <part name='description' type='xsd:string' />
    <part name='price' type='xsd:integer' />
    <part name='dtstart' type='xsd:string' />
    <part name='dtend' type='xsd:string' />
    <part name='dtstamp' type='xsd:string' />
  </message>
  <message name='rms_AddBookingDetailResponse'>
    <part name='addbookingdetailreturn' type='xsd:string' />
  </message>
  <portType name='rms_addBookingSearchPortType'>
    <operation name='rms_addBooking'>
      <input message='tns:rms_AddBookingRequest' />
      <output message='tns:rms_AddBookingResponse' />
    </operation>
    <operation name='rms_addBookingDetail'>
      <input message='tns:rms_AddBookingDetailRequest' />
```

```

    <output message='tns:rms_AddBookingDetailResponse'/>
  </operation>
</portType>
<binding name='rms_addBookingBinding' type='tns:rms_addBookingSearchPortType'>
  <soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http'/>
  <operation name='rms_addBooking'>
    <soap:operation soapAction=' http://anbieter1.sirka.at /rms_addBooking'/>
    <input>
      <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
    </input>
    <output>
      <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
    </output>
  </operation>
</binding>
<operation name='rms_addBookingDetail'>
  <soap:operation soapAction=' http://anbieter1.sirka.at /rms_addBookingDetail'/>
  <input>
    <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
  </input>
  <output>
    <soap:body use='encoded' encodingStyle='http://schemas.xmlsoap.org/soap/encoding'/>
  </output>
</operation>
</binding>
<service name='rms_addbooking'>
  <port name='rms_addbookingPort' binding='rms_addBookingBinding'>
    <soap:address location=' http://anbieter1.sirka.at /rms_addbook.php'/>
  </port>
</service>
</definitions>

```

Tabelle 20: WSDL - Beschreibung des Web Services zur Buchung

Auch für dieses Web Service wurde natürlich eine PHP-Datei erstellt, über die es verwendbar ist. Die Buchungs-Klasse wurde ebenfalls an ein *SoapServer*-Objekt übergeben, um deren Funktionen im Web Service freizugeben.

Somit bietet das Anbietersystem Web Services zur Suche und zur Buchung von Räumlichkeiten. Diese können natürlich beliebig verwendet werden und müssen nicht nur im Service-System eingesetzt werden. Anbietern bieten sich damit mehrere Möglichkeiten, die Web Services einzusetzen (verschiedene Portale) und somit ihre Räumlichkeiten online zu vertreiben.

Zum Test des Prototypens wurden mehrere Instanzen des Anbietersystems angelegt, um das Service-System entsprechend testen zu können. Dabei handelt es sich immer um den selben Programmcode, jedoch sind der Datenbestand und der Webserver unterschiedlich, um wirklich unabhängige Systeme zu erhalten. In einer vollständigen Umsetzung des Service-Systems wäre es natürlich wünschenswert und sogar notwendig, wenn heterogene Anbietersysteme eingesetzt werden könnten.

4.2.3 Das Service-System

Der Prototyp des Service-Systems umfasst mehrere Funktionen:

- Zum einen können Benutzer hinzugefügt werden und diesen eine bestimmte Rolle im System zugewiesen werden.
- Es können Web Services und deren Operationen erfasst werden und diesen ein bestimmter Operationstyp zugeordnet werden. Dadurch wird das Serviceverzeichnis abgebildet.
- Benutzer können Suchanfragen starten und dabei verschiedene Kriterien auswählen.
- Ebenfalls können Benutzer Ressourcen entsprechend einer Terminauswahl buchen.

Das im Prototyp entwickelte Service-System selbst besteht aus mehreren Relationen, wobei folgende Abbildung das dazugehörige ER-Diagramm zeigt.

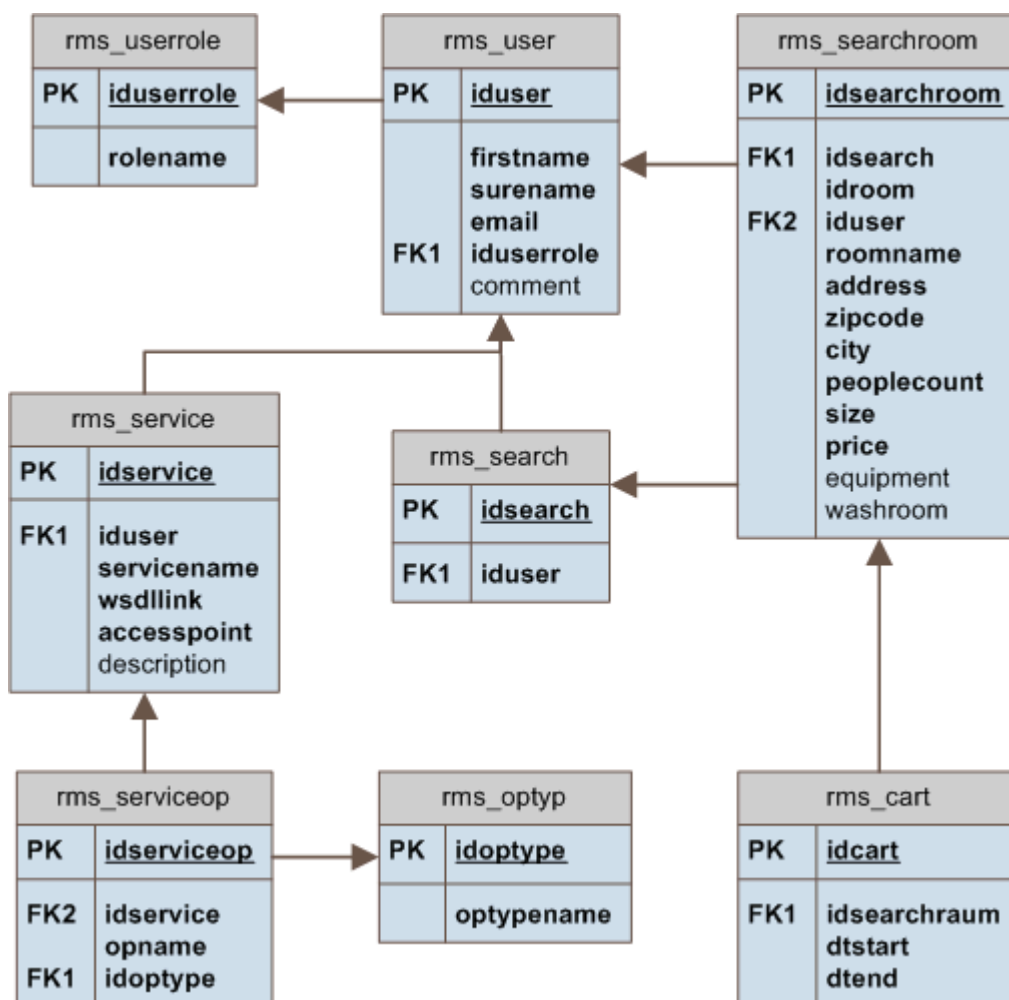


Abbildung 22: ER-Diagramm - Service-System

Die Relation *rms_user* dient dazu, um Benutzer beziehungsweise Entitäten jeglicher Art anzulegen. Das kann der einfache Benutzer sein, der das Service-System verwendet, um eine Räumlichkeit zu finden, aber es kann sich auch um einen Anbieter handeln, der über das Service-System Räumlichkeiten vermieten möchte und seine Web Services deshalb darin eingliedert. Aber auch die Benutzer zur Systemverwaltung werden in der Tabelle erfasst.

An dieser Stelle muss ebenfalls erwähnt werden, dass in einer professionellen Umsetzung mehr Stammdaten zu den Benutzern gespeichert werden müssen. Die Benutzer-Relation kommt natürlich an vielen Stellen des Systems zum Einsatz, da bei jeder Aktion ein Benutzer beteiligt ist und seine Zugehörigkeit hinterlegt werden muss. Jeder Benutzer hat also eine gewisse Rolle im System, auf die in der Relation *rms_userrole* verwiesen wird. Aufgrund der Rolle wird im System entschieden, welche User zu welchen Systemteilen Zugangsberechtigungen haben. Im Prototypen wird aufgrund der eindeutigen Rollen-ID innerhalb der Anwendung entschieden, ob ein Benutzer berechtigt ist oder nicht. Dies sollte erweitert werden, indem in der Relation Berechtigungen zu einzelnen System-Modulen vergeben werden können. Das macht das System leichter wart- und erweiterbar.

Auf der Userverwaltungs-Seite des Systems werden die verschiedenen User in einer Liste angezeigt. In dieser PHP-Datei werden abermals mittels den MySQL-Funktionen die Daten – in diesem Fall die Userdaten – aus der Datenbank ausgelesen. Es können natürlich auch neue Benutzer und Benutzerrollen hinzugefügt werden. Die Eingabe der Daten erfolgt abermals über Web-Formulare, wobei in PHP die Eingabedaten wie immer mithilfe des Superarrays $\$_POST$ abgerufen werden.

Die *rms_service*-Relation dient zur Speicherung der im Service-System integrierten Web Services, wobei ein Web Service einem Benutzer zugeordnet sein muss. Weiters wird zu jedem Web Service ein Servicename als Titel und eine Beschreibung hinterlegt. Wie bereits im Systementwurf geschildert, sollte in einem Livesystem besonders der Servicebeschreibung ein hoher Stellenwert eingeräumt werden. Es sollte verpflichtend ein komplettes Serviceprofil mit dem Web Service selbst eingereicht werden, bei dem eine gewisse Nomenklatur vorgegeben sein sollte, um einheitliche Beschreibungen zu erzielen. Einerseits macht eine manuelle Überprüfung von Web Services vor ihrer Freischaltung Sinn, andererseits wäre es natürlich optimal, wenn dies vollautomatisch durchführbar wäre. Hier sei nochmal besonders das Resource Description Framework und die Beschreibungssprache Web Ontology Language erwähnt, auf die im Kapitel Servicetechnologien eingegangen wurde.

Indem Web Services mehrere Operationen beinhalten können, dient die Relation *rms_serviceop* zur Erfassung dieser. Eine Operation verweist dabei auf ein darüber liegendes Web Service und wird durch einen Namen gekennzeichnet. Weiters wird jeder Operation ein Operationstyp zugeteilt. Operationstypen werden in der Tabelle *rms_optyp* verwaltet. Über den Operationstyp wird im Prototypen die Service-Komposition gesteuert, wobei die Kompositionslogik hauptsächlich im Programmcode verankert ist. Für ein generisches Konzept, bei dem es jederzeit möglich sein soll, ein Service durch ein anderes zu ersetzen, ist dieses Modell etwas zu simpel und müsste deshalb erweitert werden. Im Systementwurf wurden die Grundlagen eines solchen Konzepts vorgestellt. Über die Serviceverwaltung-Seiten können neue Services dazugehörige Operationen und Operationstypen angelegt, bearbeitet oder auch gelöscht werden. Die Operationen müssen dabei einem Operationstyp zugeordnet werden. Über den Operationstyp einer Operation wird bei der Ausführung der Komposition abgerufen und dadurch entschieden, welche Operationen zu welchem Zeitpunkt in der Komposition dran kommen.

Sobald im Service-System über die dazugehörige Webseite eine Suchabfrage gestartet wird, wird dazu in der *rms_search*-Relation ein Datensatz angelegt. Dieser beinhaltet eine eindeutige Such-ID und einen Verweis auf den Benutzer, der die Suche durchgeführt hat. Für die Suche selbst können mehrere Suchbegriffe eingegeben werden. Darunterliegend werden dann in der Relation *rms_searchraum* die Ergebnisse entsprechend den Suchkriterien eingetragen. Dies ist notwendig, da die Ergebnisse aus mehreren Anbieterdatenstämme kommen.

Da durch die Suche nachfolgend die einzelnen Anbieter-Services aufgerufen werden, können nicht mehr Kriterien eingegeben werden, als die diversen Anbieter-Funktionen unterstützen. Im Falle des Prototyps können, wie bereits beim Anbietersystem erwähnt, folgende Kriterien spezifiziert werden:

- die Postleitzahl,
- der Ort,
- eine minimale und maximale Flächenangabe,
- ein minimaler und maximaler Preis,
- sowie eine minimale und maximale Personenanzahl.

Falls man zukünftig verschiedenartige Anbieter-Services unterstützen möchte, könnten aus den einzelnen Funktionen die Parameter ausgelesen werden und daraus das Web-Formular automatisch generiert werden. Anbieter-Funktionen, die dann gewisse Kriterien nicht als Parameter in der Funktion berücksichtigen könnten zwar trotzdem aufgerufen werden, jedoch ohne den fehlenden Parameter zu übergeben. Da kann es unter Umständen natürlich passieren, dass Suchergebnisse angezeigt werden, die nicht genau dem Kriterien-Filter entsprechen. Eine andere Option wäre es, die Funktionen mit den fehlenden Parameter wegzulassen. Der Suchende müsste natürlich darüber informiert werden oder eine Auswahl über die zu verwendete Suchmethode treffen können.

Nach Eingabe der Suchkriterien und Aufruf der Suche startet das Servicesystem mit der Datenkollektion der Anbieter. Dabei wird zuerst das Serviceverzeichnis – also die Relation *rms_service* bzw. *rms_serviceop* – nach Such-Operationen von Anbietern durchsucht, was anhand des Operationstyps entscheidbar ist. Danach werden die Anbieter-Funktionen sequentiell aufgerufen und die eingegebenen Suchkriterien als Funktionsparameter übergeben.

Um die Web Services aufrufen zu können, wird aus dem Serviceverzeichnis der Link zur WSDL-Datei des aktuell bearbeiteten Anbieters ausgelesen. Mithilfe der WSDL-Beschreibung kann anschließend ein *SoapServer*-Objekt angelegt und darüber die dahinter liegende Suchfunktion aufgerufen werden. Die Suchergebnisse aus dem Datenbestand des Anbieters werden vom Web Service zurückgeliefert und in der Tabelle *rms_searchraum* gespeichert, wobei die dazugehörige Such-ID, die Benutzer-ID des Anbieters, die eindeutige ID und die Stammdaten der Räumlichkeit abgelegt werden. So wird ein Anbieter nach dem anderen abgearbeitet und die Suchergebnisse gespeichert.

Die sequentielle Abarbeitung kann jedoch ein großes Problem darstellen und sollte in einer professionellen Systemumsetzung auf keinen Fall fehlen, da es sonst zu unnötigen Wartezeiten kommen könnte. Beispielsweise könnte ein einzelnes Anbieterservice ausfallen oder zu langsam reagieren, was zu einem Timeout führen könnte und somit die komplette Abwicklung aufhält. Um dieses Problem zu vermeiden, sollten alle oder zumindest mehrere Anbieter parallel abgearbeitet werden. Auch hier können natürlich Timeouts auftreten, der Abarbeitungsprozess steht zu diesem Zeitpunkt aber nicht ganz still. Generell sollte es konfigurierbar sein, wie lange auf eine Reaktion eines Web Services gewartet wird. Diese und ähnliche Optionen sollten zentral verwaltbar sein. Zur Durchführung einer parallelen Abarbeitung müssen mehrere Prozesse (Threads) mit der Abarbeitung beauftragt werden. PHP selbst unterstützt in der aktuellen Version nativ kein Threading, weshalb zur Unterstützung ein im Hintergrund laufendes Programm (ein Daemon) zur Abarbeitung von parallelen Abfragen dazu entwickelt werden sollte (beispielsweise Java).

Die Relation *rms_cart* dient zur Umsetzung eines Warenkorbs. Die Tabelle beinhaltet die Start- und Endzeit der Bestellwünsche. Außerdem enthält sie noch einen Verweis auf das jeweilige Suchergebnis. Nachdem alle Anbieter abgearbeitet und deren Suchergebnisse in der Tabelle gespeichert wurden, wird die Ergebnisliste dem Benutzer angezeigt. In dieser Ergebnismaske können einzelne Elemente ausgewählt und für diese dann ein gewünschter Buchungs-Zeitraum angegeben werden. Dabei wird in der Tabelle *rms_cart* ein neuer Datensatz angelegt. Die dabei erstellten Einträge werden in einem Warenkorb angezeigt. Im Warenkorb wird mit der Aktion "Bestellung abschicken" der eigentliche Buchungsvorgang durchgeführt. Dabei werden über die aktuelle Suchabfrage die Bestellwünsche des Benutzers aus *rms_cart* ausgelesen, wobei wieder ein Anbieter nach dem anderen abgearbeitet wird. Im Serviceverzeichnis des Systems werden dabei die zu den einzelnen Anbietern gehörenden Web Service-Beschreibungen zur Buchung ausgelesen. Für die Buchung sind zwei Operationen notwendig. Bei jedem Anbieter, dessen Ressourcen im Warenkorb beteiligt waren, wird mit der einen Operation eine neue Buchung angelegt und mit der anderen die Räumlichkeiten zur Buchung hinzugefügt. Wieder kann aufgrund des Operationstyps eruiert werden, welche Funktion eingesetzt werden muss. Auch hier ist eine parallele der sequentiellen Abwicklung vorzuziehen.

Im Prototypen sind Buchung direkt abhängig von Suchergebnissen. Das heißt, es kann nicht der Warenkorb aus mehreren Suchabfragen gefüllt werden, was bei einer Erweiterung der Benutzbarkeit zugutekommen würde.

Das Service-System wurde weitgehend serviceorientiert umgesetzt. Sprich: Die meisten Funktionen sind auch über Web Services durchführbar. Um das Entwurfparadigma der Serviceorientierung einzuhalten, sollten soweit wie möglich, alle Teile – sofern es sinnvoll erscheint – so realisiert werden, dass sie durchgehend von dessen Vorteilen profitieren können. So beinhaltet das Service-System – wie bereits zu Kapitelbeginn erwähnt – Services zur System- und Kompositionsverwaltung sowie zur Authentifizierung. Auch diese Services wurden ähnlich wie die zuvor beschriebenen gestaltet. Zu allen diesen Services wurden ebenfalls die dazugehörigen WSDL-Dokumente und Klassen erstellt, um die jeweilige Funktionalität umzusetzen.

5. Evaluierung

Dieses Kapitel dient zur rückblickenden Darstellung der innerhalb der Arbeit angewandten wissenschaftlichen Methoden und Verfahrensweisen, mit deren Hilfe die Ergebnisse erzielt wurden. Weiters dient das Kapitel der Bewertung und zur Ergebnis-Darstellung. Dabei wird nochmals auf die Fragestellungen der Diplomarbeit eingegangen und versucht, diese entsprechend zu beantworten. Somit soll ermöglicht werden Rückschlüsse auf die zentralen Ergebnisse der Arbeit durchführen zu können. Anschließend wird erläutert, worauf bei einer vollständigen Umsetzung des Service-Systems noch eingegangen werden muss und wie der Prototyp erweitert werden kann.

Zum einen war die prototypische Umsetzung eines auf Web Services basierenden Service-System das Ziel dieser Diplomarbeit. Diesbezüglich wurde eine Literaturrecherche der für solch eine Umsetzung relevanten Themenbereiche durchgeführt und in einem theoretischen Teil analysiert. Dabei wurde einerseits auf den Service-Begriff selbst eingegangen, andererseits wurden aber auch viele aktuelle Aspekte der Dienstleistungsforschung und von Servicetechnologien behandelt. Weiters wurde näher auf Methoden zur Qualitätssicherung und auf Leistungsbündel eingegangen.

Anschließend wurde mit der Entwurfsphase des Service-Systems begonnen. Dabei wurden Methoden des "Rational Unified Process" (RUP), der als Vorgehensmodell diente, eingesetzt. Dies beinhaltete eine Prozessmodellierung, die Durchführung einer Anforderungsanalyse sowie die Analyse und Design des Prototyps. Darauffolgend wurde eine Softwarearchitektur gewählt und die Prototypentwicklung durchgeführt. Dieser Prototyp wurde mit Test auf Funktionalität geprüft.

Die Hauptfrage in dieser Arbeit war, ob eine serviceorientierte Architektur Innovation unterstützt, weil mit ihrer Hilfe sehr flexibel neue Geschäftsideen und -modelle umgesetzt werden können. Innovation wird durch Erweiterung, Verbesserung oder Tausch von einzelnen Komponenten erreicht. Es muss jedenfalls etwas an einem bestehenden Konzept verändert werden, um Innovation herbeizuführen. Diese Veränderung soll so einfach wie nur möglich gestaltet sein, um sie so rasch, simpel und kostengünstig durchführen und dabei möglichst uneingeschränkt agieren zu können. Die grundlegenden Eigenschaften einer serviceorientierten Architektur, wie die lose Koppelung, die nachrichtenbasierte Interaktion, die dynamische Auffindung, die Portabilität, die Implementierungsneutralität, die Autonomie, die Konfigurierbarkeit, die späte Bindung und das richtlinienbasiertes Verhalten unterstützen diese Forderungen. Durch diese Eigenschaften kann effizient und rasch auf veränderte Anforderungen reagiert werden.

Die erstmalige Implementierung einer SOA innerhalb einer Organisation ist bei ordentlicher Durchführung sicher keine rasche Angelegenheit, aber in späterer Zukunft kann von ihr aufgrund ihrer Eigenschaften profitiert werden. Aufgrund einer SOA können einzelne Services zu Kompositionen zusammengestellt werden. Wie auch im Prototyp ersichtlich war, können diese Kompositionen ein Geschäftsmodell umsetzen. Es gab davor noch keine vergleichbare Technologie, mit der dies so unabhängig vollzogen werden konnte. Zwar gab es auch ähnliche Konzepte (wie z.B. CORBA), diese hatten jedoch in ihrer Verwendung viel mehr Einschränkungen und waren deshalb nicht so flexibel einsetzbar.

Das hauptsächliche Novum ist die Bildung neuer Kompositionen aus bestehenden Diensten, wodurch neue Funktionen kreiert werden. Dies kann durchaus auch automatisiert vollzogen

werden. Bei diesen neuen Gebilden besteht wiederum die Möglichkeit der separaten Vermarktung. Obwohl der Kompositionsersteller, abgesehen von der Zusammenstellung "nichts" getan hat, kann er davon profitieren.

Somit kann behauptet werden, dass sich Web Services als Umsetzungsinstrument einer SOA dazu eignen, Geschäftsmodelle flexibel anzupassen, zu erweitern und damit völlig neue Geschäftsmodelle zu erstellen. Der Prototyp zeigt auch, dass ein Dienstleistungsgeschäft durch ein Web Service implementiert werden kann, beziehungsweise, dass ein Web Service ein Dienstleistungsgeschäft darstellen kann.

Ein Hauptproblem bei dieser Aussage ist, dass Flexibilität in diesem Sinn schwer messbar ist. Es könnte versucht werden, verschiedene Umsetzungen miteinander anhand von Beispielen zu vergleichen und messbare Parameter für den direkten Vergleich zu bestimmen, wobei Sinnhaftigkeit und klare Aussagekraft fraglich sind.

Beispielsweise wäre es auch möglich, das Service-System einem vergleichbaren klassischen System gegenüberzustellen und unterschiedliche Parameter (z.B. Nettozeit bis zur erfolgreichen Buchung, Kosten, Qualität) zu vergleichen, oder eine Befragung der beteiligten Personen durchzuführen. Auch hier ist der Nutzen fraglich, da bei der Parametergegenüberstellung die Aussagekraft der daraus resultierenden Auswertung fragwürdig wäre und eine Personenbefragung aufgrund sehr individuellen und unterschiedlich bewerteten Qualitätskriterien ebenfalls nicht unbedingt zielführend und nur schwer durchführbar wäre.

Um eine Geschäftsidee umzusetzen, ist die bloße Implementierung einer serviceorientierten Architektur jedoch nicht ausreichend. Es bedarf einer Verbindung zwischen SOA und der Geschäftsidee, um diese auch realisieren zu können. Dazu muss die Geschäftsidee erst in einem Geschäftsmodell dargestellt werden und daraus einzelne Geschäftsprozesse identifiziert werden. Diese einzelnen Geschäftsprozesse können wiederum mittels eines oder mehrerer Web Services umgesetzt werden. Dazu benötigt es eine formale Beschreibung der Geschäfts-services. Einzelne Web Services können dann zu Kompositionen zugeteilt werden, um zusammen einen Prozess zu realisieren. Dazu bedarf es einer bestimmten Kompositionslogik, um auch automatisch konfigurierbare Geschäftsprozesse realisieren zu können.

Weiters ist eine Planung von Aktionen notwendig, die dann zu bestimmten Ereignissen durchgeführt werden. Dabei kann es sich um regelmäßige Aktionen handeln, aber auch um welche, die nur von bestimmten "Events" ausgelöst werden. Dies kann mit dem Begriff Scheduling betitelt werden.

Abschließend wird noch detailliert darauf eingegangen, was abseits des Prototyps für eine komplette Umsetzung des Service-Systems berücksichtigt werden muss und welche Möglichkeiten der Erweiterung bestehen.

Das Service-System wurde in dieser Arbeit nur prototypisch umgesetzt; es sollten bei einer vollständigen Umsetzung beziehungsweise bei einem kommerziellen Einsatz noch einige andere Thematiken vertieft berücksichtigt werden, auf welche nun in diesem Kapitel hingewiesen wird.

Dabei handelt es sich um Anforderungen aus nicht nur rein technischer Sicht, sondern aus einer betriebswirtschaftlichen beziehungsweise rechtlichen Perspektive. Wir bewegen uns zwar in einem globalen Markt, aber es gibt jedoch in den meisten Ländern unterschiedliche Rechtslagen, gerade was Fernabsatzgeschäfte betrifft.

Auch auf kulturelle und politische Anforderungen sollte genau eingegangen werden. Dienste und Kompositionen sollten beispielsweise mehrsprachig zur Verfügung gestellt werden. Regionstypische Regelungen, Vorgehensweisen und Sitten können einen großen Einfluss auf einen Geschäftsablauf haben. Weiters sollte auch nicht auf die Vermarktung des Systems vergessen und auf Maßnahmen zur Kundenbindung eingegangen werden.

Wie bereits im Kapitel *Methoden zur Qualitätssicherung von Services* ist ebenfalls hier zu beachten, dass beim Online-Handel ganz andere Kriterien als beim Offline-Einkauf eine tragende Rolle spielen. Die Autoren Zeithaml, Parasuraman und Malhotra liefern in ihrem Artikel *Service Quality Delivery Through Web Sites: A Critical Review of Extant Knowledge* [ZePM02] einen guten Überblick und Vergleich dazu.

5.1 Zusätzliche funktionale Anforderungen

Die Funktionalität des Service-Systems kann zwar durch Einbindung zusätzlicher Services theoretisch beliebig erweitert werden, aber trotzdem möchte ich hier noch ein paar grundlegende, im Prototyp fehlende Funktionen, ausdrücklich erwähnen.

5.1.1 Automatische Portfolioerstellung

Der Benutzer soll seinen Nachfragewünschen entsprechend ein Suchprofil mit mehreren Kriterien erstellen können. Das System erstellt daraufhin mehrere, möglichst optimale, Portfolios aus Ressourcen verschiedener (oder unter Umständen von einem einzigen) Anbieter und liefert dem Suchenden diese möglichst effiziente Kombination als Suchergebnis zurück. Suche und Berechnung solcher Portfolios kann sehr zeit- und rechenintensiv sein. Ein Dienst zur verteilten Berechnung oder eine spätere Benachrichtigung über das Ergebnis könnten hier sinnvoll sein.

So könnte ein Konferenzorganisator beispielsweise drei Räumlichkeiten am selben Tag mit möglichst minimaler Distanz zwischen den Räumlichkeiten und einem möglichst günstigen Preis suchen. Gewisse Ausstattungskriterien könnten eventuell auch noch in die Suchkriterien einfließen und es könnte für Kriterien noch eine Gewichtung vergeben werden. Für das optimale Portfolio müssten alle möglichen Ressourcen miteinander kombiniert und verglichen werden. Dies ist ein Problem der Komplexitätsklasse NP-vollständig und somit nichtdeterministisch in polynomialer Zeit lösbar, da ein deterministischer Algorithmus exponentiellen Rechenaufwand erfordert. Autor Dorn schlägt dafür in seinem Artikel *Service Scheduling* als mögliche Lösungsvariante "Tabu Search" vor, um möglichst effiziente Portfolios in annehmbarer Zeit zu generieren (siehe dazu [Dorn09]).

5.1.2 Zusätzliche Servicebeschreibungen

Um allen Nutzern des Systems mehr Aufschluss und Information zu den Services und den dahinter stehenden Ressourcen zu liefern, gilt es innovative Beschreibungsmöglichkeiten zu entwerfen. Trotz einheitlicher Darstellung und Standardisierung von Werten sind Ressourcenmerkmale nicht immer eindeutig darstellbar, da eine einheitliche Beschreibung diverser Details und Präferenzen (z.B. Qualität) nicht leicht umsetzbar ist. Ziel ist es den Konsumenten so viel wie möglich Information zu den technischen Services sowie auch zu den Dienstleistungen selbst im Vorhinein zu liefern.

Trotzdem gilt es, den Nutzern so viel Information wie nur möglich gegliedert aufzubereiten. Dies macht aber auch nur dann Sinn, wenn der einzelne Nutzer schnell und einfach genau die Information findet, nach der er sucht. Eine reine Informationsflut würde zu einer Überladung

führen. Beschreibungen können auch mit multimedialen Inhalten, wie z.B. Bildern und Videos, erweitert werden. Zusätzliche Serviceinformationen könnten auch von den Konsumenten in Form von Feedback eingefügt werden, welche beispielsweise Aufschluss über die Qualität, das Verhalten oder die Benutzung des Services geben.

Ein potenzieller Nutzer soll bei der Suche nach dem geeigneten Web Service im Verzeichnis zusätzliche Suchkriterien Anforderungen an die Qualität des Web Services angeben können, wobei nur die Web Services als Suchergebnis zurückgegeben werden, die auch die Qualitätskriterien erfüllen. Um diese Funktionalität zu verwirklichen, muss ein SLA in einer maschinenlesbaren Form vorliegen; es muss einen Mechanismus geben, der SLAs in die Suche einbezieht.

5.1.3 Erweitertes Kompositionsmodell

Wie bereits erwähnt, wurde im Prototyp ein recht simples Kompositionsmodell umgesetzt, welches auch hauptsächlich über den Programmcode koordiniert wird. Ein erweitertes Kompositionsmodell ist die Basis für mehr Flexibilität im Service-System. Wünschenswert wäre ein Modell, wo Service-Kompositionen angelegt werden und diese ohne zusätzliche Programmierung eingesetzt werden können.

Zusätzlich zum Prototypen wäre eine Unterteilung von Einzelservices und Servicegruppen praktisch. Im Prototypen wird beispielsweise zwischen Service-Operationen zur Buchung und Suche von Ressourcen verwendet, wobei es sich um eine Servicegruppe handelt, die unabhängig von Operationsergebnissen ausgeführt wird. In einer Komposition soll aber auch, wie im Entwurf modelliert aufgrund von Operationsergebnissen unterschieden werden können, wie der Koordinator die Komposition fortsetzt. Beispielsweise kann ein Wertebereich zur anschließenden Ausführung der Operation A und ein zweiter Wertebereich zur Ausführung der Operation B führen. Diese Entscheidung ist für einfache Datentypen noch relativ unkompliziert. Bei komplexen Datentypen ist ein direkter Vergleich nicht mehr so einfach. Man könnte beispielsweise für Services eines gewissen Typs ein Datenformat vorgeben und daraus dann ein Modell für Kriterien bestimmen. Dadurch beschränkt man allerdings wiederum die Flexibilität. Deshalb gilt es, für die jeweilige Einsatzumgebung ein adäquates Modell zu finden.

Die Entwicklung und Verwendung von Standards spielt sowohl bei Web Services, als auch beim kompletten Service-System eine wesentliche Rolle. Der reguläre und problemlose Nachrichtenaustausch sowie die automatisierte Weiterverarbeitung der Informationen sind nur durch ein einheitliches und vordefiniertes Datenformat gewährleistet.

Web-Services setzen wie bereits mehrfach erwähnt und beschrieben mitunter auf die Standards XML, SOAP und WSDL. Gerade für die Modellierung von Kompositionen sollte ein ähnlich detailliertes Regelwerk vorliegen, um eine dynamische und reibungslose Funktionalität des Service-Systems sicherzustellen.

5.1.4 Eingliederung bestehender Web Services zur Ressourcenbuchung

Im Prototyp werden nur eigens an das Service-System angepasste Web Services eingesetzt.

Den Anbietern könnte es ermöglicht werden, ihre bestehenden Web Services zur Ressourcenbuchung in das Service-System zu integrieren, um damit auf den eigenen Datenbestand zuzugreifen und Ressourcen am Markt anzubieten. Dabei bietet sich auch die Anwendung von Vokabularen an, um die einzelnen Servicefunktionen und dessen Parameter zu

beschreiben. So könnte beispielsweise ein Vokabular für Suchkriterien eingeführt und ein Mapping von Kriterien auf Funktions-Parametern umgesetzt werden.

Standardisierte und häufig wiederverwendbare Services könnten beispielsweise Services für Karten, Adress- und Routenberechnungen, Services für den Versand von E-Mails, Services zur Bildaufbereitung und Services für die Zahlungsabwicklung sein.

Im Bezug auf eine Konferenzorganisation könnte es zusätzlich Services zur Einladung, Anmeldung, Benachrichtigung und Unterbringung (Hotelbuchung) von Teilnehmern geben.

Es handelt sich dabei um Services, die in vielen unterschiedlichen Kompositionen häufig zum Einsatz kommen könnten und ein Ausfall besonders nachteilig wäre, weshalb solche Services mit besondere Aufmerksamkeit betrachtet werden sollten.

5.1.5 Vokabulare

Vokabulare sollen dazu dienen, Services besser beschreib- und auffindbar zu machen. Dadurch soll es auch ermöglicht werden, Servicebeschreibungen für Maschinen "lesbar" zu machen. Servicemerkmale, -kontexte, -schlüsselwörter und andere Formen von Metainformationen müssen unbedingt konsistent bezeichnet und klassifiziert werden. Wenn unterschiedliche Entitäten bzw. Personen verschiedene Konventionen benutzen, kann dies die Kompositionsfähigkeit von Services nachhaltig schädigen und die Verwaltung der Serviceinventare erschweren. [Erl08]

Dies gilt sowohl für maschinen- als auch für menschenlesbare Inhalte, wobei vor allem auf Konsistenz und breite Verfügbarkeit der offiziellen Vokabulare für alle Service-Entwickler geachtet werden muss.

Zumindest folgende Begriffe sollten für Vokabulare standardisiert werden:

- Begriffe des serviceorientierten Computings
- Begriffe zur Klassifikation von Services
- Begriffe für Merkmalstypen, Kategorien, Bezeichnungen
- Schlüsselwörter für Serviceprofile

5.2 Bestehende Lösungen

Es existieren bereits viele verschiedene, auch teilweise freie Softwarelösungen (sog. Frameworks), die Bibliotheken für den Umgang mit Web Services, zur Fehlerbehandlung, zur Vereinfachung von Datenbankzugriffen und ähnliches anbieten. Dadurch werden den Entwicklern schon viele Arbeitsvorgänge abgenommen und erleichtert, weshalb oft nicht selbst programmiert werden muss, sondern fertig übernommen werden kann. Dies kann häufig zu einer Kostenreduktion bei der Entwicklung führen. Weiters werden diese Frameworks häufig überarbeitet und man kann auf aktuelle Anpassungen zurückgreifen.

Gerade im Bezug auf Sicherheitsfragen ist eine gründliche Analyse und Begutachtung der unterschiedlichen Frameworks vor dessen Einsatz notwendig. Im Vorhinein muss bewertet werden, ob sich ein Framework für den Einsatz in der jeweiligen Umgebung anbietet oder nicht. Wichtig ist jedoch aber auch, soweit wie möglich, unabhängig zu bleiben, und das etwaige Umstellungen und Anpassungen trotzdem realisierbar bleiben.

Für PHP-Frameworks können beispielsweise folgende genannt werden, die alle das "Model View Controller"-Architekturmuster (MVC) anwenden:

- Das weit verbreitete Zend-Framework³⁴
- Das CakePHP-Framework³⁵
- Das Open Source Framework Symfony³⁶

5.3 Standardisierung

Standardisierungen sind notwendig, um den breiten Einsatz eines solchen Systems überhaupt zu ermöglichen. Es sollten unter anderem für Kategorien, Funktionen, Dokumentationen und Services gewisse Richtlinien aufgestellt werden, um einerseits Ordnung im System zu halten und andererseits auch in gewisser Weise vor Missbrauch zu schützen.

Beispielsweise kann der WS-Policy-Standard dazu verwendet werden, um Serviceanbietern die Möglichkeit zu geben, Richtlinien und Informationen bezüglich Sicherheit, Qualität und Version des Services in Form von maschinenlesbaren XML-Daten für den Servicenutzer bereitzustellen.

Durch die Standardisierung soll jedoch die Spezialisierung gewisser Dienste nicht verhindert werden. Qualitative Merkmale der Ressourcen eines Anbieters sollen bemerk- und darstellbar sein, da sonst die Tendenz steigt, nur Vergleiche auf Preisebene durchzuführen. So müssen Dienste anpassbar sein, um die verschiedensten Ressourcenkriterien berücksichtigen zu können.

5.4 Nichtfunktionale Anforderungen

Abgesehen von funktionalen Anforderungen gibt es auch eine Menge nichtfunktionaler Anforderungen, die bei einer kompletten Umsetzung des Systems berücksichtigt werden müssen.

Darunter fallen, unter anderem, Themen wie Sicherheitsanforderungen, Usability, Zuverlässigkeit, Leistung und Effizienz, Wartbarkeit und Änderbarkeit.

5.4.1 Abstraktion

Web Services bieten aufgrund ihrer Plattformunabhängigkeit eine hohe Abstraktion, weshalb dem Entwickler dadurch mehr Freiheiten und Möglichkeiten für die Weiterentwicklung offen bleiben. So kann der Entwickler beispielsweise aufgrund funktionaler Vorteile die Programmiersprache wechseln, ohne dass der Endkunde davon erfährt und vertragliche Voraussetzungen verletzt werden. Der Benutzer erhält Informationen, die zur Ausführung und Interaktion des Programms notwendig sind; weitere dahinter stehende Informationen bleiben dem Endkunden aber verborgen oder sind "verschleiert" (z.B. SOAP-Bibliothek, Datenbankumgebung, Programmiersprache und System-Ressourcen, die zur Ausführung benötigt werden).

Die funktionale Abstraktion bestimmt, welche Fähigkeiten eines Programms in dessen Beschreibung veröffentlicht werden. Bei der Abstraktion der Programmlogik muss entschieden

³⁴ Zend Framework – Komponenten-orientiertes Framework für PHP 5. [ZeTe10]

³⁵ CakePHP Framework – Quelloffenes Web-Framework für PHP 4 und 5. [CaSF10]

³⁶ Symfony Framework – Open-Source PHP Web Framework. [Symf10]

werden, welche internen Details des Programms absichtlich vor der Außenwelt verborgen bleiben. Gründe dafür können vielseitiger Natur sein - Algorithmen, Funktionen zur Ausnahmebehandlung oder Protokollroutinen werden beispielsweise meist nicht veröffentlicht. Auch bei der Servicequalität ist Abstraktion üblich. Es wird beispielsweise im Servicevertrag vereinbart, dass ein Service zwanzig Stunden am Tag verfügbar sein muss. Es ist aber unbestimmt, was in der restlichen Zeit mit dem Service passiert. Es könnte beispielsweise die Zeit für Wartungen oder auch für Fehlerbehandlung verwendet werden.

5.4.2 Usability

Die Benutzbarkeit, das Aussehen und die Handhabung sind entscheidend, um bei den Nutzern Akzeptanz zu erlangen. Wenn Nutzer eine Anwendung nur schwer bedienen können, werden sie ihr gegenüber keine positive Haltung einnehmen und diese nicht gerne wieder verwenden. Somit ist bei der Konzeption auch auf die Einhaltung diverser Usability-Richtlinien zu achten. Dabei sind die Anwendungen auf Verständlichkeit, Erlernbarkeit und Bedienbarkeit zu prüfen. Es empfiehlt sich beispielsweise, eine Aufstellung und Kontrolle der Einhaltung von Entwurfstandards für grafische Benutzeroberflächen und Services durchzuführen.

5.4.3 Serviceverträge und Servicequalität

Um einen Betrieb des Webservice-Systems auf Dauer gewährleisten zu können, ist es sowohl aus technischer Sicht als auch aus nichttechnischer Sicht notwendig standardisierte Serviceverträge zu vereinbaren.

Darin sollten mitunter Regelungen über Zuverlässigkeit, Leistung und Effizienz, Wartbarkeit, Portierbarkeit und Übertragbarkeit des Service-Systems und dessen Services getroffen werden. Die Art, wie Serviceverträge entworfen werden, nimmt unmittelbar Einfluss darauf, in welchem Maße die Serviceprinzipien umgesetzt werden können. In Serviceverträgen können aus technischer Sicht beispielsweise technische Protokolle, Syntaxen und Normen definiert sein während aus nichttechnischer Sicht Wartungszyklen, Verfügbarkeiten, Zahlungsmodalitäten oder Konditionen definiert sein könnten. Auf weitere Kriterien wurde bereits im Kapitel *Methoden zur Qualitätssicherung* eingegangen.

Somit ist es ratsam, die Serviceverträge um Service Level Agreements zu erweitern, damit verbindliche Aussagen zu gewissen Parametern getroffen werden können beziehungsweise die Konsequenzen gezogen werden, falls die Verträge nicht eingehalten werden. Die Servicequalität (QoS) wird dadurch direkt beeinflusst und festgelegt.

Ein Dienst kann dann als erfolgreich eingestuft werden, wenn er regelmäßig und in verschiedenen Kompositionen eingesetzt werden kann. Um dies langfristig zu ermöglichen, sollte die Verarbeitungslogik des Services im Laufe der Zeit optimiert werden, um möglichst viele Kompositionen zu unterstützen und den Ressourcenbedarf zu minimieren.

Je komplexer eine Servicekomposition ist, umso mehr Daten müssen während dessen Ausführung gespeichert und verwaltet werden. Dies könnte die Systemressourcen des Servicebetreibers massiv belasten, falls zu viele Instanzen eines Services gleichzeitig laufen sollten. Bei der aktiven Speicherung von Zustandsinformation am Server kann dies bei vielen simultanen Zugriffen zu hoher CPU- und Arbeitsspeicherauslastung führen. Es besteht somit das Risiko, dass es bei stark genutzten Services unerwartet zu Engpässen kommen könnte. [Erl08]

Deshalb ist es wichtig, die Dienstqualität soweit wie möglich sicherzustellen. Nutzer von Web Services müssen folglich eine Möglichkeit haben, die Qualität eines Web Services vor der

Nutzung in Erfahrung bringen, zu evaluieren und mit anderen Angeboten vergleichen zu können.

Serviceausfall

Services, die stark und häufig wiederverwendet werden, können sowohl für den Servicebetreiber als auch für die Servicekonsumenten ein bedeutendes Risiko darstellen, da sie einen sogenannten "Single Point of Failure" darstellen und somit alle Kompositionen, an denen sie beteiligt sind, gefährden. Es gibt mehrere Möglichkeiten, dem Problem entgegenzuwirken: Stark frequentierte Services könnten redundant (also mehrfach) abgebildet werden. Dies könnte die Performance des Services für alle Beteiligten verbessern, da die Lasten auf alle redundanten Teile verteilt werden könnten sowie beim Ausfall einer Serviceabbildung können die redundanten Teile die folgenden Anfragen beantworten, und zwar solange bis das ausgefallene Serviceabbild wieder aktiv und einsatzfähig ist.

Im Fall eines fehlerhaften Ablaufs einer Service-Komposition ist eine komplette Beendigung der Ausführung keine gute allgemeine Lösung. Der Ausführungsumgebung sollte es möglich sein, ein neues oder anderes Service auszuwählen, um die Komposition reorganisieren zu können. So kann für das ausgefallene Service ein geeigneter Ersatz verwendet werden, oder es muss die Komposition überarbeitet und mehrere Services getauscht werden - falls es für das ausgefallene Service keinen optimalen Ersatz gibt. Dies wird natürlich nicht immer möglich sein.

Zustände

Während der Laufzeit kann ein Programm (Service) mehrere Zustände durchlaufen oder einnehmen, wobei folgende Arten von Zustandsbedingungen und -daten vorhanden sein können: [Erl08]

- Aktive und passive Zustände
- Zustandsbehaftete und zustandslose Bedingungen
- Zustandsdaten über Kontext, Sitzung und Geschäft
- Kontextdaten und Kontextregeln

Die Zustandsverwaltung spielt im Rahmen der Serviceorientierung ebenfalls eine wichtige Rolle, wobei es verschiedene Möglichkeiten gibt damit umzugehen. Bei auftretenden Engpässen kann sie angewendet werden, wo Services redundant abgebildet werden könnten und eine andere Instanz des Services mit den Zustandsdaten einer anderen weiterarbeiten, um die Ressourcenknappheit auf einem Server auszugleichen. Zusätzlich macht es Sinn, Zustände während der Verarbeitung zu sichern (z.B. in einer Datenbank), um bei möglichen Abstürzen und Ausfällen an derselben Stelle fortsetzen zu können.

Eine andere Variante, um möglichen Ausfällen entgegenzuwirken, ist es in der Service-Komposition Ersatzkomponenten für einzelne Services zu definieren - falls diese vorhanden sind. Dies wird jedoch bei sehr speziellen bzw. einzigartigen Services nur schwer gelingen. Sollte ein Service während dessen Ausführung abstürzen, kann es von redundanten Abbildungen nur dann weitergeführt werden, wenn der letzte Ausführungszustand vorhanden ist. Als Folge eines Absturzes könnte es notwendig sein, nochmals die komplette Komposition auszuführen, da Ergebnisse der Zwischenschritte womöglich nicht mehr verfügbar sind.

5.4.4 Sicherheit & Kontrolle im System

Die Berücksichtigung von Sicherheitsanforderungen ist in jedem System unablässig. Themen wie Vertraulichkeit, Datenintegrität, Verfügbarkeit und Datenschutz müssen unbedingt behandelt werden. Gerade in einem kommerziellen Umfeld sind alle Beteiligten so gut wie möglich vor jeglichem Missbrauch zu schützen. Ein System wird von den Benutzern erst dann regelmäßig benutzt, wenn diese darin vertrauen sowie Missbrauch und Betrug weitgehend ausgeschlossen werden können.

Deshalb müssen Sicherheitsanforderungen eines Services bereits beim Entwurf bedacht werden, da potenzielle User das Service ohne entsprechende Sicherheitsbestimmungen womöglich nicht benutzen würden. Nur wenn die Zuverlässigkeit von Web Services verbindlich zugesichert wird, werden Anbieter und Kunden bereit sein, diese einzusetzen. Trotz verteilter Architektur gilt es die Durchsetzung einer gemeinsamen Sicherheitsrichtlinie zu erwirken.

Grundsätzlich sollen die Anwender und Anbieter so viel Freiheiten wie nur möglich bei der Gestaltung von Diensten haben, jedoch muss jederzeit die Sicherheit gewährleistet werden können. Die Überprüfung der Sicherheitsrichtlinien kann einen riesigen Aufwand darstellen und darf nicht überhand nehmen, weshalb auch hier eine möglichst weitreichende Automatisierung wünschenswert ist.

Folgende Punkte sollen berücksichtigt werden: ([MeSt05] et al.)

Integrität	Daten, Informationen und Systeme müssen vollständig und richtig zur Verfügung stehen. Sie müssen vor nicht autorisiertem Zugriff, Änderungen und Manipulationen geschützt sein; Datenkonsistenz muss garantiert werden können. Eine versandte Nachricht muss ohne Modifikation beim Empfänger eintreffen. Auch hier ist der Einsatz von Sicherheitszertifikaten angebracht.
Verfügbarkeit	Unternehmen müssen nötige Hardware, Software, Daten und Informationen dauerhaft bereitstellen, um den normalen Geschäftsbetrieb aufrechterhalten zu können.
Vertraulichkeit	Der Zugang zu den Daten muss unberechtigten Dritten verwehrt bleiben. Daten dürfen nicht weitergegeben werden, ohne dass eine Berechtigung für die Weitergabe vorliegt. Die Vertraulichkeit bei der Übermittlung von Daten muss gewährleistet sein. Dies betrifft insbesondere die Kommunikation über öffentliche Netzwerke. Eine Nachricht soll ausschließlich für den jeweils adressierten Empfänger lesbar sein. Dies ist beispielsweise durch den Einsatz von Verschlüsselungsmechanismen realisierbar.
Authentizität	Ein Geschäftsvorfall muss einem Verursacher eindeutig zuzuordnen sein. Bei der Kommunikation in SOAs kann dies beispielsweise durch die Verwendung elektronischer Signaturen geschehen.
Autorisierung	Nur explizit berechtigte Personen sollen dem IT-System in vordefiniertem Umfang interagieren dürfen. Das garantiert die Integrität der Daten und des IT-Systems als Ganzes. Ein Dienst-Anfordernder muss auch zur Nutzung des angeforderten Dienstes berechtigt sein (Zugangsüberprüfung).
Verbindlichkeit	Verbindlichkeit bedeutet in diesem Zusammenhang "gewollte Rechtsfolgen bindend herbeizuführen". Verbindlichkeit ist die Basis jeglicher Transaktion. Kann sie nicht hergestellt werden, kann nicht verlässlich gehandelt werden. Der Sender soll im Nachhinein die Urheberschaft einer durch ihn erstellten Nachricht nicht leugnen können.
Glaubwürdigkeit	Eine Nachricht muss nachprüfbar durch den vermeintlichen Sender erstellt worden sein. Insbesondere darf es einem potenziellen Eindringling nicht möglich sein, eine andere Identität vorzutäuschen.

Tabelle 21: Sicherheitsrelevante Themen

Wie bei den meisten anderen verteilten Computersystemen können auch bei Web Services Sicherheitsansätze auf allen Schichten des OSI-Schichtenmodells angewandt werden. Bekanntester Ansatz bei Transportsicherheit ist TLS³⁷, eine Weiterentwicklung des SSL³⁸. Wer die Nachricht geschickt hat, kann ohne zusätzliche Vorkehrungen nicht festgestellt werden. Für eine Kommunikation zwischen genau zwei Partnern, wie z.B. einer Bank und einem Kunden beim Online-Banking, ist TLS eine sehr gute Lösung. Bei einer echt verteilten Anwendung mit einigen Beteiligten, wie sie in einer SOA realisiert wird, ist TLS als Sicherheitslösung alleine

³⁷ TLS – Transport Layer Security

³⁸ SSL – Secure Socket Layers

ungenügend. Für die Gewährleistung von Sicherheit, genauer der Vertraulichkeit, ist es notwendig, dass Teile einer Sendung verschieden behandelt werden können und somit zum Beispiel ein Beteiligter nur einen Teil lesen kann. In komplexen Web-Services-Szenarien ist es also nicht möglich, eine allgemeine Lösung nur auf Basis von Transportsicherheit zu entwickeln. Um dies zu erreichen, werden Methoden der Nachrichtensicherheit eingesetzt. [Melz08]

Aufgrund der Tatsache, dass die Daten im XML-Format übertragen werden, ergibt sich die Möglichkeit der feingranularen Streuung der Sicherheit, das heißt einzelne Dokumentteile können separat mit Sicherheitsmerkmalen versehen werden. [Melz08]

Der Standard WS-Security³⁹ wurde von Microsoft, IBM und Verisign entwickelt, um eine Möglichkeit zu bieten, verschiedene bereits bestehende Sicherheitstechnologien im Web-Services-Umfeld standardisiert nutzbar zu machen, ohne dabei jedoch künftige Entwicklungen zu blockieren. Hierbei wird SOAP so erweitert, dass bereits bestehende Techniken, wie zum Beispiel XML-Signatur und -Verschlüsselung, Public-Key-Infrastrukturen oder auch die Security Assertion Markup Language (SAML), eingesetzt werden können. Es werden also keine neuen Vorgehensweisen entwickelt, sondern lediglich der standardisierte Einsatz bewährter Verfahren vorgeschlagen. [Melz08]

Weiters soll WS-Security als Basis für alle weiteren Sicherheitsstandards für Web Services genutzt werden können. WS-Policy⁴⁰ ermöglicht die Formulierung von Sicherheitsanforderungen und Richtlinien, die erfüllt werden müssen, um mit einem Dienst interagieren zu können. Zu diesem Zweck wird ein erweiterbares Rahmenwerk definiert, mit dem bevorzugt, aber nicht notwendigerweise, einfache Anforderungen beschrieben werden können. Typische Beispiele hierfür sind die Wahl von Transportprotokollen oder Verfahren zur Authentifizierung. Durch die Nutzung der in WS-Trust⁴¹ definierten Erweiterungen wird es ermöglicht eine sichere Kommunikation im Kontext von Web Services durchzuführen. [Melz08]

Verteilte Eigentumsverhältnisse von Services machen die Herausforderung und deren Handhabung noch viel schwieriger. Eine Veränderung eines einzelnen Services durch den Serviceeigentümer könnte verschiedenste Kompositionen nutzlos machen. Generell ist es schwer, Schlussfolgerungen über das System in seiner Gänze zu machen, wenn sich die Umgebung in einer unvorhersehbaren Art und Weise ändert. Diesbezüglich müssen Regeln aufgestellt werden, wie und wann Services verändert werden dürfen, oder wer bei einer Änderung zu kontaktieren ist.

Deshalb sollte vor Ausführung einer Komposition eine Überprüfung (z.B. mittels Regressionstests) der einzelnen Services auf Veränderung durchgeführt werden. Dies wiederum erfordert Methoden zur Analyse von Web Services.

Es gibt eine Fülle möglicher Angriffsvarianten, wobei die Angreifer mit verschiedenen Methoden versuchen, unberechtigt Zugang zum System oder zu vertraulichen Informationen zu gelangen. Es muss vorab versucht werden, potenzielle Sicherheitslücken⁴² zu schließen und bereits bei der Entwicklung von Services darauf zu achten, sich vor Missbrauch zu schützen.

³⁹ WS-Security – Ein Standard der OASIS Organisation [OAS10]

⁴⁰ WS-Policy – Eine Erweiterung von WS-Security

⁴¹ WS-Trust – Ebenfalls eine Erweiterung von WS-Security

⁴² Sicherheitslücken – eine umfangreiche Liste ist auf Wikipedia zu finden, wobei dem Thema sogar eine ganze Kategorie gewidmet wurde. [Wiki10]

Auch Angriffe mittels "Denial of Service"-Attacken (DoS), wobei versucht wird Dienste arbeitsunfähig zu machen und diese zu überlasten, sind zu beachten.

5.4.5 Fehlerbehandlung

Im Prototyp werden Laufzeit- und Eingabefehler nicht abgefangen und behandelt, worauf in einer vollendeten Version natürlich verzichtet werden kann, da es sonst, unter anderem, zu inkonsistenten Datenbeständen, Sicherheitslücken im System und ungewollten Systemabstürzen kommen könnte.

PHP bietet, wie andere Programmiersprachen auch, einige Methoden zur Behandlung von Fehlern, zur Fehlersuche, zum Debugging und zum Reporting von Fehlern. Auch in PHP kann ein Ausnahmefehler (eine sogenannte Exception) geworfen (throw) und abgefangen (catch) werden.

Um mögliche Exceptions abfangen zu können, muss der betroffene Code von einem *try*-Block umgeben sein, wobei jedem *try*-Block mindestens ein *catch*-Block zugeordnet sein muss, in dem der aufgetretene Fehler dann behandelt werden kann. Es gibt auch unterschiedliche Fehlertypen, für die auch jeweilig mehrere *catch*-Blöcke angelegt werden können. So kann beispielsweise ein *catch*-Block für eine gewisse Fehlerklasse angelegt werden und ein weiterer, der in allen anderen auftretenden Fehlerfällen eingesetzt wird.

Im sogenannten Error-Reporting von PHP kann bestimmt werden, welche PHP-Fehlermeldungen angezeigt werden. Grundsätzlich sollten dem Benutzer keine detaillierten Fehlerbeschreibungen zurückgeliefert werden, sondern lediglich einheitliche Rückmeldungen, dass es bei der Ausführung zu Fehlern gekommen ist. Dies steigert einerseits die Systemsicherheit, da potenzielle Angreifer durch eventuell absichtlich hervorgerufene Fehler keine heiklen Systeminformationen auslesen können. Andererseits wird dadurch auch die Benutzbarkeit des Systems verbessert. Für die Systemverwaltung ist es jedoch natürlich äußerst wichtig alle auftretenden Fehler zu dokumentieren und dadurch nachvollziehen zu können, um entsprechend darauf reagieren zu können.

Auch für Web Services hat PHP über das Objekt *SoapClient* mehrere Methoden zur Überprüfung des SOAP-Nachrichtenverkehrs für die Entwickler parat. Die Funktion *getLastRequest* liefert die Daten der letzten SOAP-Anfrage in XML-Form zurück. Als Gegenstück liefert die Funktion *getLastResponse* die Daten der letzten SOAP-Antwort des aufgerufenen Web Services zurück. Mit den Funktionen *getLastRequestHeaders* und *getLastResponseHeaders* können wiederum die Header der letzten Anfrage bzw. Antwort einer SOAP-Nachricht abgerufen werden. Falls beim Aufruf eines Web Services ein Fehler auftritt, wird eine Exception vom Typ *SoapFault* geworfen, wobei natürlich auch die mittels *try* und *catch* abgefangen werden können.

Besonders in der Entwicklungsphase eines Web Services ist zu beachten, dass PHP einen WSDL-Cache besitzt. Falls man also öfters Änderungen an einem WSDL-Dokument vollziehen muss, sollte dieser Cache deaktiviert sein, da sonst die WSDL-Datei nicht bei jedem Aufruf neu analysiert wird. In einer Produktionsumgebung ist es wiederum leistungsfördernd, wenn diese Option aktiv gesetzt ist.

5.5 Methoden zur Überprüfung von Web Services

Um potenzielle Fehler verhindern und Sicherheitsrichtlinien umsetzen zu können, sollten Web Services mehrfach überprüft werden.

Diesbezüglich gibt es verschiedene Ansätze: [BaNi07]

- Analysetechniken: Werden an einem formalen Modell eines Services oder einer Servicekomposition angewendet.
- Test-Techniken: Unit-Testing von Servicekompositionen bis Regression-Testing von einzelnen Services und Kompositionen.
- Monitoring-Techniken: Zur ständigen Überprüfung von Kompositionen zur Laufzeit.

5.5.1 Analysetechniken

Die BPEL⁴³ ist der De-facto-Standard für die Implementierung von Prozessen mit Web Services, womit versucht wird das Verhalten von Web Services zu analysieren.

Es werden statische und dynamische Analyseverfahren eingesetzt, um beispielsweise die Verwundbarkeit von Web-basierten Systemen zu bestimmen und Modelle (z.B. Petri-Netze) zur Vorhersehbarkeit von Web-Service-Verlässlichkeit zu erstellen. Ein gebräuchliches Vorgehen dabei ist es, die Prozesse in einzelne Teile zu partitionieren und diese dann separat zu untersuchen.

5.5.2 Testtechniken

Tests werden ebenfalls verwendet, um Aufschluss über Web Services zu geben und es werden beispielsweise nach der Entdeckung eines Services darauf angewendet, um weitere Details über dessen Verhalten zu erfahren.

Es gibt unterschiedliche Test-Methoden, die eingesetzt werden können: [BaNi07]

- Unit Testing: Dabei handelt es sich um Tests, bei denen eine einzelne Klasse oder Routine getestet wird.
- Integration Tests: Die verbunden oder integrierten Komponenten in einem Softwaresystem werden getestet, um Fehler zu erkennen, die bei der Kombination unterschiedlicher Komponenten entstehen.
- System Tests: Werden verwendet, um das komplette, finale System zu testen.

Ein weit verbreitetes Tool für den Test von Web Services ist beispielsweise *soapUI*⁴⁴.

5.5.3 Monitoring

Auch Monitoring des Systems und seiner Services macht Sinn, um zusätzliche Informationen über das Verhalten dieser zu erlangen und dieses überhaupt kontrollieren zu können. Dazu sollten auf mehreren Ebenen (z.B. SOAP-Message-Level) Daten gesammelt werden.

⁴³ BPEL - WS-Business Process Execution Language - XML-basierte Sprache zur Beschreibung von Geschäftsprozessen

⁴⁴ soapUI – Ein Werkzeug zum Softwaretest für SOAP. [Eviw10]

Mittels Monitoring kann auch überprüft werden, ob SLAs eingehalten wurden. Weiters können Performance-Auswertungen bei unterschiedlicher Zugriffs- und Prozesszahl Aufschluss darüber geben, ob Investitionen oder Anpassungen notwendig sind, um SLAs zu erfüllen.

Folgende Parameter sollten zur Laufzeit überprüft werden:

- Verfügbarkeit
- Zugänglichkeit
- Leistung
- Sicherheit
- Zuverlässigkeit

5.6 Transaktionen

Mit Transaktion ist im Fall des Systems eine Folge von Operationen gemeint, die zu einer logischen Einheit gehören. Kommunikationsvorgänge zwischen Diensten können zu Transaktionen zusammengefasst werden und sollen soweit wie möglich vollständig automatisiert ablaufen. Transaktionen bezogen auf Web Services sind ein recht neuer Bereich, jedoch sind die dahinter stehenden Theorien schon wohl bekannt.

Sobald mehrere Web Services ein gemeinsames Ziel erreichen wollen, benötigen sie ein transaktionales Verhalten, welches in jedem Fall konsistent sein muss. Diese Web Services sind somit Teilnehmer einer verteilten Transaktion. [Melz08]

Die wichtigsten Merkmale einer Transaktion sind: [Melz08]

- Mehrere Teilnehmer sind an der Transaktion beteiligt.
- Diese Teilnehmer durchlaufen eine ganz bestimmte, fest im Voraus vorgegebene Prozedur. Die Transaktion führt zu einem bestimmten Ergebnis.
- Dieses Ergebnis muss auf jeden Fall konsistent sein, wobei die Frage, welche Zustände als konsistent betrachtet werden können, von der jeweiligen Situation abhängt.
- Alle Teilnehmer der Transaktion müssen sich nach deren Ende über das Ergebnis der Transaktion bewusst sein und diesem Ergebnis gegebenenfalls zustimmen.

Eine recht allgemein gehaltene Definition wäre: Eine Transaktion ist eine Menge von Arbeitsschritten mit einem konsistenten Ergebnis. Anders ausgedrückt: Die Transaktionen sind ein Hilfsmittel, um inkonsistente Ergebnisse von Operationen oder Mengen von Operationen zu verhindern. Im Bereich von Web-Service-Architekturen kann die Ausführung von Transaktionen besonders komplex werden, da hier die Teilnehmer unter Umständen nur über (SOAP)-Nachrichten miteinander kommunizieren können, wobei die hierbei verwendeten Netzwerkverbindungen jederzeit ausfallen können.

Das ACID-Paradigma ist für Transaktionen von elementarer Bedeutung: [Melz08]

- Die einer Transaktion zugrunde liegende Menge von Arbeitsschritten wird entweder ganz oder gar nicht ausgeführt (Atomarität – atomicity).
- Eine Transaktion führt das ihr zugrunde liegende Informationssystem (Datenbanksystem) von einem konsistenten Zustand in einen anderen konsistenten Zustand (Konsistenz – consistency).
- Die Durchführung einer Transaktion erfolgt isoliert von allen anderen gleichzeitig ausgeführten Transaktionen, d.h. aus Sicht einer jeden Transaktion ist diese die einzige, die in diesem Moment auf dem System ausgeführt wird (Isolation – isolation). Durch das Prinzip der Isolation wird verhindert, dass sich in Ausführung befindliche Transaktionen gegenseitig beeinflussen.
- Nach dem Ende der Transaktion wird ihr Ergebnis in irgendeiner Form persistent gespeichert, um im Fall eines Systemabsturzes beim Wiederhochfahren des Systems wiederhergestellt werden zu können (Dauerhaftigkeit – durability).

Es gibt jedoch Anwendungs-Szenarien, bei denen die so genannte ACIDity, d.h. die Erfüllung aller vier ACID-Eigenschaften, explizit nicht gefordert ist. Ein Szenario, bei dem ACIDity nicht immer erforderlich oder möglich ist, sind verteilte Transaktionen. Eine verteilte Transaktion findet zwischen mehreren Computern statt, die in einem Netzwerk lose miteinander gekoppelt sind. Transaktionen zwischen Web Services sind deshalb grundsätzlich verteilt. [Melz08]

In einem verteilten Umfeld gibt es zwei Arten von Sub-Systemen, die an der Transaktion beteiligt sind: Mehrere Teilnehmer und einen so genannten Koordinator. Ein Teilnehmer einer Transaktion führt einen Teil der Menge von Arbeitsschritten der Transaktion durch. Die Geschäftslogik einer Transaktion liegt somit auf der Seite der Teilnehmer der Transaktion. Ein Koordinator ist diejenige Partei des verteilten Systems, welche die Erzeugung einer neuen Transaktion genauso überwacht wie den Beitritt von Teilnehmern zu einer bereits laufenden Transaktion. [Melz08]

Typischerweise wird die Implementierung des Koordinators durch das Transaktions-Framework bereits mitgeliefert, d.h. die Transaktions-Logik oder das Transaktions-Protokoll ist meistens bereits implementiert. Der wirklich konzeptionell grundlegende Unterschied zwischen verteilten und lokalen Transaktionen ist, dass es für eine verteilte Transaktion sogar von Nachteil wäre, wenn diese die ACID-Eigenschaften erfüllt. [Melz08]

Wann immer eine verteilte Transaktion atomar, konsistent, isoliert oder persistent gemacht wird, erfolgt ein Austausch zwischen der zugesicherten Eigenschaft und der dazugehörigen Einschränkung:

- Atomarität führt zu verlorener Arbeit: Wann immer ein Arbeitsschritt einer Transaktion aus einer großen Anzahl von Schritten, Operationen oder Aktivitäten besteht und diese Transaktion fehlschlägt, so gehen bei einem Rollback alle bereits erfolgreich ausgeführten Arbeitsschritte verloren.
- Konsistenz führt zu vielen Bestätigungs-Nachrichten: Soll eine verteilte Transaktion um jeden Preis ein konsistentes Ergebnis haben, so müssen gegebenenfalls sehr viele Statusabfragen versandt und verarbeitet werden, und zwar auch nach dem Ende der Transaktion (Bestätigung an den Koordinator von allen Teilnehmer).

- Isolation führt zu hohen Sperr-Raten: Isolation bedeutet für eine wichtige Ressource, dass diese für den exklusiven Zugriff eines einzelnen Teilnehmers markiert wird und somit andere Teilnehmer nicht auf diese Ressource zugreifen dürfen. In einer verteilten Umgebung, in der aber Netzwerkverbindungen jederzeit ausfallen können, kann dies besonders schädlich sein.
- Dauerhaftigkeit führt zu einer hohen Anzahl von Log-Dateien: Wann immer das Ereignis einer verteilten Transaktion auf jeden Fall persistent sein soll, um Systemausfälle unbeschadet zu überdauern, muss jeder einzelne Teilnehmer sicherstellen, dass im Falle eines Systemausfalls sein Urzustand lückenlos wiederhergestellt werden kann. Da in einer verteilten Umgebung jedoch Daten nicht an einer zentralen Stelle gespeichert werden, müsste also jeder Teilnehmer den Aufwand betreiben und eine persistente Log-Datei implementieren.

Diese vier Nachteile sind der Grund, warum nur wenige verteilte Transaktionen alle ACID-Eigenschaften erfüllen. Im Kontext von verteilten Transaktionen spielt das so genannte zwei-Phasen-commit-Protokoll (2PC) eine wichtige Rolle. Das 2PC kommt bei verteilten Transaktionen zum Einsatz, sobald die Transaktion beendet werden soll. [Melz08]

Phase 1 – Vorbereiten auf Commit: Bevor der Koordinator entscheiden kann, ob der Ausgang der Transaktion erfolgreich war, muss jeder einzelne Teilnehmer dazu aufgefordert werden, seine Arbeitsschritte, d.h. seinen Teil der Arbeit, zu beenden. Dies wird dadurch realisiert, dass der Koordinator an alle Teilnehmer die so genannte Prepare-Nachricht sendet. Sobald ein Teilnehmer seine Arbeit erledigt hat, muss er dies dem Koordinator gegenüber mit der Prepared-Nachricht bestätigen – oder, falls während des Beendens beim Teilnehmer ein Fehler aufgetreten ist, mit der Cancel-Nachricht. [Melz08]

Phase 2 – Commit oder Abort: Nun liegen dem Koordinator die Ergebnisse der Entscheidungen aller Teilnehmer vor. In dem eindeutigen Fall, in dem alle Teilnehmer mit Prepared geantwortet haben, ist es offensichtlich, dass die Transaktion erfolgreich war. Der Koordinator versendet daraufhin an alle Teilnehmer die Commit-Nachricht, d.h. jeder Teilnehmer soll einen erfolgreich abgeschlossenen Teil der Transaktion als definitiv beendet betrachten. Entscheidend ist nun die Frage, was passiert, wenn mindestens einer der Teilnehmer nicht mit Prepared, sondern mit Cancel geantwortet hat. Handelt es sich um eine atomare Transaktion, so ist die Transaktion fehlgeschlagen, da ein Teilnehmer nicht erfolgreich beendet werden konnte. Der Koordinator versendet die Abort-Nachricht, was dazu führt, dass alle Teilnehmer ihren Teil des Arbeitsschrittes rückgängig machen müssen. Bei einer nicht notwendigerweise atomaren Transaktion bleibt es dem Koordinator überlassen, ob er die Transaktion dennoch als erfolgreich ansieht und an die Teilnehmer die Commit-Nachricht versendet. Hierfür kann der Koordinator eine Geschäftslogik aufrufen, um zu entscheiden, ob die Menge der Teilnehmer, die mit Cancel gestimmt haben, für den Ausgang der Transaktion wirklich entscheidend war oder nicht. [Melz08]

Alle Transaktionen, die bisher betrachtet wurden, waren innerhalb von wenigen Sekunden oder sogar Millisekunden beendet. Dabei war es auch faktisch nie ein Problem, wenn eine (kurz laufende) Transaktion eine exklusive Sperre auf eine Ressource (zum Beispiel eine Datenbank-Tabelle) hatte, um bestimmte Arbeitsschritte isoliert durchführen zu können, da die Transaktion diese innerhalb kürzester Zeit wieder freigeben konnte. Genau genommen sind alle ACID-Transaktionen kurz laufend, wohingegen verteilte Transaktionen entweder kurz laufend oder auch lang laufend sein können. Diese zwei Arten von verteilten Transaktionen benötigen allerdings grundlegend unterschiedliche Implementierungsstrategien. Kurz laufende verteilte

Transaktionen werden typischerweise über das 2PC-Protokoll implementiert. Für die Implementierung lang laufender verteilter Transaktionen hingegen gibt es eine ganze Reihe von so genannten erweiterten Transaktions-Modellen (z.B. *Open Nested Transaction Model*). [Melz08]

Das Web Service Transaction Framework (WS-TX) unterstützt Teilnehmer dabei, einen einheitlichen Verlauf der Kommunikation zu erzielen und hat sich als der Standard für Web-Service-Transaktionen etabliert. WS-TX bietet eine ganzheitliche Spezifikation von Transaktionen für Web Services. WS-TX ist eine Spezifikations-Familie und besteht aus den Spezifikationen WS-Coordination, WS-AtomicTransaction und WS-BusinessActivity. [Melz08]

WS-Coordination beschreibt ein erweiterbares Framework, das Protokolle zur Verfügung stellt, die es ermöglichen Aktionen verteilter Anwendungen miteinander zu koordinieren. Bei einer Transaktion für Web Services sind mehrere Teilnehmer involviert, die möglicherweise zu Beginn der Transaktion noch nicht verfügbar sind. Deshalb muss es für einen solchen Teilnehmer auch möglich sein, einer bereits laufenden Transaktion beizutreten. [Melz08]

Die WS-AtomicTransaction-Spezifikation unterscheidet zwischen drei Rollen:

- Der Koordinator überwacht als zentrale Instanz die Transaktion und erlaubt den Zutritt weiterer Teilnehmer zur Transaktion. Er sammelt das Ergebnis der einzelnen Teilnehmer ein und propagiert das Gesamtergebnis der globalen Transaktion an die Teilnehmer.
- Der Initiator erzeugt einen Transaktionskontext und startet die verteilten Arbeitsschritte. Am Ende veranlasst er den Abschluss oder den Abbruch der globalen Transaktion durch Senden der jeweiligen Status-Nachricht an den Koordinator.
- Ein Teilnehmer führt Arbeitsschritte einer Transaktion aus und teilt den Ausgang seiner Arbeit dem Koordinator mit.

In der Praxis wird WS-BusinessActivity hauptsächlich dazu eingesetzt, mehrere kleine AtomicTransactions zu einer im wahrsten Sinne des Wortes größeren Business-Aktivität zusammenzukoppeln. Tritt während der Ausführung einer BusinessActivity ein Fehler auf, so müssen alle bereits erfolgreich durchgeführten Arbeitsschritte kompensiert werden. Kompensation bedeutet, dass explizit hierfür vorgesehener Code aufgerufen wird, um die Effekte einer bereits erfolgreich beendeten Aktion rückgängig zu machen. [Melz08]

5.7 Zahlungsverfahren

Im Service-System wird zwar beispielhaft eine Buchung durchgeführt, jedoch wird auf den Zahlungsvorgang nicht näher eingegangen. Im Service-System könnte dieser beispielsweise durch einzelne Services umgesetzt werden und wie gewünscht in Kompositionen eingegliedert werden. Weiters ist zu bedenken, dass es aufgrund unterschiedlicher Verträge zu unterschiedlichen Zahlungsmethoden oder -konditionen kommen kann, was ebenfalls berücksichtigt werden sollte. Entsprechend der Komposition sollten unterschiedliche Zahlungsmethoden zur Verfügung stehen.

Elektronische Zahlungsvorgänge können wie folgt klassifiziert werden: [MeSt05]

- Aufgrund der Höhe des Betrags: Pico-, Micro- und Macropayment
- Durch den Zeitpunkt der Zahlung: Prepaid, Pay-Now, Pay-later
- Anhand des technologischen Konzept: Differenzierung bei Art der Speicherung, z.B. Konto beim Betreiber, virtuelle Münzen

Weiters lässt sich auch eine Unterteilung in unterschiedliche Zahlungsverfahren durchführen: [MeSt05]

- Kreditkartenbasierte Verfahren, wie Paypal oder SET
- Guthabenbasierte Verfahren, wie Geldkarte oder Paysafecard
- Innovative Lösungen wie, eCash, Millicent, PayWord oder MicroMint
- Lösungen für Shopbetreiber, wie Google Checkout oder Amazon FPS
- Lösungen für kostenpflichtige Websites, wie Firstgate Click & Buy oder Allopass
- Zufälliges Bezahlen, bei dem nicht alle Benutzer zahlen, sondern von einem Zufallsalgorithmus ausgewählt.

6. Schlussfolgerungen, Fragen und Ausblicke

Um die Serviceorientierung ist in den letzten Jahren ein richtiger Hype entstanden. Viele Beratungsunternehmen nutzen dies aus und versuchen, mit modernen Schlagwörtern die Kundschaft zu beeindrucken. Wie schon breit diskutiert, ist die Ausrichtung des Unternehmens auf eine serviceorientierte Umgebung nicht zwangsweise von Erfolg gekrönt. Innerhalb der Arbeit wurden vielfach Kriterien besprochen, die vorab geklärt und berücksichtigt werden müssen.

In der Arbeit konnte prototypisch gezeigt werden, wie Web Services eingesetzt werden können, um ein flexibles verteiltes System zu erstellen. Zusätzlich wurde geschildert, was getan werden muss, um solch ein System zu erweitern, damit es dynamischer eingesetzt werden kann. Weiters wurde darauf hingewiesen, wo aufgrund der Softwarearchitektur Probleme auftauchen könnten und mit welchen Alternativen diese umgangen werden können.

Weiters wurde dargestellt, was für Vorteile solch ein Service-System für alle Beteiligten mit sich bringen kann und wie es sich zu einem großen Online-Marktplatz etablieren könnte. Das Internet hat sich längst fest in den Alltag von Organisationen und Menschen integriert. Dennoch zögern einige Unternehmen noch damit, ihre Angebotspalette und Abläufe um E-Services zu erweitern. Die Konsumenten fordern allerdings immer mehr danach, da meist dadurch der Ablauf oder ein Prozess vereinfacht oder der Aufwand verringert werden kann. Dies könnte auf ungenügende Kenntnisse, schlechte Beratung oder auf zu geringe finanzielle Mittel zurückzuführen sein. Ein frei zugängliches System könnte auch diesen Unternehmen den Zugang ermöglichen bzw. Probleme vereinfachen und Anliegen umzusetzen.

Generell kann von immer wichtiger werdenden Online-Marktplätzen berichtet werden, die auch an immer mehr Einfluss gewinnen (z.B. ebay, amazon, myhammer, etc.). Viele davon beziehen sich allerdings nach wie vor auf Sachleistungen, wobei es durchaus schon auch Online-Portale für Dienstleistungen gibt. Viele Anbieter nutzen diese Portale heutzutage mittlerweile als einzigen Auftritt gegenüber den Kunden. Vorteile, wie geringere Werbekosten, breitere Zielgruppen oder ein größerer Marktplatz wurden ebenfalls innerhalb dieser Arbeit diskutiert.

Die Idee des Service-Systems hätte ebenfalls das Potenzial solch eine Präsenz zu übernehmen, wobei es viele unterschiedliche Geschäftsmodelle und Umsetzungsmöglichkeiten geben würde. Einerseits könnte der Zugang für Anbieter frei sein, eine Umsatzbeteiligung bei Verkäufen über das System könnte jedoch eingefordert werden. Andererseits könnte auch der Service-Zugang in Rechnung gestellt werden. Auch ein komplett offenes System, das sich mit Spenden oder Werbung finanziert wäre denkbar.

Bei einer steigenden Benutzeranzahl oder bei einer Erhöhung der Servicequalität ist gleichzeitig mit zunehmenden Kosten für den laufenden Betrieb zu rechnen. Unabhängig vom gewählten Geschäftsmodell möchte man natürlich zumindest genug Einnahmen lukrieren, um den laufenden Betrieb und mögliche notwendige Investitionen abdecken sowie alle beteiligten Mitarbeiter entsprechend entlohnen zu können.

Bei einer funktionierenden Partnerschaft sehen sich heutzutage die Beteiligten nicht unbedingt regelmäßig nach Alternativen um. Wenn dies über eine Plattform, wie das Service-System, automatisiert möglich ist, könnte sich die Tendenz ändern. Elementar dabei ist jedoch, dass die Dienstleistungsqualität vor Auftragsabschluss gut eingeschätzt werden kann. Eine detaillierte Beschreibung ist da nur der Grundbaustein, wobei hier noch reichlich Innovationsbedarf

besteht. Für den Kunden sollen die Leistungen vor dessen Konsum besser belichtet werden können.

Um ein erfolgreiches System auf die Beine zu stellen, macht es auch Sinn, sich an den momentan dominierenden Internetfirmen zu orientieren und deren derzeit sehr gut funktionierende Geschäftsmodelle zu analysieren. Häufig werden frei zugängliche, anstatt lizenzgebundene oder kostenpflichtige Dienste angeboten, was aber zu einem steigenden Popularitätsgrad und einem stetigen Userwachstum führt. Oft werden dabei die Inhalte dieser Portale ausschließlich von den Usern gefüllt und dieses von einem Betreiber nur verwaltet. Die Benutzer stellen dabei meist uneigennützig Informationen zur Verfügung, wobei die Portale dadurch einen stetigen Wissenszuwachs erfahren, von dem alle gemeinsam profitieren können. Beispielhaft können hier Vertreter wie Facebook, Wikipedia, YouTube oder Amazon genannt werden, die zwar verschiedene Geschäftsmodelle einsetzen, das grundlegende Prinzip der frei zugänglichen Services doch bei allen vorhanden ist. Da diese Entwicklungen noch eher am Anfang stehen, bietet das Internet noch viel Platz für solche innovative Ideen.

Viele dieser großen Anbieter stellen ihre Dienste zusätzlich im Internet über Web Services zur Verfügung, wobei es sich mittlerweile schon um ganze Schnittstellenpakete handelt, die zur freien Benutzung bereit stehen. So verwenden beispielsweise häufig Hotel-, Flug- und Reiseportale Web Services, um auf die einzelnen Anbieter zuzugreifen und deren Daten abzufragen. Diese bereits realisierte Umsetzung spiegelt die Grundfunktionalität des Service-Systems wieder. Wo früher noch regelmäßig Datenbestände synchronisiert werden mussten, kann heute in Echtzeit auf aktuelle Buchungsdaten zugegriffen werden. Wünschenswert wäre es, genau solch ein System um möglichst viele Branchen und deren Anbieter zu erweitern, damit ein möglichst großer Marktplatz erzielt wird.

Durch Hinzufügen der Kompositionsmöglichkeit können neue Kooperationen geschlossen und die Funktionalitäten für die Benutzer erweitert werden – unter Umständen auch automatisiert. Dynamisch gebundene Kooperationspartner können dem Kunden gegenüber als ein virtuelles Unternehmen auftreten.

Dies kann freilich auch Nachteile mit sich bringen, da es in diesem Fall zu virtuellen Unternehmen kommen kann, bei denen sich die Teilnehmer untereinander nicht kennen. Das muss kein Nachteil sein, wobei manche Organisationen womöglich mit bestimmten anderen nicht zusammen auftreten und genannt werden wollen. Beispielsweise könnten Unternehmen eine Kooperation verweigern wollen, da sie durch gemeinsames Auftreten Angst hätten, die eigene Reputation zu beschädigen. Deshalb müssen genaue Richtlinien und Regeln definierbar sein, um solche Wünsche abzubilden.

Deshalb muss besondere Beachtung dem Online-Vertrag geschenkt werden. Es muss sichergestellt werden, dass alle Beteiligten wissen, wer ihr Vertragspartner ist und wer für die Durchführung der Leistung und Einhaltung der Qualität verantwortlich ist. Weiters muss klar sein, welches Recht gilt und im Ernstfall anzuwenden ist. Recht ist auch ein Gebiet mit vielen Veränderungen, die große Auswirkungen auf das System haben könnten, bei dem eine schnelle Anpassungsfähigkeit von klarem Vorteil ist.

Die meisten Länder haben in ihrer Gesetzgebung schon ein E-Commerce-Gesetz oder diverse Fernabsatzrichtlinien verankert, wobei es große nationale Unterschiede gibt. Solche juristische Rahmenbedingungen dürfen bei der Gestaltung des Geschäftsmodells nicht fehlen, da sie dieses essentiell beeinflussen können.

In dieser Arbeit wurde gezeigt, wie einzelne Geschäfts-Services mittels Web Services unterstützt oder teilweise auch komplett umgesetzt werden können. Web Services sind jedoch nicht die einzige Möglichkeit dies zu tun. Im Laufe des Grundlagenkapitels wurde gezeigt, dass am Abwicklungsprozess eines Services der Konsument immer in gewisser Weise beteiligt ist und als Koproduzent betrachtet werden kann. Je nach Beteiligungsgrad ist dies mehr oder weniger der Fall. Es stellt sich somit die Frage, ob der Konsument auch nach wie vor bei einer Geschäftsautomatisation beteiligt ist. Im Fall des prototypisch umgesetzten Service-Systems muss der Kunde zumindest Kriterien angeben, welche Art von Ressourcen er benötigt und anschließend in der Ergebnisliste auswählen, welche dieser Ressourcen er in seinem Portfolio für die Buchung enthalten haben möchte. Demnach ist der Konsument auch in diesem Fall immer an der Erstellung beteiligt, da er jedenfalls einmalig seine Präferenzen angegeben hat. Und selbst, wenn automatisiert aus einem generierten Kundenprofil (z.B. aufgrund Surfverhalten in einem Online-Shop) ein Angebot erstellt wird, ist es noch immer der Kunde (bzw. seine "Charakteristika") selbst, der den notwendigen Ausschlag für die Angebotsausprägung gegeben hat. Abgesehen davon ist in jedem Fall die Willenserklärung zum Kauf – in welcher Form sie auch immer ausfallen mag – des Kunden notwendig, um die Übereinkunft abzuschließen.

Die Service-Innovation ist derzeit sowohl in der Forschungs- als auch in der Unternehmerwelt ein stark verfolgtes Thema, wobei die Entwicklung eher noch am Anfang steht. Ob sich auf akademischer Ebene eine eigene Disziplin des Service Science bilden kann, bleibt noch ungeklärt. Die interdisziplinäre Forschung wird in den nächsten Jahren sicherlich noch fortgesetzt und die Serviceorientierung in den Unternehmen weiter forciert werden.

7. Literaturverzeichnis

- [ApFr10] Apache Friends . XAMPP – Apache Friends-Projekt . Online-Quelle. <<http://www.apachefriends.org/de/xampp.html>>, 15.03.2010.
- [ASFo10] Apache Software Foundation. Apache HTTP Server. Online-Quelle. <<http://httpd.apache.org>>, 15.03.2010.
- [Appl10] Apple. Appstore – Applikationen für das iPhone. Online-Quelle. <<http://www.apple.com/at/iphone/apps-for-iphone>>, 15.03.2010.
- [Appl10a] Apple. Apple iPhone. Online-Quelle. <<http://www.apple.com/at/iphone>>, 15.03.2010.
- [BaGO04] Baida, Z., Gordijn, J. und Omelayenko, B.: *A Shared Service Terminology for Online Service Provisioning*, ICEC'04, Sixth International Conference on Electronic Commerce, 2004.
- [BaNi07] Baresi, L. und Nitto, E. D.: *Test and Analysis of Web Services*, Springer, 2007.
- [Berg07] Berger, T.: *Service-Level-Agreements: Konzeption und Management von Service-Level-Agreements für IT-Dienstleistungen*, 1. Auflage, Vdm Verlag Dr. Müller, 2007.
- [Bett01] Bettag, U.: *Web-Services*, Informatik-Spektrum (24, 5), S. 302-304, 2001.
- [BoKr06] Böhmman, T. und Krmar, H.: *Modulare Servicearchitekturen*, Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, Springer, 2006, S. 377-400.
- [Bruh08] Bruhn, M.: *Qualitätsmanagement für Dienstleistungen*, 7. Auflage, Springer Berlin Heidelberg, 2008.
- [Bull06] Bullinger, H.-J.: *Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen*, Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, Springer, S. 54-85, 2006.
- [BuSc06] Bullinger, H.-J. und Scheer, A.-W.: *Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen*, in Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen, Springer, S. 3-18, 2006.
- [CaSF10] Cake Software Foundation. CakePHP Web Framework. Online-Quelle. <<http://cakephp.org>>, 15.03.2010.
- [Ches05] Chesbrough, H.: *Toward a new science of services*, Harvard Business Review (83), S. 43-44, 2005.
- [ChSp06] Chesbrough, H. und Spohrer, J.: *A research manifesto for services science*, Communications of the ACM (49), S. 35-40, 2006.
- [Cors01] Corsten, H.: *Dienstleistungsmanagement*, 4. Auflage, Oldenbourg Wissenschaftsverlag, 2001.
- [Diet06] Dietrich, B.: *Resource planning for business services*, Communications of the ACM (49), S. 62-64, 2006.
- [Dorn09] Dorn, J.: *Service Scheduling*, Service Science and Logistics Informatics: Innovative Perspectives, 2009.

- [DoWe08] Dorn, J. und Werthner, H.: *Service-Oriented Resource Management*, Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008.
- [Enge08] Engelmann, K.: *Where Practice Meets Theory*, Services Science, Fundamentals Challenges and Future Development, Springer, S.119-136, 2008.
- [Erl05] Erl, T.: *Service-Oriented Architecture*. Concepts, Technology and Design, Prentice-Hall, 2005.
- [Erl08] Erl, T.: *SOA: Entwurfsprinzipien für service-orientierte Architekturen*, 1. Auflage, Addison-Wesley, 2008.
- [Eviw10] Eviware Software. soapUI. Online-Quelle. <<http://www.soapui.org>>, 15.03.2010.
- [Ganz08] Ganz, W.: *Research Needs in the European Service Sector*, Services Science, Fundamentals Challenges and Future Development, Springer, S. 85-87, 2008.
- [Gron94] Grönroos, C.: *From Scientific Management to Service Management: A Management Perspective for the Age of Service Competition*, International Journal of Service Industry Management (5), S. 5-20, 1994.
- [Grub07] Gruber, T.: *Ontology*. Online-Quelle. <<http://tomgruber.org/writing/ontology-definition-2007.htm>>, 15.03.2010.
- [Hess02] Hesse, W.: *Ontologie(n)*, Informatik-Spektrum (25, 6), S. 477-480, 2002.
- [HKRS08] Hitzler, P., Krötzsch, M., Rudolph, S. und Sure, Y.: *Semantic Web*, Springer, 2008.
- [Hrad08] Hradil, S.: *Sozialstruktur und gesellschaftlicher Wandel*, Die EU-Staaten im Vergleich, 3. Auflage, VS Verlag für Sozialwissenschaften, S. 89-123, 2008.
- [IBMI01] IBM und Microsoft. Web Services Inspection Language. Online-Quelle. <<http://www.ibm.com/developerworks/library/specification/ws-wsilspec>>, 15.03.2010.
- [Kotl88] P. Kotler.: *Marketing Management: Analysis, Planning, Implementation and Control*, 6. Auflage, Prentice Hall, Englewood Cliffs, NJ, 1988.
- [LeDL07] Leyking, K., Dreifus, F. und Loos, P.: *Vergleichende Literaturstudie: Serviceorientierte Architekturen*, Wirtschaftsinformatik (49), S. 394-401, 2007.
- [Lemm05] Lemmink, J.: *The need for more multidisciplinary research*, International Journal of Service Industry Management (16), S. 7-9, 2005.
- [LuVa04] Lusch, R. F. und Vargo, S. L.: *Evolving to a New Dominant Logic for Marketing*, Journal of Marketing (68), S. 1-17, 2004.
- [LuVT09] Lusch, R. F., Vargo, S. L. und Tanniru, M.: *Service, value networks and learning*, Journal of the Academy of Marketing Science, 2009.
- [Masa07] Masak, D.: *SOA? Serviceorientierung in Business und Software*, Springer, 2007.
- [MeBr00] Meffert, H. und Bruhn, M.: *Dienstleistungsmarketing. Grundlagen - Konzepte - Methoden*, 3. Auflage, Gabler, 2000.
- [Melz08] Melzer, I.: *Service-orientierte Architekturen mit Web Services*, 3. Auflage, Spektrum Akademischer Verlag, 2008.

- [MeSt05] Meier, A. und Stormer, H.: *eBusiness & eCommerce, Management der digitalen Wertschöpfungskette*, 1. Auflage, Springer, 2005.
- [OASI04] OASIS. UDDI Version 3.0.2 Spec Technical Committee Draft. Online-Quelle. <http://uddi.org/pubs/uddi_v3.htm>, 15.03.2010.
- [OASI10] OASIS. Web Services Security. Online-Quelle. <www.oasis-open.org/committees/wss>, 15.03.2010.
- [Orac10] Oracle. MySQL Server. Online-Quelle. <<http://www.mysql.com>>, 15.03.2010.
- [OSEH02] O'Sullivan, J., Edmond, D., A. H. M. ter Hofstede: *Service Description: A survey of the general nature of services*, Report FIT-TR-2003-02, 2002.
- [OvTu08] Overhage, S. und Turowski, K.: *Serviceorientierte Architekturen – Konzept und methodische Herausforderungen*, in *Service-orientierte Architekturen*, Gabler, S. 3-17, 2008.
- [PaZB88] Parasuraman, A., Zetthaml, V. and Berry, L.: *SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perception of Service Quality*, *Journal of Retailing* (64/1), S. 12-40, 1988.
- [PHP10] PHP. PHP: Hypertext Preprocessor. Online-Quelle. <<http://www.php.net>>, 15.03.2010.
- [Reck06] Reckenfelderbäumer, M.: *Kundenmitwirkung bei der Entwicklung von industriellen Dienstleistungen – eine phasenbezogene Analyse*, in *Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen*, Springer, S. 142-166, 2006.
- [Roth06] Roth, S.: *Preismanagement für Leistungsbündel*, DUV, 2006.
- [RoWo06] Roth, S. und Woratschek, H.: *Preisbildung und Preisdifferenzierung für Absatzleistungen und Leistungsbündel*, *Innovatives Dienstleistungsmarketing in Theorie und Praxis*, DUV, S. 313-335, 2006.
- [ScGK06] Scheer, A.-W., Grieble, O. and Klein, R.: *Modellbasiertes Dienstleistungsmanagement*, *Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen*, Springer, S. 19-51, 2006.
- [SEKL08] Stauss, B., Engelmann, K., Kremer, A. und Luhn, A.: *Services Science, Fundamentals Challenges and Future Development*, Springer, 2008.
- [SiKI09] Sichtmann, C. and Klein, M.: *Qualitätssicherung bei der Internationalisierung von Dienstleistungen*, *Aktuelle Forschungsfragen im Dienstleistungsmarketing*, S. 169-187, 2009.
- [Soel08] Söllner, A.: *Einführung in das Internationale Management*, Gabler, 2008.
- [Spoh08] Spohrer, J.: *Services Science, Management Engineering (SSME) and Its Relation to Academic Disciplines*, *Services Science, Fundamentals Challenges and Future Development*, Springer, S. 11-40, 2008.
- [Stau08] Stauss, B.: *International Service Research - Status Quo, Developments, and Consequences for the Emerging Services Science*, *Services Science, Fundamentals Challenges and Future Development*, Springer, S. 57-70, 2008.
- [Symf10] Symfony. Symfony Web Framework. Online-Quelle. <<http://www.symfony-project.org>>, 15.03.2010.

- [W3C04] W3C. Web Service Architectures. Online-Quelle. <<http://www.w3.org/TR/ws-arch>>, 15.03.2010.
- [W3C04a] W3C. OWL Web Ontology Language: Overview. Online-Quelle. <<http://www.w3.org/TR/owl-features>>, 15.03.2010.
- [W3C04b] W3C. RDF Primer. Online-Quelle. <<http://www.w3.org/TR/rdf-primer>>, 15.03.2010.
- [W3C04c] W3C. RDF/XML Syntax Specification. Online-Quelle. <<http://www.w3.org/TR/rdf-syntax-grammar>>, 15.03.2010.
- [W3C06] W3C. Primer: Getting into RDF & Semantic Web using N3. Online-Quelle. <<http://www.w3.org/2000/10/swap/Primer.html>>, 15.03.2010.
- [W3C07] W3C. SOAP Version 1.2. Online-Quelle. <<http://www.w3.org/TR/soap12-part1>>, 15.03.2010.
- [W3C07a] W3C. Web Services Description Language (WSDL) Version 2.0: Core Language. Online-Quelle. <<http://www.w3.org/TR/wsdl20>>, 15.03.2010.
- [W3C07b] W3C. Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. Online-Quelle. <<http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626>>, 15.03.2010.
- [Weig07] Weigle, J.: *Informationsverarbeitung und -verteilung virtualisierter Organisationen*, DUV, 2007.
- [Wiki10] Wikipedia. Kategorie: Sicherheitslücke. Online-Quelle. <<http://de.wikipedia.org/wiki/Kategorie:Sicherheitsl%C3%BCcke>>, 15.03.2010.
- [ZeBi96] V. Zeithaml und M. J. Bitner. *Services Marketing*. McGraw-Hill, New York, NY, 1996.
- [ZePM02] Zeithaml, V. A., Parasuraman, A. and Malhotra, A.: *Service Quality Delivery Through Web Sites*, *Journal of the Academy of Marketing Science* (30/4), S. 362-375, 2002.
- [ZeTe10] Zend Technologies. Zend Framework. Online-Quelle. <<http://www.zend.com>>, 15.03.2010.
- [ZuGK04] Zuser, W., Grechenig, T. und Köhle, M.: *Software Engineering. Mit UML und dem Unified Process*, 2. Auflage, Pearson Studium, 2004.

8. Abbildungsverzeichnis

Abbildung 1: Beteiligte einer SOA.....	34
Abbildung 2: Systemkonzept	80
Abbildung 3: Anwendungsfälle - Überblick.....	84
Abbildung 4: Anwendungsfall - Authentifizierung.....	84
Abbildung 5: Anwendungsfall - System verwalten	85
Abbildung 6: Anwendungsfall - Systemanschluss verwalten.....	86
Abbildung 7: Anwendungsfall - Ressourcen reservieren	87
Abbildung 8: Anwendungsfall - Ressourcen buchen	88
Abbildung 9: Datenmodell - Userverwaltung	89
Abbildung 10: Datenmodell - Serviceverwaltung	90
Abbildung 11: Datenmodell - Raumverwaltung	90
Abbildung 12: Datenmodell - Raumverwaltung2	91
Abbildung 13: Datenmodell - Ressourcenverwaltung	91
Abbildung 14: Datenmodell - Kompositionsverwaltung.....	92
Abbildung 15: Datenmodell - Reservierungsverwaltung.....	93
Abbildung 16: Datenmodell - Buchungsverwaltung	93
Abbildung 17: Datenmodell - Gesamtmodell	93
Abbildung 18: Systemabläufe - Suchvorgang	94
Abbildung 19: Systemabläufe - Reservierungsvorgang	94
Abbildung 20: Systemabläufe - Buchungsvorgang	95
Abbildung 21: ER-Diagramm - Anbietersystem	100
Abbildung 22: ER-Diagramm - Service-System	105

9. Tabellenverzeichnis

Tabelle 1: Gegenüberstellung - SDL und GDL	21
Tabelle 2: Vorgehensmodell - Dienstleistungsentwicklungsprozess	25
Tabelle 3: Vorgehensmodell - Methodenbeispiele	25
Tabelle 4: Beispiel einer SOAP-Nachricht	40
Tabelle 5: SOAP - Elemente eines Fehlereintrags	41
Tabelle 6: SOAP - Fehlercodes	42
Tabelle 7: UDDI - businessEntity Datentyp	47
Tabelle 8: UDDI - businessService Datenstruktur	47
Tabelle 9: UDDI - bindingTemplate Datenstruktur	48
Tabelle 10: UDDI - tModel Datenstruktur	48
Tabelle 11: Qualitätsdimensionen - Kennzahlen	64
Tabelle 12: Räumlichkeiten - Ausstattungsmerkmale	73
Tabelle 13: Räumlichkeiten – Umgebungsmerkmale	73
Tabelle 14: Service-System - Glossar	77
Tabelle 15: Serviceprofile - Datenstruktur	82
Tabelle 16: Servicefähigkeiten - Datenstruktur	83
Tabelle 17: Anwendungsfallbeschreibung - Ressource buchen	88
Tabelle 18: Buchung - iCalendar-Termin.....	93
Tabelle 19: WSDL - Beschreibung des Web Services zur Suche.....	102
Tabelle 20: WSDL - Beschreibung des Web Services zur Buchung.....	104
Tabelle 21: Sicherheitsrelevante Themen	118

