

Selektion von Web Services anhand nicht-funktionaler Zielgrößen

DIPLOMARBEIT

zur Erlangung des akademischen Grades
eines Magisters
der Sozial- und Wirtschaftswissenschaften

eingereicht bei Herrn

Univ.-Prof.

Dr. Bernd Heinrich

Institut für Wirtschaftsinformatik, Produktionswirtschaft und Logistik

Fakultät für Betriebswirtschaft

der Leopold-Franzens-Universität Innsbruck

von

Matthias Paulsteiner

Innsbruck, September 2011

**Qualität ist die Gebrauchstauglichkeit einer
erstellten Leistung in den Augen der Kunden**

Joseph M. Juran

Inhaltsübersicht

Abkürzungsverzeichnis	v
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
1 Einleitung, Definitionen und Forschungsfrage	1
1.1 Motivation	1
1.2 Web Service Grundlagen	1
1.3 Forschungsfrage und erwartetes Ergebnis	11
1.4 Gliederung der Arbeit	12
2 Nicht-funktionale Zielgrößen	13
2.1 Definition	13
2.2 Unterscheidung zwischen funktionalen und nicht-funktionalen Zielgrößen	14
2.3 Spezifizierung und Ausführung der einzelnen nicht funktionalen Zielgrößen	14
3 Selektion von Web Services	21
3.1 Anforderungen zur Selektion von Web Services basierend auf nicht-funktionalen Zielgrößen	21
3.2 Verfahren zur Selektion von Web Services	30
3.3 Re-planning	57
4 Detailliertes Verfahrensbeispiel	61
4.1 Allgemeine Angaben.....	61
4.2 Berechnung exaktes Verfahren – BBLP Algorithmus	64
4.3 Berechnung heuristisches Verfahren – WS_HEU Algorithmus	67
4.4 Berechnung heuristisches Verfahren – MCSP-K Algorithmus	74
5 Monitoring zur Bestimmung der Qualität von Web Services bei Web Service Anbietern und in Dienstgüteverzeichnissen	86
6 Zusammenfassung, kritische Würdigung und Ausblick	94
Literatur	98

Inhaltsverzeichnis

Abkürzungsverzeichnis	v
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
1 Einleitung, Definitionen und Forschungsfrage	1
1.1 Motivation	1
1.2 Web Service Grundlagen	1
1.2.1 Was ist ein Web Service?	2
1.2.1.1 Definition Web Service	2
1.2.1.2 Vorteile der Verwendung von Web Services	4
1.2.1.3 Beispiele zur Einsatzmöglichkeit von Web Services	4
1.2.2 Web Service - Web Service Komposition	5
1.2.3 Geschäftsprozesse und Workflows	6
1.2.3.1 Geschäftsprozess	6
1.2.3.2 Workflow	7
1.2.4 Web Service Selektion/ Selektionskriterien	8
1.2.5 Serviceorientierte Architektur (SOA)	9
1.2.6 Web Service Technologie	9
1.3 Forschungsfrage und erwartetes Ergebnis	11
1.4 Gliederung der Arbeit	12
2 Nicht-funktionale Zielgrößen	13
2.1 Definition	13
2.2 Unterscheidung zwischen funktionalen und nicht-funktionalen Zielgrößen	14
2.3 Spezifizierung und Ausführung der einzelnen nicht funktionalen Zielgrößen	14
3 Selektion von Web Services	21
3.1 Anforderungen zur Selektion von Web Services basierend auf nicht-funktionalen Zielgrößen	21
3.1.1 Erstellung der optimalen Servicekette	21
3.1.2 Ausgangssituation der Selektion	23
3.1.3 Aggregation der nicht-funktionalen Zielgrößen	24
3.1.4 Herausforderungen der Selektion - Performance der Selektionsverfahren	26
3.1.4.1 Selektionszeit/ Echtzeitselektion	26

3.1.4.2	Web Service Abhängigkeiten und Konflikte.....	28
3.1.4.3	Zusammenpassen der Schnittstellen (Interface-Matching)	29
3.2	Verfahren zur Selektion von Web Services	30
3.2.1	Verfahrensbeispiel	30
3.2.2	Vereinfachungen der Selektion.....	34
3.2.3	Explizite Selektionsverfahren - ein Optimierungsproblem	37
3.2.3.1	Exakte Selektionsverfahren.....	38
3.2.3.2	Heuristische Verfahren.....	42
3.2.4	Strukturierter Vergleich und Bewertung der Verfahren zur Selektion von Web Services basierend auf nicht-funktionalen Zielgrößen	49
3.2.4.1	Leistungsvermögen der Selektionsverfahren	49
3.2.4.2	Entwicklung der Selektionsverfahren	51
3.3	Re-planning.....	57
3.3.1	Definition Re-planning	57
3.3.1.1	Sicherstellung der Ausführbarkeit.....	57
3.3.1.2	Gewährleistung der Gültigkeit des Prozesses hinsichtlich der Restriktionen des Nutzers	59
3.3.1.3	Gewährleistung des optimalen Prozesses hinsichtlich der Präferenzen des Nutzers.....	59
3.3.2	Re-planning Beispiel	60
4	Detailliertes Verfahrensbeispiel	61
4.1	Allgemeine Angaben.....	61
4.2	Berechnung exaktes Verfahren – BBLP Algorithmus	64
4.3	Berechnung heuristisches Verfahren – WS_HEU Algorithmus	67
4.3.1	Angaben zu WS_HEU	67
4.3.2	Berechnung WS_HEU	69
4.4	Berechnung heuristisches Verfahren – MCSP-K Algorithmus	74
4.4.1	Angaben zu MCSP-K	74
4.4.2	Berechnung MCSP-K	75
5	Monitoring zur Bestimmung der Qualität von Web Services bei Web Service Anbietern und in Dienstgüterverzeichnissen	86
5.1.1	Web Service Dienstverzeichnisse.....	86
5.1.2	Web Service Anbieter.....	89

6	Zusammenfassung, kritische Würdigung und Ausblick.....	94
	Literatur	98

Abkürzungsverzeichnis.

bspw.	beispielsweise
bzgl.	bezüglich
bzw.	beziehungsweise
ca.	circa
d. h.	das heißt
et al.	et altera
etc.	et cetera
o. V.	ohne Verfasser
S.	Seite
u. a.	unter anderem
z. B.	zum Beispiel

Abbildungsverzeichnis

Bild 1-1 Web Service Programming Stack (Gottschalk et al. 2002, S. 171).....	10
Bild 1-2 Aufbau einer Web Service Architektur in Anlehnung an Bächle und Lehmann (2010, S. 32).....	11
Bild 2-1 Kategorisierung der nicht-funktionalen Zielgrößen in Anlehnung an Berbner (2007, S. 37).....	16
Bild 3-1 Die fünf Schritte des Konzepts (Heinrich et al. 2011b, S. 90).....	22
Bild 3-2 UML-Aktivitätsdiagramm Reisebuchung (Berbner 2007, S. 31)	25
Bild 3-3 Verbindungsstrukturen in Anlehnung an Yu et al. (2007, S. 5).....	25
Bild 3-4 Zeitpunkte der Selektion	27
Bild 3-5 Prozessmodell Kreditwürdigkeitsprüfung in Anlehnung an Heinrich et al. (2011b, S. 93-96)	31
Bild 3-6 Selektion einer Servicekette des Beispiels Kreditwürdigkeitsprüfung, in Anlehnung an Heinrich et al. (2011b, S. 90-95)	32
Bild 3-7 Beispiel für die Notation des SZG (Heinrich et al. 2011b, S. 92).....	58
Bild 4-1 Aktionskette sequentieller Struktur.....	61
Bild 4-2 Berechnung BBLP Algorithmus	65
Bild 4-3 Ablauf WS_HEU (Yu et al. 2007, S. 10).....	68
Bild 4-4 Definitionen des Ablauf WS_HEU (Yu et al. 2007, S. 11).....	69
Bild 4-5 Algorithmus MCSP (Yu et al. 2007, S. 12).....	74
Bild 4-6 Algorithmus MCSP-K.....	75
Bild 4-7 Topologische Sortierung der Knoten	76
Bild 4-8 Selektierte Servicekette durch MCSP-K.....	85
Bild 5-1 Dokumentation nicht-funktionaler Zielgrößen bei eSigma.....	87
Bild 5-2 Dokumentation nicht-funktionaler Zielgrößen bei seekda.....	88

Tabellenverzeichnis

Tabelle 2-1 Erläuterung nicht-funktionaler Zielgrößen 1.....	17
Tabelle 2-2 Erläuterung nicht-funktionaler Zielgrößen 2.....	18
Tabelle 2-3 Erläuterung nicht-funktionaler Zielgrößen 3.....	19
Tabelle 3-1 Aggregationsformeln zur Berechnung des Gesamtpräferenzfunktionswerts in Anlehnung an Strunk (2010, S. 68).....	26
Tabelle 3-2 Übersicht exakte Selektionsverfahren 1 in Anlehnung an Strunk (2010, S. 71)...	53
Tabelle 3-3 Übersicht exakte Selektionsverfahren 2 in Anlehnung an Strunk (2010, S. 71)...	54
Tabelle 3-4 Übersicht heuristische Selektionsverfahren 1 in Anlehnung an Strunk (2010, S. 72).....	55
Tabelle 3-5 Übersicht heuristische Selektionsverfahren 2 in Anlehnung an Strunk (2010, S. 72).....	56
Tabelle 4-1 Übersicht der Web Services und der Werte der nicht-funktionalen Zielgrößen ..	62
Tabelle 4-2 Parameter und Notationen (Yu et al. 2007, S. 6)	62
Tabelle 4-3 Berechnung Servicenutzen F.....	64
Tabelle 4-4 Berechnung der Werte für erste zulässige Lösung.....	70

1 Einleitung, Definitionen und Forschungsfrage

1.1 Motivation

Im Zuge der Globalisierung, der Deregulierung der Märkte und anspruchsvoller gewordener Kunden und Investoren sind Unternehmen einem hohen Kostendruck ausgesetzt. Dadurch hat die Art des wirtschaftlichen Handels gerade in der jüngsten Vergangenheit einen dramatischen Wandel durchlebt. Um ihr Geschäft zu optimieren und sich von der Konkurrenz abheben zu können, müssen sich Unternehmen auf ihre Kernkompetenzen konzentrieren. (Lotz 2007, S. 644-645) Um als Unternehmen Erfolg zu haben, sind die wichtigsten Faktoren Prozesse innovativ und dynamisch adaptierbar zu gestalten. Für dynamisch adaptierbare Prozesse benötigt man eine flexible IT-Architektur, da verschiedene heterogene Kommunikationsmechanismen, Betriebssysteme, Legacy-Anwendungen und Plattformen darin integriert werden müssen. Jedoch hat in der Praxis diese Heterogenität innerhalb der IT-Landschaft zu einer kaum noch in den Griff zu bekommenden Inflexibilität und Komplexität geführt. (Berbner et al. 2005, S. 268)

Um zur Lösung dieses Problems beizutragen, wird im Zuge des immer schärfer werdenden Wettbewerbs auf dem globalen Markt versucht, dem Wettbewerb durch globale Zusammenarbeit zu begegnen. Eine wesentliche Rolle kommt dem Internet zu. Das Internet ermöglicht es, einst unternehmensintern lokal beschränkte Prozesse nun unternehmensübergreifend zu realisieren. (Kopp et al. 2006) Im Zuge der technischen Weiterentwicklung des Internets gewinnen Web Services enorm an Bedeutung und spielen in der Zukunftsplanung vieler Unternehmen eine essentielle Rolle. (Tian et al. 2004, S. 252)

Web Services sind das technologische Flaggschiff der serviceorientierten Architekturen (SOA) und haben die Entwicklung der heutigen Unternehmensanwendungen (Enterprise Applications) verändert. (Maamar et al. 2011, S. 90) Web Services ermöglichen es lokale Funktionen eines Geschäftsprozesses zu einem verteilten Geschäftsprozess zusammenzufassen und spielen daher in der Zukunft des Business Process Management (BPM) eine immer wichtigere Rolle. (Reynolds 2010, S. 73) Um derartige Geschäftsprozesse mit Web Services möglichst effizient zu unterstützen, ist es notwendig die qualitativ besten Web Services zu selektieren. Für diese Selektion der Web Services sind nicht-funktionale Zielgrößen entscheidend. Aus diesem Grund müssen daher geeignete Verfahren gefunden werden, welche eine Web Service Selektion anhand nicht-funktionaler Zielgrößen durchführt und zugleich die Präferenzen des Unternehmens beachtet. (Berbner 2007, S. 9)

1.2 Web Service Grundlagen

Die wichtigste Frage zu Beginn dieser Arbeit ist: Was ist ein Web Service und wofür wird dieser eigentlich verwendet? Um diese Frage zu beantworten, wird im einleitenden Kapitel 1.2.1 definiert, was ein Web Service ist, wofür er verwendet werden kann und worin die Vorteile bei der Verwendung von Web Services liegen. Unter Punkt 1.2.2 werden die Begriffe

Web Service und Web Service Komposition voneinander abgegrenzt. Kapitel 1.2.3 behandelt die Begriffe Geschäftsprozess und Workflow, da diese Begriffe bezüglich der Einsatzmöglichkeiten von Web Services grundlegend sind. In Kapitel 1.2.4 wird ein Überblick über einen der zentralen Begriffe dieser Arbeit gegeben, nämlich die nicht-funktionalen Zielgrößen, bevor dieser in Kapitel 3 nochmals genauer definiert wird. Ein weiterer zentraler Begriff dieser Arbeit, die Selektion von Web Services, wird in Kapitel 1.2.5 kurz erläutert. In Kapitel 3 wird die Web Service Selektion als Hauptteil dieser Arbeit nochmals genau ausgeführt. Einen Überblick über die Technologie, auf welcher Web Services basieren und funktionieren, wird in Kapitel 1.2.6 gegeben.

1.2.1 Was ist ein Web Service?

Web Services stehen im Zentrum dieser Arbeit. Aus diesem Grund wird zu Beginn der Arbeit in Kapitel 1.2.1.1 definiert was ein Web Service ist. Im Kapitel 1.2.1.2 werden die Vorteile und die Verwendungsmöglichkeiten von Web Services vorgestellt. Außerdem wird unter Kapitel 1.2.1.3 an Beispielen die Einsatzmöglichkeit von Web Services dargestellt.

1.2.1.1 Definition Web Service

Um die Entwicklung im Bereich des E-Business voranzutreiben, starteten IBM, Microsoft und SUN im Jahre 1999 die Entwicklung eines Konzepts, das die Kommunikation mit Hilfe verteilter Anwendungen über Unternehmensgrenzen ermöglicht. Dieses Konzept wird heute als „Web Services“ bezeichnet. Zwar wurden für den Datenaustausch vor 1999 schon verteilte Anwendungen eingesetzt wie bspw. Electronic Data Interchange (EDI), jedoch mussten die Teilnehmer dabei denselben Value Added Network (VAN) Service Provider verwenden, damit der Datentransfer abgesichert werden konnte. Diese Einschränkungen wurden abgebaut, wodurch es mittels Web Services nun möglich ist, entfernte spezifische Funktionalitäten zu nutzen, welche über das Internet aufgerufen werden können. (Bächle und Lehmann 2010, S. 31-32)

Aber was ist ein Web Service genau? Der Begriff Web Service ist in der Literatur nicht ganz einheitlich definiert und fällt je nach Sichtweise und je nachdem, unter welchen Aspekten der Begriff betrachtet wird unterschiedlich aus. In Folge dessen existiert eine Vielzahl an Definitionen. Als Beispiel werden an dieser Stelle die Definitionen von zwei der renommiertesten Institutionen aus dem Bereich der Information und Telekommunikation aufgeführt:

“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.” (W3C Working Group 2004b)

“A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software

systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.” (W3C Working Group 2004a)

“A software concept and infrastructure – supported by several major computing vendors (notably Microsoft and IBM) – for program-to-program communication and application component delivery. The Web services concept treats software as a set of services accessible over ubiquitous networks using Web-based standards and protocols.” (Gartner 2011)

Eine der treffendsten Definitionen liefert das Web Service Verzeichnis StrikeIron:

“A “Web service” is a specific term created by the industry to describe a function provided by one software system or application that is available for access by another software system or application over the Internet (or any network). This communication is achieved using a standardized mechanism such as text-based XML, eliminating the need for the communicating software applications to be running on the same hardware and software platforms. For example, a software system utilizing Windows .NET can communicate with a system built in Java on a Unix platform via a “Web service”.” (StrikeIron 2011)

Berbner et al. (2005) definiert Web Services als über Internet-Protokolle aufrufbare, lose gekoppelte Softwarekomponenten, welche mittels Austausch von Nachrichten untereinander kommunizieren. Web Services beruhen auf offenen XML-Standards wie SOAP (Simple Object Access Protocol), WSDL (Web Service Definition Language) und UDDI (Universal Description, Discovery and Integration). (Berbner et al. 2005, S. 268)

Es ist erkennbar, dass je nach Sichtweise Web Services etwas anders definiert bzw. charakterisiert werden. Aus jeder Definition ist jedoch erkennbar dass der Web Service eine IT-Technik darstellt, welche für die Interoperation zwischen Maschine und Maschine eingesetzt werden kann. Die Kommunikation ist standardisiert und erfolgt durch Softwaresysteme über das Internet.

Zusammengefasst erlaubt ein Web Service die Nutzung einer spezifischen Funktionalität einer entfernten Anwendung in einem internetbasierten Netzwerk. Aufgebaut und ermöglicht wird dies durch eine serviceorientierte Architektur (SOA). In dieser serviceorientierten Architektur müssen die Anwendungen die Funktionalitäten anderer im Netzwerk befindlichen Applikationen nutzen können. Web Services setzen XML-strukturierte Daten für die Übertragung ein. Daher ist es möglich durch Web Services eine plattformunabhängige Kommunikation in einer heterogenen Umgebung zu schaffen. (Bächle und Lehmann 2010, S. 32)

Bettag (2001, S. 302-303) hat die einzelnen Charakteristiken von Web Services zusammengefasst und wie folgt dargestellt:

Web Services sind programmierbar. Sie können über programmierbare Schnittstellen erreicht werden und sind zur Anwendungskommunikation entwickelt worden. Sie dienen nicht zur menschlichen Informationsverarbeitung und haben keine graphische Benutzeroberfläche.

Web Services sind selbstbeschreibend. Die Metadaten werden von anderen Web Services zur Auswertung genutzt. Durch den Namen, die Version und die Beschreibung des Web Services

erhalten andere Web Services Informationen über die Funktionalität. Die nicht-funktionalen Zielgrößen geben weitere Auskunft über die Qualität des Web Services.

Web Services sind gekapselt. Web Services erfüllen eine exakt definierte Aufgabe. Sie sind dabei unabhängig und stellen eine in sich geschlossene Anwendung dar.

Web Services sind lose gekoppelt. Implementierungsdetails können von den Web Service Anwendern und den Web Service Anbietern nicht eingesehen werden.

Web Services sind ortstransparent. Durch entsprechende Zugriffsrechte können Web Services von jedem Ort aus aufgerufen und ausgeführt werden. Sie können somit ortsunabhängig eingesetzt werden.

Web Services sind protokolltransparent. Web Service Operationen unterstützen unterschiedliche Protokolle der Internet Protokoll Suite, wie bspw. Hypertext Transfer Protokoll (http) oder Simple Mail Transfer Protokoll (SMTP).

Web Services sind komponierbar. Web Services können zu Kompositionen zusammengefügt werden. Damit entsteht ein neuer Web Service, welcher eine erweiterte Funktion erfüllen kann.

1.2.1.2 Vorteile der Verwendung von Web Services

Web Services bieten Unternehmen noch nie da gewesene Möglichkeiten zur Gestaltung ihrer Prozesse. Sie bieten die Möglichkeit einer vielseitigeren und flexibleren Zusammenarbeit innerhalb des Unternehmens und auch mit anderen Unternehmen. Die nahezu überall verfügbare und in hohem Maße standardisierte Web Service Technologie ermöglicht eine hochgradige Business to Business Interoperability (B2Bi). Geschäftsprozesse können durch miteinander in Verbindung stehenden Web Services (Web Service Kompositionen) durchgeführt werden und bei Bedarf jederzeit wieder verändert werden. (Zeng et al. 2004, S. 311; Reynolds 2010, S. 72) Die Web Services werden unabhängig von ihrer Client Applikation entwickelt, typischerweise von verschiedenen Entwicklern, und sie agieren in einer offenen Umgebung, nämlich dem Web, wo Wettbewerb und Differenzierung die Hauptmerkmale sind. (Zeng et al. 2004, S. 312)

1.2.1.3 Beispiele zur Einsatzmöglichkeit von Web Services

Durch die Vielfältigkeit der bereits existierenden Web Services können diese für nahezu alle Dienste im Web eingesetzt werden. Die Funktionalität der Web Services reicht von Wetterdiensten über Übersetzungshilfen hin bis zu anspruchsvollen Web Services für Bezahlssysteme oder Web Services für Banken. Wirft man einen Blick in das Web Service Verzeichnis seekda! (<http://webservices.seekda.com>), steht an erster Stelle der meistgenutzten Web Services ein kostenloser Web Service für Wetterinformationen zu Großstädten auf der ganzen Welt. Dieser Web Service kann auf einer Internetseite integriert werden und informiert die Besucher der Webseite kostenlos über das Wetter. Allein im seekda! Web Service Verzeichnis stehen

22.272 Web Services zur Auswahl (Stand 11.08.2011). Bereits heute existieren eine Fülle an Web Service Anbietern und Web Service Dienstverzeichnissen. Im Kapitel 5 werden weitere Web Service Anbieter und Web Service Dienstverzeichnisse aufgelistet.

Ein konkretes Beispiel für die Einsatzmöglichkeit von Web Services bietet die SCHUFA Holding AG aus Wiesbaden. Die SCHUFA Holding AG ist im Besitz von 479 Millionen Datensätzen über kreditrelevante Informationen zu 66,2 Millionen Bürgern in Deutschland. Das Unternehmen arbeitet mit 6000 Vertragspartnern aus dem Bereich Banken, Versicherungen, Handel etc. in Deutschland zusammen und bietet ihnen Informationen als Basis zur Kreditvergabe d. h. über die Kreditwürdigkeitsprüfung ihrer Kunden. Jährlich werden ca. 102,9 Millionen Auskünfte an die Vertragspartner erteilt und über eine Million Eigenauskünfte bedient. Um diese Auskünfte über kreditrelevante Informationen ihren Vertragspartnern einfacher zugänglich zu machen, bietet die SCHUFA Holding AG eine Reihe verschiedener Web Service Produkte aus dem Bereich B2C (Business-to-Consumer) und B2B (Business-to-Business) an. So ist neben Produkten zu Kreditwürdigkeitsprüfungen ein spezielles Produkt der SCHUFA aus dem Bereich B2C der SCHUFA-KontonummernCheck. Der KontonummernCheck ermöglicht es Unternehmen aus dem Bereich eCommerce, Handel und Versandhandel, die von Kunden eingegebenen Daten zur Bankverbindung auf Qualität hin zu prüfen. Dabei wird die Plausibilität der Kontodaten überprüft und außerdem Auskunft erteilt, ob das Konto wirklich existiert und ob der angegebene Name tatsächlich der Kontoinhaber ist. Im Detail werden durch die SCHUFA nach Übermittlung der Personenstammdaten und Bankverbindung des Kunden des Vertragspartners die übermittelten Daten mit dem Datenbestand der SCHUFA abgeglichen. Anschließend wird dem Vertragspartner das Ergebnis der Überprüfung mitgeteilt. Das Ziel dieser Datenauskunft über einen Web Service ist, eine schnellere Auskunft und einen qualitativen Datenabgleich zu erhalten. Durch die unmittelbare Antwort können insbesondere im Handel unabsichtliche Falscheingaben oder Betrugsversuche der Kunden unterbunden werden. Das Geschäftsrisiko des Geschäftspartners der SCHUFA wird somit reduziert und im Hinblick auf mögliche Folgekosten, wie sie etwa durch Rücklastschriften entstehen, können die Vertragspartner Kosten senken. (SCHUFA Holding AG 2011) Ein weiteres Web Service Angebot der SCHUFA, die Kreditwürdigkeitsprüfung, wird in Kapitel 3.2.1 im Rahmen eines Beispiels zu einem Web Service Selektionsverfahren vorgestellt.

1.2.2 Web Service - Web Service Komposition

Wie in Kapitel 1.2.1.1 bereits angedeutet, können die spezifischen Funktionen einzelner Web Services zu einer Web Service Komposition zusammengeführt werden. Durch Web Service Komposition entsteht eine neue Funktion, mit deren Hilfe Geschäftsprozesse durchgeführt werden können. (Ma und Zhang 2008, S. 1093; Canfora et al. 2004, S. 36) Notwendig ist eine Web Service Komposition, weil einzelne Web Services nicht komplette Prozesse abbilden können. Durch Aggregation von Web Services können komplexere Prozesse abgebildet werden und auf einem höheren Level abstrahiert werden. An dieser Stelle entsteht der große Mehrwert von Web Services. Mit Web Services, eingebettet in einer Komposition, können Geschäftsprozesse mit einer höheren Flexibilität ausgeführt, umgesetzt und außerdem bei Be-

darf wieder verändert werden. Graphisch darstellen lassen sich Web Service Kompositionen bspw. durch UML-Aktivitätsdiagramme, Petri-Netze etc. (Dustdar und Schreiner 2005, S. 3-4; Heinrich et al. 2011b, S. 89-93)

Eine Web Service Komposition wird oftmals in der Literatur ebenfalls Web Service genannt. In dieser Arbeit wird der Begriff jedoch nicht synonym verwendet. Ein Web Service stellt einen einzelnen funktional genau definierbaren Web Service dar, welcher in einer Web Service Komposition zum Einsatz kommen kann. Web Service Kompositionen und die Erstellung einer Web Service Komposition werden in Kapitel 4 im Zuge der Web Service Selektion näher erläutert.

1.2.3 Geschäftsprozesse und Workflows

Da Web Services auch in Unternehmen zum Einsatz kommen, müssen im Rahmen dieser Arbeit die Begriffe Geschäftsprozess und Workflow definiert und voneinander abgegrenzt werden. In den folgenden zwei Kapiteln werden daher beide Begriffe definiert.

1.2.3.1 Geschäftsprozess

Geschäftsprozesse bezeichnen Prozesse, welche zur Erfüllung der obersten Geschäftsziele einer Unternehmung beitragen und dabei das zentrale Geschäftsfeld beschreiben. (Krcmar 2010, S. 142)

Von der Workflow Management Coalition (WFMC) wird ein Geschäftsprozess folgendermaßen definiert:

„A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.“ (Workflow Management Coalition (WFMC) 1999)

Demnach sind Geschäftsprozesse eine Menge an verlinkter Aktivitäten, die zusammen zur Erreichung der obersten Geschäftsziele beitragen. Bei den Aktivitäten handelt es sich um die Beschreibung einer Teilaufgabe; sie stellen logische Prozessschritte dar. (Workflow Management Coalition (WFMC) 1999) Ein Prozessschritt kann in den drei Dimensionen Ressourcen, Aufgaben und Anwendungsfälle ausgeprägt sein. So ist bspw. in einem Versicherungsunternehmen das Senden einer Empfangsbestätigung (Aufgabe) durch einen Mitarbeiter (Ressource) nach Empfang einer Schadensmeldung (Anwendungsfall) eine Aktivität. (Bebner 2007, S. 30) In der Wirtschaftsinformatik wird ein Geschäftsprozess als eine inhaltlich abgeschlossene, zeitlich-sachlogische Abfolge von Funktionen definiert, welche zur Bearbeitung eines für die Leistungserbringung des Unternehmens relevanten Objekts notwendig sind. Graphisch dargestellt werden Geschäftsprozesse durch ein Geschäftsprozessmodell. Ein Geschäftsprozessmodell ist eine vereinfachte, zweckorientierte Abbildung, deren Struktur die zeitlich-sachlogische Abfolge der betrachteten Aktion darstellt. (Mertens et al. 2001, S. 210-211)

Geschäftsprozesse sind aber nicht nur innerhalb eines Unternehmens zu finden. Sie laufen auch funktions-, hierarchie-, standort- und unternehmensübergreifend ab. Beschrieben werden sie durch ein definiertes Anfangsereignis und ein definiertes Endereignis. Sie erfordern außerdem Eingaben, welche verarbeitet werden, und produzieren ein Ergebnisse die mit den Geschäftszielen vereinbar sein müssen. (Berbner 2007, S. 30) Prozesse können auch organisationsübergreifend gestaltet sein. Durch organisationsübergreifende Prozesse können mehrere Unternehmen zu einem Wertschöpfungsnetzwerk zusammengefasst werden. Da Unternehmen in ihrer Tätigkeit weitgehend spezialisiert sind, sind diese Wertschöpfungsnetzwerke für die Unternehmen erfolgsentscheidend und damit gleichzeitig ein kritischer Erfolgsfaktor. (Berbner 2007, S. 31) Im Zuge der Spezialisierung und auch Globalisierung hat insbesondere das Business Process Outsourcing in den vergangenen Jahren enorm an Bedeutung gewonnen. (Braunwarth und Heinrich 2008, S. 98; Berbner 2007, S. 31) Ein Beispiel für einen Geschäftsprozess stellt die Abbildung 3-2 in Kapitel 3.1.3 dar.

1.2.3.2 Workflow

Workflows bezeichnet Geschäftsprozesse, die informationstechnisch realisiert werden und damit gleichzeitig (teil-)automatisiert sind. (Berbner 2007, S. 32) Die Workflow Management Coalition (WFMC) definiert einen Workflow folgendermaßen:

„The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.“ (Workflow Management Coalition (WFMC) 1999)

Um Workflows zu realisieren und zu steuern, benötigt es ein Workflow Management System, das ebenfalls von der Workflow Management Coalition (WFMC) wie folgt definiert wird:

“A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.“ (Workflow Management Coalition (WFMC) 1999)

D. h. ein Workflow Management System ist ein anwendungsneutrales, generisches Werkzeug, welches eingesetzt wird, wo geplante, strukturierte und aufgeteilte Arbeitsabläufe durchgeführt werden. (Berbner 2007, S. 32)

Eine speziellere Form von Workflows sind serviceorientierte Workflows. Serviceorientierte Workflows stellen die Komposition von (Web) Services im Kontext einer SOA dar. Durch den sequentiellen Aufruf von Services ist es möglich die Funktionalität vorhandener einzelner Services zu verbinden. (Berbner 2007, S. 33) Speziell im Kontext dieser Arbeit wird dies durch Web Service Kompositionen dargestellt.

1.2.4 Web Service Selektion/ Selektionskriterien

Um den Anforderungen des Web Service Nutzers möglichst exakt zu entsprechen, muss ein Web Service oder eine Web Service Komposition anhand aussagekräftiger Kriterien selektiert werden können. Dabei müssen entsprechend der Problemstellung die bestmöglichen Web Services selektiert werden. Für die Selektion von Web Services kann dies anhand nicht-funktionaler Zielgrößen gemacht werden. Denn einzelne Web Service Angebote, aber auch Web Service Kompositionen lassen sich in ihrer Qualität durch nicht-funktionale Zielgrößen evaluieren. Dazu werden von den Anbietern der Web Services mittels Service Level Agreements (SLAs) dem Nutzer Garantien gegeben. Ein SLA ist eine Dienstleistungsverpflichtung des Web Service Anbieters an den Kunden. Mit SLAs werden rechtsverbindliche Verträge zwischen Web Service Anbieter und Nutzer unter anderem über die nicht-funktionalen Zielgrößen festgelegt. Web Service Angebote ohne Garantien zu den nicht-funktionalen Zielgrößen, wie beispielsweise Antwortzeit, Sicherheit, Verfügbarkeit, Zuverlässigkeit, Ausführungskosten etc., können nur in einfachen Fällen zur Anwendung kommen. Beim Einsatz von Web Services innerhalb bedeutender Anwendungen oder einer Komposition von Web Services jedoch müssen die Anbieter Garantien geben bzw. die nicht-funktionalen Zielgrößen müssen bekannt sein. (Menascé 2002, S. 72-73; Berbner et al. 2005, S. 269) Wie schon erwähnt existiert in der Praxis eine ganze Reihe an funktional identischen Web Services, daher werden für die Web Service Selektion nicht-funktionale Zielgrößen immer bedeutender. Die nicht-funktionalen Zielgrößen können unterschiedlichster Ausprägung sein. Wesentliche Selektionskriterien sind bspw. die Ausführungskosten und die Ausführungszeit eines Web Services. Alle nicht-funktionalen Zielgrößen werden im Kapitel 2 detailliert beschrieben.

Einen Prozess mithilfe von nicht-funktionalen Kriterien zu beschreiben und zu gestalten generiert vier unterschiedliche Vorteile. Der erste Vorteil ist folgender: Weil ein Prozess anhand nicht-funktionaler Zielgrößen gestaltet werden kann, können Unternehmen ihre Unternehmensvisionen wesentlich leichter in ihre Geschäftsprozesse einbringen. Der zweite Vorteil ergibt sich aus der Möglichkeit, Prozesse anhand der nicht-funktionaler Zielgrößen zu selektieren und auszuführen. Besonders wenn unternehmenskritische und komplexe Prozesse durchgeführt werden, sind nicht-funktionale Zielgrößen der entscheidende Indikator für die Erfolgsmessung und um die Anforderungen der Nutzer erfüllen zu können. Das Monitoring der Prozesse basierend auf nicht-funktionalen Zielgrößen ist der dritte Vorteil der Gestaltung von Prozessen anhand nicht-funktionaler Zielgrößen. Die zum Einsatz kommenden Web Services müssen kontinuierlich und streng überwacht werden, um die Übereinstimmung zwischen den Anforderungen an nicht-funktionale Zielgrößen und den tatsächlichen nicht-funktionalen Zielgrößen zu gewährleisten. Das Monitoring der nicht-funktionalen Zielgrößen erlaubt es Anpassungen zu machen, wenn unerwünschte Ereignisse eintreffen oder kritische Grenzwerte erreicht werden. Der vierte und letzte Vorteil ist, dass im Falle von notwendigen Anpassungen alternative Web Services evaluiert und ausgeführt werden können. Üblicherweise wirken viele verschiedene Faktoren aus dem Umfeld auf die Strategie, die Methodologie und die Struktur der Geschäftsprozesse ein. Um mögliche Zeitverzögerungen, unerwartete Verläufe oder technische Unregelmäßigkeiten zu vermeiden, ist es notwendig die Web Service Kompositionen bspw. bei einer Zeitverzögerung anzupassen bzw. neu zu planen (Re-planning), damit die

Prozesse gemäß den Anforderungen abgeschlossen werden können. (Cardoso et al. 2004) Das Re-planning wird im Kapitel 3.3 nochmal ausführlicher behandelt.

1.2.5 Serviceorientierte Architektur (SOA)

In der heutigen Zeit müssen Unternehmen in immer kürzeren Zyklen neue Geschäftsprozesse adaptieren. Des Weiteren müssen sie auf Änderungen der Unternehmenskultur flexibel reagieren und stehen im Prinzip ständig unter einem Kostendruck. Um diese Herausforderungen bewältigen zu können, bieten sich serviceorientierte Architekturen als Lösung an. SOA ist ein Architekturmuster, mit dem eine Anwendungslandschaft aus einzelnen Anwendungsbausteinen aufgebaut und beschrieben werden kann. Die Anwendungsbausteine können eine festgelegte fachliche Aufgabe übernehmen. Die Funktionalitäten der Anwendungsbausteine werden in Form von Services dargestellt und sind lose gekoppelt. Als Service wird dabei eine fest definierte Leistung gesehen, welche in einem oder vielen übergeordneten Prozessen eingesetzt werden kann. Durch einen einheitlichen Mechanismus werden die Services aufgerufen und verbinden somit die Anwendungsbausteine plattformunabhängig. (Richter et al. 2005, S. 413)

Eine Definition des Begriffes SOA liefert die Firma Sun Microsystems, Inc. Demnach ist eine SOA „an architectural style that emphasizes well-defined, loosely coupled, coarse-grained, business-centric, reusable shared services.“ (Sun Microsystems, Inc. 2004, S. 3)

Die entscheidenden Vorteile der SOA für Web Service Entwickler sind folgende: Eine SOA ermöglicht eine kürzere Produkteinführungszeit (Time to market) für neue Web Services. Die IT-Architekten und Entwickler können sich verstärkt auf die Entwicklung und Lieferung des eigentlichen Produkts fokussieren und weniger auf die Middleware. Der zweite Vorteil ist die Reduzierung der Gesamtbetriebskosten (Total cost of ownership). Es muss kein teure fremdbezogene Middleware angeschafft werden. Stattdessen können offene leistungsfähige Standards eingesetzt werden. Außerdem können einzelne Geschäftsfunktionen als Web Service abgebildet werden, welche auch von anderen Unternehmen genutzt werden können. (Sun Microsystems, Inc. 2004, S. 4)

Web Services stellen eine Umsetzung des Architekturparadigmas SOA dar, können aber nicht mit einer SOA gleichgesetzt werden. Eine SOA kann als fachliches Architekturmuster gesehen werden, welches technologieunabhängig verwendet wird. (Richter et al. 2005, S. 413) Der Web Service Technologie zugrunde liegt das Paradigma der serviceorientierten Architektur, was durch die lose Kopplung und die Ortsunabhängigkeit der Web Services deutlich wird. (Berbner 2005, S. 269)

1.2.6 Web Service Technologie

Die Web Service Technologie baut auf dem Paradigma der serviceorientierten Architektur auf. (Berbner et al. 2005, S. 269) Wie schon in der Definition des Begriffes Web Service erwähnt, zeichnen sich Web Services durch Interoperabilität aus. Damit diese Interoperabilität erreicht wird, werden XML-basierte Standards eingesetzt, welche Internetprotokolle wie http,

smtp, ftp, ect. unterstützen können. Die eingesetzten Standards sind WSDL, SOAP und UDDI und stellen gleichzeitig die Kerntechnologie der Web Services dar. (Berbner 2007, S. 18-19; Kossmann und Leymann 2004, S. 120) Die Abbildung 1-1 stellt eine Übersicht dar, welche Protokolle und Standards für die Web Service Technologie eingesetzt werden.

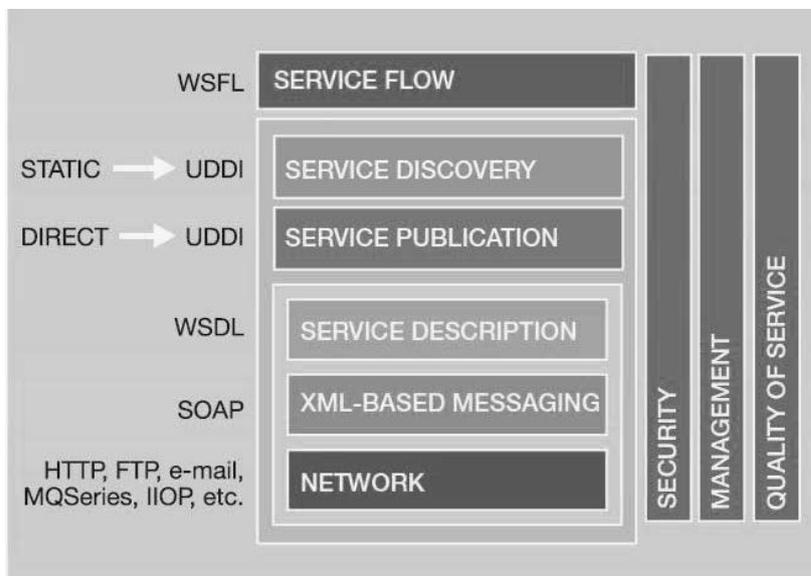


Bild 1-1 Web Service Programming Stack (Gottschalk et al. 2002, S. 171)

Die Abbildung 1-1 zeigt die Netzwerkschicht als Grundlage für das Web Service Programming Stack. Die Netzwerkschicht stellt sicher, dass Web Services über alle Netzwerke hinweg verfügbar sind. Wie in Kapitel 1.2.1.1 schon beschrieben, sind Web Services protokolltransparent und unterstützen verschiedene Protokolle der Internet Protokoll Suite. In den meisten Fällen basiert das Netzwerk auf http, jedoch auch andere Netzwerkprotokolle wie ftp oder MQSeries von IBM können verwendet werden. (Gottschalk et al. 2002, S. 171)

In der darüber liegenden Schicht können durch das XML-basierte SOAP (Simple Object Access Protocol) andere Systeme (Clients) mit dem Web Service kommunizieren. Durch SOAP wird das XML-Kommunikationsprotokoll definiert, welches den Aufbau der Nachrichten zur Kommunikation mit Web Services spezifiziert. Dabei werden XML-Dokumente ausgetauscht, die die Veröffentlichung, das Finden, die Bindung und den Aufruf der Web Services ermöglichen. (Berbner 2007, S. 19; Gottschalk et al. 2002, S. 171)

Mit WSDL (Web Service Description Language) wird die Beschreibung der verfügbaren Web Services für andere Systeme spezifiziert. Explizit werden mit WSDL die Schnittstellen beschrieben, welche die plattformunabhängige Kommunikation zwischen Web Services und anderen Systemen regeln. WSDL baut auf SOAP auf. Dem Client werden technische Daten und Nutzungsdetails des angeforderten Web Services zur Verfügung gestellt. Web Service Anbieter könne mit Hilfe von WSDL ihre Web Services anbieten. (Krcmar 2010, S. 347; Berbner 2007, S. 21)

Web Services müssen neben der Möglichkeit, sie mittels SOAP zu nutzen und mit WSDL Informationen über ihn zu erhalten, im Internet auch gefunden werden können. Diese Aufgabe wird durch den von Microsoft, IBM und Ariba eingeführten de-facto-Standard UDDI (Universal Description, Discovery and Integration) übernommen. Durch UDDI können Unternehmen ihre eigene Web Services anbieten und nach anderen Web Services suchen. Ähnlich wie die gelben Seiten eines Telefonbuches ist UDDI ein Verzeichnis, welches verfügbare Web Services katalogisiert. Die UDDI-Verzeichnisdienste können in öffentliche und private unterteilt werden. Während der Zugriff auf private Verzeichnisdienste nur bestimmten Kunden und Geschäftspartnern möglich ist, sind öffentliche Verzeichnisdienste jedem frei zugänglich und können durchsucht werden. (Krcmar 2010, S. 347; Berbner 2007, S. 24; Kossmann und Leymann 2004, S. 122)

Die folgende Abbildung 1-2 zeigt den beschriebenen Sachverhalt nochmals anhand der Web Service Architektur. Darin ist zu erkennen, wie SOAP, WSDL und UDDI die Kerntechnologie der Web Service Architektur bilden und damit den Einsatz von Web Services ermöglichen. (Bächle und Lehmann 2010, S. 32)

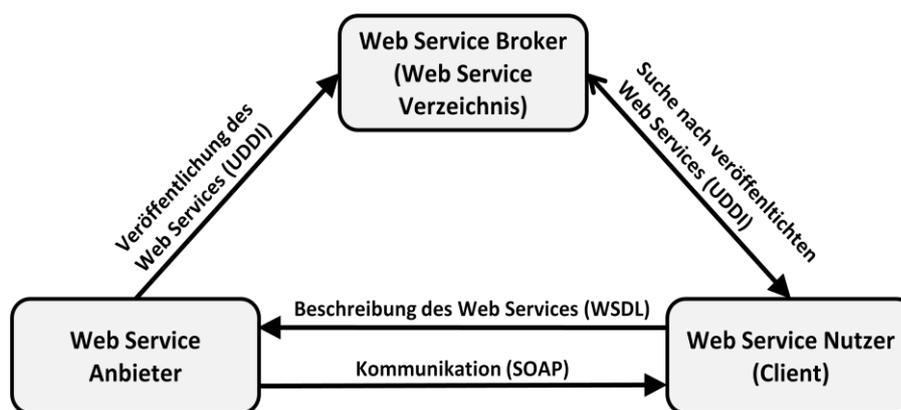


Bild 1-2 Aufbau einer Web Service Architektur in Anlehnung an Bächle und Lehmann (2010, S. 32)

1.3 Forschungsfrage und erwartetes Ergebnis

Aufgrund der Anforderungen an nicht-funktionale Zielgrößen eines einzelnen Web Services oder einer Web Service Komposition ist es wichtig, die richtige Auswahl der einzelnen Web Services zu treffen. Für jeden einzelnen Web Service gibt es alternative Web Services mit gleicher Funktionalität, jedoch mit unterschiedlichen nicht-funktionalen Zielgrößen. (Ma und Zhang 2008, S. 1094) Das Ziel ist es, gemessen an den nicht-funktionalen Zielgrößen, zu jedem Zeitpunkt die optimale Servicekette zu ermitteln und zu realisieren. (Patel et al. 2003, S. 826-830)

Mittels dieser Arbeit werden daher ein strukturierter Vergleich und eine Bewertung verschiedener Verfahren geschaffen, die eine (automatische) Selektion anhand von nicht-funktionalen Zielgrößen durchführen. Außerdem soll gezeigt werden, in welchem Umfang einzelne nicht-

funktionale Zielgrößen in der Praxis dokumentiert werden und somit als Kriterien zur Selektion verwendet werden können.

1.4 Gliederung der Arbeit

Die Arbeit ist in drei Hauptteile gegliedert. Der erste Teil der Arbeit behandelt die nicht-funktionalen Zielgrößen. Dabei sollen die die am häufigsten vorkommenden nicht-funktionalen Zielgrößen näher erläutert werden; damit soll Verständnis darüber vermittelt werden, weshalb nicht-funktionale Zielgrößen das entscheidende Auswahlkriterium sind. Im zweiten Teil der Arbeit wird auf die Selektion der Web Services eingegangen. Hierbei werden neben unterschiedlichen Selektionsverfahren auch die Herausforderungen und Schwierigkeiten der Web Service Selektion diskutiert. Ziel ist es, dem Leser zu verdeutlichen, wie eine Web Service Selektion gemacht wird, welche Herausforderungen an die Selektion geknüpft sind und welche Selektionsverfahren in der Praxis existieren. Des Weiteren wird in Kapitel 4 anhand konkreter Web Service Selektionsverfahren die Selektion einer Web Service Komposition dargestellt und berechnet. Der dritte Teil dieser Arbeit behandelt die Dokumentation der nicht-funktionalen Zielgrößen in der Praxis, dabei soll vor allem der Unterschied zwischen Theorie und Praxis verdeutlicht werden. Dabei werden die Quantität und Qualität der Dokumentation nicht-funktionaler Zielgrößen durch Web Service Anbieter und Web Service Dienstverzeichnisse überprüft und dargestellt.

2 Nicht-funktionale Zielgrößen

In diesem Kapitel wird zunächst der Begriff „nicht-funktionale Zielgrößen“ näher beschrieben. Zum besseren Verständnis wird anschließend eine Unterscheidung und Abgrenzung zu funktionalen Zielgrößen gemacht. In Kapitel 2.3 werden die wichtigsten nicht-funktionalen Zielgrößen, mit deren Hilfe die Qualität und Güte von Web Services messbar und dargestellt wird, aufgeführt und erläutert.

2.1 Definition

Für den Begriff der nicht-funktionalen Zielgröße werden in der Literatur verschiedene Begriffe verwendet, am Häufigsten in der deutschsprachigen Literatur die Begriffe Dienstgütekriterien, Dienstgüteeigenschaften, nicht-funktionale Zielgrößen und nicht-funktionale Eigenschaften. In der englischsprachigen Literatur werden die Begriffe Quality of Service Attributes (Abk. QoS Attributes), Non-functional Requirements, Non-functional Constraints oder Quality Attributes verwendet. In dieser Arbeit wird ausschließlich der Begriff nicht-funktionale Zielgrößen verwendet. Er steht damit gleichzeitig für alle oben genannten Begriffe.

In Berbner (2007, S. 35) wird der Begriff Dienstgüte als Überbegriff für alle nicht-funktionalen Zielgrößen angegeben. Dieser Begriff ist aber nicht einheitlich definiert. So hängt die Definition stark davon ab, unter welchen Aspekten welche Art von Dienst betrachtet wird. Folglich setzt sich die Bewertung der Dienstgüte aus unterschiedlichen Gesichtspunkten zusammen. (Berbner 2007, S. 35) Im Folgenden wird der Begriff Dienstgüte nicht verwendet.

Nicht-funktionale Eigenschaften beinhalten Fähigkeiten und Beschränkungen eines Web Services. Fähigkeiten sind Verhalten bzw. Charaktereigenschaften eines Web Services, also jene Eigenschaften, die Interessensgruppen betreffen und die Zufriedenheit der Interessenten mit dem Web Service beeinflussen. Beschränkungen können von den Entwicklern eines Web Services nicht oder sehr wenig beeinflusst werden und sind als feste Größen gegeben. So beziehen sich Beschränkungen zum einen auf die Rahmenbedingungen, die die Web Service Technologie vorgibt, bspw. das Maximum an Sicherheitstechnologie für Web Services, um Missbrauch zu vorzubeugen, zum anderen auf die Fähigkeiten der Entwickler der Web Services. (Malan und Bredemeyer 2001, S. 2)

In weiterer Folge unterscheiden Malan und Bredemeyer (2001, S. 3) nicht-funktionale Eigenschaften in Laufzeiteigenschaften und Entwicklungszeiteigenschaften. Laufzeiteigenschaften entsprechen den Zielen der Nutzer bzw. sind in deren Interesse. Sie verdeutlichen, mit welchem Zufriedenheitsgrad die funktionalen Eigenschaften umgesetzt werden. Entwicklungszeiteigenschaften sind vorrangig für die Entwickler der Web Services relevant, denn sie beeinflussen die Entwicklungskosten oder Anstrengungen, die gemacht werden müssen, um einen Web Service zu entwickeln. Ein Beispiel für Entwicklungszeiteigenschaften ist die Fähigkeit, in den Web Service zukünftig weitere Funktionen zu integrieren. (Malan und Brede-

meyer 2001, S. 2-3) Entwicklungszeiteigenschaften sind nicht Gegenstand dieser Arbeit und werden im Folgenden nicht weiter beachtet.

2.2 Unterscheidung zwischen funktionalen und nicht-funktionalen Zielgrößen

Bevor im Kapitel 2.3 die nicht-funktionalen Zielgrößen einzeln aufgeführt und spezifiziert werden, ist eine Unterscheidung zu funktionalen Eigenschaften notwendig. Funktionale Eigenschaften beschreiben das spezifische Verhalten, die Funktion oder Dienstleistung eines Systems, einer Softwareanwendung oder im Kontext dieser Arbeit eines Web Service. Sie verdeutlichen, was genau ein System wie ein Web Service machen soll. Aus Sicht des Anwenders tragen funktionale Eigenschaften dazu bei, den Zweck der Tätigkeit des Anwenders zu erfüllen. Nicht-funktionale Eigenschaften beziehen sich auf die Qualität, Anforderungen und Einschränkungen eines Web Service. Sie machen deutlich, in welchem Grad die funktionalen Eigenschaften umgesetzt werden. Sie haben keinen Einfluss auf die Funktionalität bzw. funktionalen Eigenschaften. (Malan und Bredemeyer 2001, S. 2-3)

2.3 Spezifizierung und Ausführung der einzelnen nicht funktionalen Zielgrößen

Web Services können abhängig von ihrer Funktionalität in verschiedenen Umgebungen eingesetzt werden. Je nachdem, unter welchem Aspekt man einen Web Service betrachtet, z. B. der Funktionalität, der Eigenschaft als Softwarekomponente oder der Eigenschaft als Komponente eines Netzwerkes, können unterschiedliche nicht-funktionale Zielgrößen definiert werden. Eine genaue Menge an nicht-funktionalen Zielgrößen ist folglich nicht bestimmbar. (Berbner 2007, S. 36-37)

In der Vergangenheit haben sich einige Forschungsarbeiten mit dem Thema nicht-funktionale Zielgrößen mit Bezugnahme zu Web Services befasst. Jedoch sind die Einteilung und die Kategorisierung und der Umfang der nicht-funktionalen Zielgrößen nicht in allen Arbeiten identisch. So kategorisieren Patel et al. (2003, S. 830-831) nicht-funktionale Zielgrößen in drei Bereiche: Generelle nicht-funktionale Zielgrößen, Internetservice-spezifische nicht-funktionale Zielgrößen und aufgabenspezifische nicht-funktionale Zielgrößen. Zu den generellen nicht-funktionale Zielgrößen gehören zum Beispiel die Kosten der Nutzung des Web Services, die Zuverlässigkeit des Services, die Wartezeit oder der Durchsatz. Internetservice-spezifische nicht-funktionale Zielgrößen sind bspw. die Verfügbarkeit des Web Services, die Sicherheit oder die Zugänglichkeit des Web Services. Aufgabenspezifische nicht-funktionale Zielgrößen beziehen sich auf die Qualität des Ergebnisses oder die Aufgabenerfüllung des Web Services. Abhängig von einer Wahrscheinlichkeitsfunktion wird bestimmt, in welchem Grad das Ergebnis dem gewünschten Ergebnis entspricht.

Ran (2003, S. 6-9) teilt nicht-funktionale Zielgrößen in vier Kategorien ein. Die erste Kategorie sind die laufzeitbezogenen nicht-funktionalen Zielgrößen, die die Skalierbarkeit, die Kapazität, Zuverlässigkeit, Verfügbarkeit, Robustheit/Flexibilität, Genauigkeit, Ausnahmebehandlung und die Performanz, welche in Durchsatz, Latenz und Antwortzeitaufgeteilt ist, umfassen. Die zweite Kategorie der nicht-funktionalen Zielgrößen bezieht sich auf die Transaktionsunterstützung. Hierzu gehört die Integrität, deren Hauptaufgabe es ist, bei der Datenverarbeitung oder Übertragung die Datenintegrität sicherzustellen. Zu der dritten Kategorie gehören die konfigurationsmanagement- und kostenbezogenen nicht-funktionale Zielgrößen. Unter dieser Kategorie sind die Regulierung der Web Services, die Unterstützung von geltenden Standards, Stabilität und Änderungszyklus, die Kosten und die Vollständigkeit zu finden. Die vierte und letzte Kategorie betrifft die sicherheitsbezogenen nicht-funktionalen Zielgrößen. Damit werden die Authentifizierung, Autorisierung, Vertraulichkeit, Haftung, Rückverfolgbarkeit/ Nachvollziehbarkeit, Datenverschlüsselung und Nachweisbarkeit angesprochen. Die einzelnen nicht-funktionalen Zielgrößen werden in diesem Kapitel noch näher beschrieben.

Grundsätzlich sind für Web Services relevante nicht-funktionale Zielgrößen in qualitative und quantitative Zielgrößen zu unterscheiden. (Berbner 2007, S. 36-37) Nach der Definition von Malan und Bredemeyer (2001, S. 3) zählen beide Arten zu den Laufzeiteigenschaften.

In der Literatur nicht einheitlich ist die Einteilung der Kosten. Anders als Berbner (2007, S. 36-39) und Berbner et al. (2005, S. 270-271) setzen bspw. Canfora et. al. (2005, S. 1069-1075) und Ran (2003, S. 6-9) die Kosten allen anderen nicht-funktionalen Zielgrößen gleich. Berbner (2007, S. 36-39) klassifiziert nicht-funktionale Zielgrößen in grundsätzlich zwei Kategorien, in Dienstgüte und Kosten. Bei gleicher Funktionalität des Web Services stellen die Kosten die erste eigene nicht-funktionale Zielgröße dar. Die Dienstgüte als zweite Kategorie lässt sich weiter in qualitative und quantitative nicht-funktionale Zielgrößen aufteilen. Qualitative Zielgrößen sind die Reputation und die Sicherheit eines Web Services. Unter quantitative Zielgrößen fallen die Performance, Verfügbarkeit, Skalierbarkeit, Robustheit, Zuverlässigkeit und die Fehlerrate eines Web Services.

In dieser Arbeit sind nicht-funktionale Zielgrößen den Dienstgütekriterien gleichgesetzt, d. h. in weiterer Folge werden die Kosten als Teil der Dienstgüte, also der nicht-funktionalen Zielgrößen behandelt und stellen eine quantitative Größe dar.

Die in der Literatur am häufigsten genannten nicht-funktionalen Zielgrößen hat Berbner (2007, S. 36-39) zusammengefasst. Die Abbildung 2-1 zeigt eine abgewandelte Zusammenfassung der nicht-funktionalen Zielgrößen, welche Berbner (2007, S. 36-39) erfasst hat.

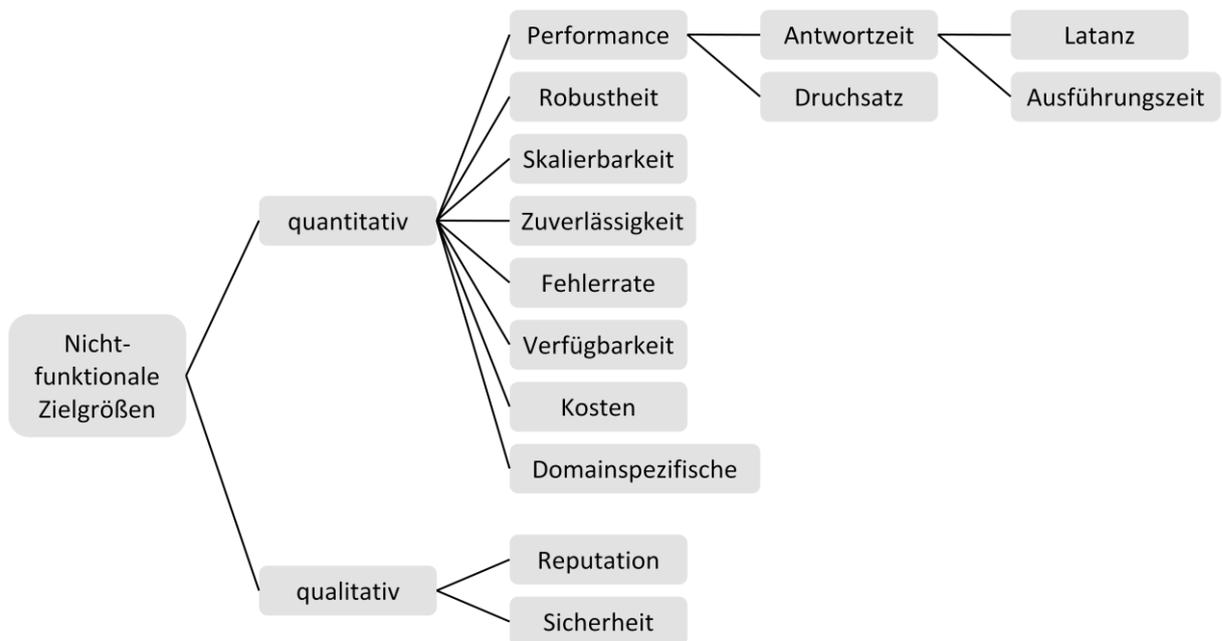


Bild 2-1 Kategorisierung der nicht-funktionalen Zielgrößen
in Anlehnung an Berbner (2007, S. 37)

Die wichtigsten und am häufigsten vorkommenden nicht-funktionalen Zielgrößen aus der Abbildung 2-1 werden in den folgenden Tabellen 2-1, 2-2, 2-3 beschrieben.

Nicht-funktionale Zielgröße	Beschreibung
Performance	Die Performance ist ein übergeordnetes Kriterium und setzt sich aus Antwortzeit und Durchsatz zusammen.
Durchsatz	Der Durchsatz wird gemessen, indem die Anzahl der bedienten Serviceaufrufe innerhalb einer gegebenen Zeitspanne gemessen wird. (Patel et al. 2003, S. 830-831)
Antwortzeit	Die Antwortzeit ist die Dauer, die vom Absenden einer Anfrage bis zum Erhalt der Antwort vergeht. Die Antwortzeit beinhaltet die Latenz und die Ausführungszeit. (Menascé2002, S. 72)
Latenz	Unabhängig von der Größe der Anfrage und der auszuführenden Operationen des Web Services ist die Latenz die Dauer, die mindestens vergeht, um einen Web Service Aufruf zu bedienen. Die sogenannte minimale Round-Trip-Time ist determiniert durch eine kleinstmögliche Nachricht, die vom Nutzer zum Web Service und wieder retour gesendet wird. (Berbner 2007, S. 38; Kalepu et al. S. 132)
Ausführungszeit	Die Ausführungszeit ist die Dauer, die ein Web Service zur kompletten Ausführung einer Anfrage benötigt. D. h. es wird die Zeitspanne zwischen dem Erhalt der Anfrage und dem Absenden der Antwort gemessen. In einer Web Service Komposition wird die maximale Ausführungszeit vom kritischen Pfad bestimmt. (Zeng et al. 2004, S. 315-316)
Robustheit	Die Robustheit zeigt die Fähigkeit eines Web Services unvollständige, ungültige oder widersprüchliche Eingaben zu verarbeiten und daraus ein korrektes Ergebnis zu produzieren. (Kalepu et al. 2003, S. 133)
Skalierbarkeit	Die Skalierbarkeit bedeutet, dass ein System abhängig von der Anzahl der Nutzeranfragen oder Operationen in einer gegebenen Zeitspanne die Verarbeitungskapazität erhöhen und so mehr Nutzeranfragen bedienen kann. Die Skalierbarkeit hängt mit der Performanz zusammen. (Ran 2003, S. 7; W3C Working Group 2003)

Table 2-1 Erläuterung nicht-funktionaler Zielgrößen 1

Nicht-funktionale Zielgröße	Beschreibung
Zuverlässigkeit/ Erfolgreiche Ausführungsrate	Die Zuverlässigkeit eines Web Services beschreibt, dass die Funktionalität, welche in der Beschreibung des Web Services genannt ist, innerhalb einer gegebenen Zeitspanne erfolgreich erbracht wird. Sie wird als Prozentsatz erfolgreicher Ausführungen dargestellt, sprich erfolgreiche Ausführungsrate. (Zeng et al. 2003, S. 414; Zeng et al. 2004, S. 315)
Fehlerrate	Die Fehlerrate gibt die Anzahl der Fehler innerhalb einer bestimmten Zeitspanne wieder. Weiter kann die Fehlerrate in unterschiedliche Fehlerarten wie bspw. Übertragungsfehler oder fehlerhaftes Ergebnis unterteilt werden. (Ran 2003, S. 8; Berbner et al. 2005, S. 270)
Verfügbarkeit/ Ausfallwahrscheinlichkeit	Die Verfügbarkeit zeigt die prozentuale Dauer an, mit der ein Web Service verfügbar und einsatzbereit ist. Eng damit verbunden sind die Zuverlässigkeit und die time-to-repair (TTP). Die TTP ist die benötigte Zeit, um einen Web Service nach dessen Ausfall wieder verfügbar zu machen. (Kalepu et al. 2003, S. 132; Mani und Nagarajan 2002) Die Ausfallwahrscheinlichkeit ist ein prozentueller Wert, der angibt, wie hoch die Wahrscheinlichkeit ist, dass der Web Service nicht verfügbar ist oder ausfällt. Sie ergibt sich aus der Gegenwahrscheinlichkeit der Verfügbarkeit. (Heinrich et al. 2011b, S. 96)
Kosten	Die Kosten sind das monetäre Entgelt, das ein Servicenehmer bezahlen muss, um den Web Service zu nutzen. Abgerechnet werden kann die Inanspruchnahme eines Web Services generell nach der Nutzungsdauer, der Anzahl der Aufrufe, nach der Größe des verarbeiteten Datenvolumens oder nach einem Pauschaltarif. (Berbner et al. 2005, S. 270-271; Zeng et al. 2004, S. 315-316)

Tabelle 2-2 Erläuterung nicht-funktionaler Zielgrößen 2

Nicht-funktionale Zielgröße	Beschreibung
Reputation	<p>Die Reputation wird vom Nutzer des Web Services bestimmt und ist das Maß der Vertrauenswürdigkeit des Services oder Anbieters. Zustande kommt die Reputation durch die Erfahrungen der Nutzer mit dem Web Service in der Vergangenheit. Gemessen und dargestellt wird sie, wie bspw. bei Amazon.com, durch eine Skala von [0, 5] in verschiedenen Bereichen. Mittels der Reputation können auch Referenzen bspw. über namhafte Geschäftspartner, welche ebenfalls den Web Service oder Anbieter nutzen, mit in Betracht gezogen werden. (Zeng et al. 2004, S. 315; Berbner et al. 2005, S. 270)</p>
Sicherheit	<p>Da Web Services über das Internet aufrufbar sind, ist die Sicherheit eine besonders wichtige nicht-funktionale Zielgröße. Ein dem Bedürfnissen des Nutzers entsprechendes Sicherheitslevel muss vom Web Service Anbieter sichergestellt sein. Hierzu müssen vom Anbieter ausreichende Sicherheitsmechanismen bereitgestellt werden. Diese beinhalten Authentifizierung, Autorisierung, Vertraulichkeit, Haftung, Rückverfolgbarkeit/ Nachvollziehbarkeit, Datenverschlüsselung und Nachweisbarkeit.</p> <p>Authentifizierung bedeutet, dass der Nutzer des Web Services identifiziert werden kann. Durch Autorisierung können für die Nutzer unterschiedliche Zugriffsberechtigungen festgelegt werden. Die Vertraulichkeit bezieht sich auf die Art und Weise, wie die Daten übertragen werden. Mit der Haftung wird sichergestellt, dass der Serviceanbieter für die Funktionalität haftbar gemacht werden kann. Dafür ist jedoch die Rückverfolgbarkeit und Nachvollziehbarkeit notwendig, mit der Web Service Aufrufe und die Ergebnisse überprüft werden können. Eine Datenverschlüsselung ist essentiell, um nicht Unberechtigten den Zugang zu ermöglichen und damit die Daten nur vom Empfänger entschlüsselt werden können. Die Nachweisbarkeit garantiert, dass Nutzer den Aufruf und die Nutzung eines Web Services nicht abstreiten können. (Ran 2003, S. 9; W3C Working Group 2003; Berbner2007, S. 39)</p>

Tabelle 2-3 Erläuterung nicht-funktionaler Zielgrößen 3

Für einen Web Service Nutzer können je nach Funktionalität des Web Services auch ganz andere nicht-funktionale Zielgrößen von Bedeutung sein. Dafür gibt es die sogenannten domainspezifischen nicht-funktionalen Zielgrößen. So sind etwa bei einem Temperaturservice

spezielle nicht-funktionale Zielgrößen wie Präzision der gemessenen Temperatur oder die Aktualisierungsfrequenz der Temperaturanzeige sehr viel wichtiger als etwa die Sicherheit des Web Service. (Canfora et. al. 2005, S. 1069) Domainspezifische nicht-funktionale Zielgrößen können jedoch nur in die Web Service Selektion mit einbezogen werden, wenn sie quantifizierbar und in reellen Zahlen darstellbar sind. (Alrifai et al. 2010, S. 13)

Wie beim Punkt „Kosten“ in Tabelle 2-2 bereits erläutert, lassen sich unterschiedliche Arten von Abrechnungen zur Web Service Nutzung definieren. Auf dem Web Service Markt wird aber auch eine große Anzahl an kostenfreien Web Services angeboten. Kostenfreie Web Services beziehen sich in diesem Zusammenhang nur auf die Nutzung der Web Services, nicht jedoch auf das eigentliche Produkt bzw. die Kerndienstleistung. (Nüttgens und Dirik 2008, S. 32) Wird ein Web Service aber kostenpflichtig angeboten, lassen sich aus Sicht des Web Service Anbieters unterschiedliche Erlösformen oder eben aus Sicht des Web Service Nutzers verschiedene Formen an Kosten definieren. In Nüttgens und Dirik (2008) werden dazu verschiedene Erlösmodelle und Geschäftsmodelle für Web Service Anbieter vorgestellt. Besteht in einem Unternehmen die Möglichkeit Web Services selbst zu entwickeln, muss ein Unternehmen, bevor es Web Services in ihre Geschäftsprozesse mit einbezieht, die grundlegende Entscheidung treffen, ob die Web Services selbst entwickelt oder vom Markt bezogen werden. Entscheidet sich das Unternehmen im Zuge dieser „Make-Or-Buy“ Entscheidung für den Kauf, ist zwischen den folgenden Fällen zu unterscheiden: Wird ein Web Service mit der gewünschten Funktionalität bereits am Markt angeboten, kann das Unternehmen diesen Web Service direkt erwerben. Im anderen Falle kann ein IT-Dienstleister wie StrikeIron mit der Entwicklung des gewünschten Web Service beauftragt werden. In beiden Fällen sind die Kosten durch sichere und im Vorhinein bekannte Auszahlungen in diskontierter Form darstellbar. (Heinrich et al. 2011a) Ist auf dem Markt kein geeigneter Web Service verfügbar, kann dieser auch selbst entwickelt werden. Für die interne Entwicklung eines Web Services benötigt es Entwicklungsarbeit, die zwangsweise zu Kosten führt. (Heinrich et al. 2011a) Des Weiteren ist die Softwareentwicklung mit Risiken behaftet, siehe dazu Ko et al. (2011, S. 29-30). Die Kosten der Web Service Entwicklung inklusive der Risiken können durch Kostenmodelle abgeschätzt werden, wie bspw. dem von Bary W. Boehm entwickelten COCOMO Modell. (Boehm et al. 2000) Ein positiver Effekt der Eigenentwicklung von Web Services kann durch die Möglichkeit entstehen, die Funktion des Web Services nach der Entwicklung vermarktbar zu machen und diese zusätzlich auch auf dem Web Service Markt anzubieten. (Heinrich et al. 2011a) Diese Form der Vermarktung gewinnt laut Nüttgens und Dirik (2008, S. 32) und Legner (2009, S. 38-40) zunehmend an Bedeutung.

3 Selektion von Web Services

Dieses Kapitel bildet den Hauptteil dieser Arbeit, nämlich die Selektion von Web Services anhand der im Kapitel 2.3 beschriebenen nicht-funktionalen Zielgrößen. In diesem Kapitel werden verschiedene Verfahren zur Selektion von Web Services dargestellt und beschrieben, miteinander verglichen und bewertet. Begonnen wird in Kapitel 3.1 zunächst mit der Erläuterung der Anforderungen, Herausforderungen und Erfolgsfaktoren, die für eine qualitative Selektion von Web Services notwendig sind. Hierbei wird auch auf den Prozess, ausgehend von der Problemstellung bis hin zur optimalen Servicekette, eingegangen. In Kapitel 3.2 werden verschiedene Verfahren dargestellt, mit deren Hilfe eine Selektion durchgeführt werden kann. Außerdem werden die Selektionsverfahren in diesen Kapitel verglichen und bewertet. In Kapitel 3.3 wird auf das Re-planning eingegangen. In Kapitel 3.4 wird abschließend ein konkretes Beispiel zur Erstellung einer Web Service Komposition dargestellt.

3.1 Anforderungen zur Selektion von Web Services basierend auf nicht-funktionalen Zielgrößen

Das Ziel der Selektion anhand nicht-funktionaler Zielgrößen ist die Erstellung einer ökonomisch optimalen Komposition an Web Services. Diese optimale Komposition wird im Weiteren als optimale Servicekette bezeichnet. Ein Verfahren zur Findung der optimalen Servicekette muss sich an den Nutzerpräferenzen des Web Service Nutzers orientieren. Daher ist eine optimale Servicekette in erster Linie eine den Nutzerpräferenzen angepasste Servicekette. Da bspw. niedrige Kosten oder Ausführungszeiten stets im Interesse des Nutzers sind, entspricht eine optimale Servicekette auch einer ökonomisch optimalen Servicekette. In Kapitel 3.1.1 wird ein Konzept beschreiben, mit welchen es möglich ist, eine optimale Servicekette in fünf Schritten zu selektieren. Außerdem wird in Kapitel 3.1.2 auf die Ausgangssituation der Web Service Selektion und in Kapitel 3.1.3 auf die Aggregation nicht-funktionaler Zielgrößen eingegangen. Des Weiteren werden in Kapitel 3.1.4 die Herausforderungen, welchen sich moderne Selektionsverfahren stellen müssen, nähergebracht.

3.1.1 Erstellung der optimalen Servicekette

Bevor eine optimale Servicekette nach nicht-funktionalen Zielgrößen erstellt werden kann, muss zunächst eine semantikbasierte Ermittlung der Abhängigkeiten zwischen den Aktionen gemacht werden. Dabei muss der Planer mit Hilfe eines Interferenzmechanismus und auf Basis der Problemstellung alle notwendigen Aktionen und deren Abhängigkeiten ermitteln. Die Grundlage dazu ist eine Menge an Aktionen, die aufgrund semantisch beschriebener Vor- und Nachbedingungen voneinander abhängig sind. Alle Aktionen sind in einer Prozessbibliothek gespeichert. Das Ergebnis dieses ersten Schritts ist eine Abfolge von Aktionen, welche durch einen Aktionsabhängigkeitsgraph (AAG) dargestellt werden. Ein Aktionsabhängigkeitsgraph beinhaltet die einzelnen Aktionen, deren Vor- und Nachbedingungen und deren Abhängigkei-

ten zueinander. Auf Basis des Aktionsabhängigkeitsgraphen werden nun im zweiten Schritt die zulässigen Ablaufreihenfolgen der Aktionen ermittelt. Resultat dieses Schritts ist ein Aktionszustandsgraph (AZG), der alle zulässigen Ablaufreihenfolgen hinsichtlich der problemorientierten Zustände repräsentiert. Im weiteren Schritt ist es nun möglich syntaktisch korrekte Prozessmodelle zu konstruieren und diese in einer Prozessbibliothek zu speichern. Diese syntaktisch korrekten Prozessmodelle stellen zulässige Abfolgen von Aktionen dar und werden als Aktionsketten bezeichnet. (Heinrich et al. 2011b, S. 89-90)

Ab hier ist es jetzt möglich funktional passende Web Services den einzelnen Aktionen zuzuordnen. Dabei werden durch den Planer zulässige Prozessmodelle abgerufen und die semantischen Beschreibungen der Aktionen mit den semantischen Beschreibungen der verfügbaren Web Services verglichen. Die Zuordnung der Web Services zu den Aktionen muss aufgrund der stetigen Veränderung des Web Service Angebots regelmäßig oder wenn notwendig bei jeder Prozessausführung aktualisiert werden. Das Resultat dieses Schritts ist ein Servicezuordnungsgraph (SZG). Der SZG zeigt einen Prozess, bei dem jeder Aktion eine bestimmte Menge an funktional gleichen (substituierbaren) Web Services zugeordnet sind. (Heinrich et al. 2011b, S. 89-90)

Die folgende Abbildung 3-1 (Heinrich et al. 2011b, S. 90) zeigt den Prozess von der Ermittlung der notwendigen Aktionen und Abhängigkeiten hin zur Erstellung der optimalen Servicekette, wie er in diesem Kapitel 3.1.1 beschrieben wird.

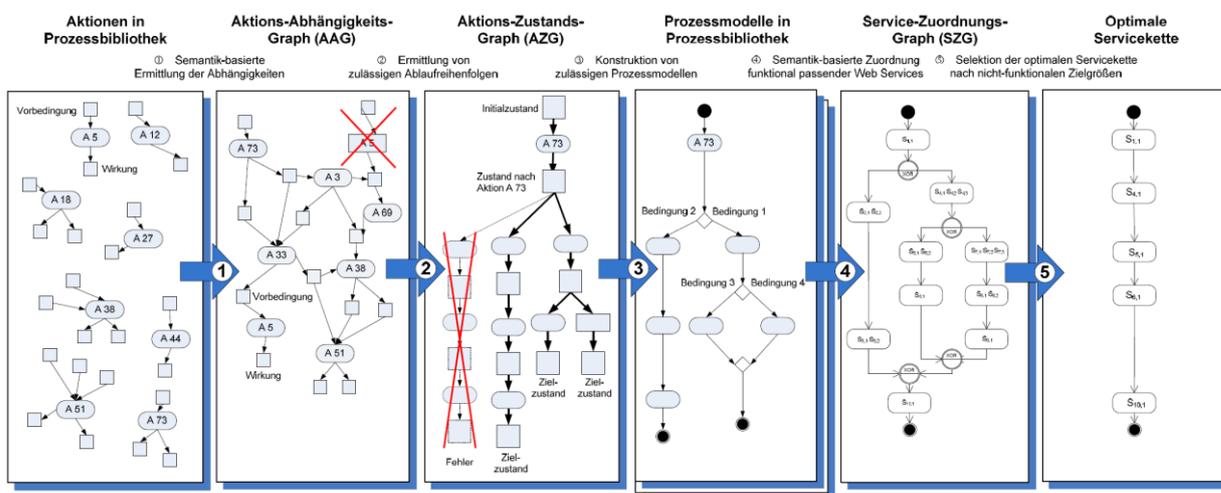


Bild 3-1 Die fünf Schritte des Konzepts (Heinrich et al. 2011b, S. 90)

In Anknüpfung an den Schritt der Zuordnung passender Web Services zu einer Aktion des Prozessmodells wird nun, ausgehend vom Servicezuordnungsgraph, die optimale Servicekette auf Basis nicht-funktionaler Zielgrößen selektiert. Dabei werden alle alternativen Serviceketten anhand nicht-funktionaler Zielgrößen wie bspw. Ausführungszeit, Ausführungskosten oder Ausfallwahrscheinlichkeit ökonomisch bewertet. Die optimale Servicekette beinhaltet diejenigen Web Services, welche den Präferenzen des Web Service Nutzers am Besten entsprechen, also eine ökonomisch vorteilhafte Servicekette bilden und für ihn den größten Nut-

zen bringen. Diese Web Services werden anschließend aufgerufen und anhand des Ablaufs der optimalen Servicekette ausgeführt. Für die Dokumentation und Aktualisierung der nicht-funktionalen Zielgrößen wird das Verhalten dieser für die spätere Prozessanalyse protokolliert, so wirkt sich bspw. ein erfolgreicher Durchlauf positiv auf die Ausfallwahrscheinlichkeit aus (Heinrich et al. 2011b, S. 90-93) Ein konkretes Beispielfahrer sowie explizite Verfahren zur Selektion von Web Services, d. h. der Übergang vom Servicezuordnungsgraph zur optimalen Servicekette, welcher in Abbildung 3-1 im Schritt 5 dargestellt wird, werden in Kapitel 3.2 vorgestellt und näher beschrieben.

3.1.2 Ausgangssituation der Selektion

Für jede Aktion des Prozessmodells wird in der Praxis eine Reihe an funktional identischen Web Services gefunden. Jedoch weisen diese Web Services unterschiedliche nicht-funktionale Zielgrößen auf und sind somit in ihrer Beschaffenheit und Qualität unterscheidbar, bspw. durch unterschiedliche Ausführungskosten oder Reputation. (Borbner et al. 2005, S. 269)

Des Weiteren können sich die nicht-funktionalen Zielgrößen der Web Services jederzeit ändern, so kann bspw. im Zeitverlauf ein Web Service Anbieter für einen Web Service die Ausführungskosten senken oder etwa durch eine hohe Serverauslastung ein Web Service eine höhere Ausführungszeit benötigen. Aus diesem Grund muss die optimale Servicekette regelmäßig oder unter Umständen vor jeder Ausführung selektiert werden. (Heinrich et al. 2011b, S. 90)

Um eine Servicekette bewerten zu können, müssen die einzelnen nicht-funktionalen Zielgrößen der Web Services im Prozessmodell aggregiert werden (siehe Kapitel 3.1.3). So müssen bspw. die Kosten oder die Ausführungszeit der Servicekette auf Basis der enthaltenen Web Services berechnet werden. In der Praxis existiert in der Regel jedoch keine Aktionskette, welche eine dominante Servicekette beinhaltet, in der alle nicht-funktionalen Zielgrößen besser als oder zumindest gleich gut wie alle anderen Servicekettensind. Deshalb muss eine Präferenzfunktion erstellt werden, mit dieser die unterschiedlichen nicht-funktionalen Zielgrößen bewertet und anschließend zu einem Gesamtpräferenzfunktionswert aggregiert werden. Mithilfe dieses Wertes ist für jede Aktionskette des SZG die lokal optimale Servicekette selektierbar. Im Anschluss an die Selektion der lokal optimalen Servicekette muss die global optimale Servicekette bestimmt werden. Die global optimale Servicekette wird durch den maximalen Gesamtpräferenzfunktionswert bestimmt und im Anschluss ausgeführt. (Heinrich et al. 2011b, S. 93)

Neben der Veränderung der nicht-funktionalen Zielgrößen der Web Services können in der Praxis Web Services oder Serviceanbieter auch komplett ausfallen oder zeitweise nicht zur Verfügung stehen. Im Falle, dass schon vor der Ausführung der global optimalen Servicekette bekannt ist, dass ein Web Service nicht zur Verfügung steht, muss der Web Service aus dem Dienstverzeichnis temporär gestrichen werden. Der Web Service ist damit nicht mehr Teil des SZG und alle Serviceketten, welchen diesen Web Service beinhalten, werden folglich nicht

mehr in Betracht gezogen. Die lokal optimale Servicekette der Aktionskette ist nun die mit dem neuen höchsten Gesamtpräferenzfunktionswert bzw. die nächstbeste der ausführbaren Serviceketten. Im Extremfall, wenn für einen Web Service kein funktionales Substitut existiert, kann eine Aktion überhaupt nicht mehr ausgeführt werden. In diesem Fall kann auch die gesamte Aktionskette nicht mehr ausgeführt werden und muss temporär aus dem SZG entfernt werden. (Heinrich et al. 2011b, S. 93)

In weiterer Folge ist es auch notwendig, den Fall zu betrachten, wenn ein Web Service oder Serviceanbieter während der Ausführung der optimalen Servicekette ausfällt. In solch einem Fall ist es nicht möglich den ausgefallenen Web Service einfach durch den nächstbesten zu ersetzen. Hier muss eine weiterführende Betrachtung der bereits gestarteten Servicekette stattfinden, da bspw. die bereits gestartete Servicekette unter Verwendung alternativer Web Services unter Umständen nicht mehr optimal ist. Des Weiteren müssen auch Laufzeitveränderungen der Web Services berücksichtigt werden, bspw. ein Web Service hat aufgrund einer hohen Serverauslastung eine längere Ausführungszeit als vorhergesagt oder ein alternativer Web Service bietet unter gleichen Ausführungskosten eine geringere Ausführungszeit. Diese Fälle können durch Re-planning gelöst werden und werden in Kapitel 3.3 nochmals detaillierter beschrieben. (Heinrich et al. 2011b, S. 93; Berbner et al. 2005, S. 110)

3.1.3 Aggregation der nicht-funktionalen Zielgrößen

Anzuwenden sind die in Kapitel 2.3 genannten nicht-funktionalen Zielgrößen nicht nur zur Bewertung eines einzelnen Web Services, sondern auch zur Bewertung der Qualität einer Web Service Komposition, also deren nicht-funktionalen Zielgrößen. Dabei werden die einzelnen nicht-funktionalen Zielgrößen abhängig von ihrer Art aggregiert. D. h. sie müssen dabei entweder wie bspw. die Kosten oder die Ausführungszeit summiert werden (Additive Parameter) oder wie etwa die Zuverlässigkeit oder die Verfügbarkeit multipliziert werden (Multiplikative Parameter). Um bspw. die Gesamtzuverlässigkeit einer Web Service Komposition zu messen, werden die einzelnen prozentuellen Zuverlässigkeitswerte miteinander multipliziert und somit zu einem Wert aggregiert, der die Gesamtzuverlässigkeit darstellt. In anderen Worten wird damit die durchschnittliche Gesamtzuverlässigkeit errechnet. Der Gesamtdurchsatz einer Web Service Komposition kann nicht durch Multiplikation oder Addition berechnet werden. Hier muss ein Minimaloperator-Parameter verwendet werden. Der geringste Wert des Durchsatzes aller verwendeten Web Services stellt den maximalen Durchsatz dar. (Berbner 2007, S. 71; Zeng et al. 2004, S. 316)

Anhand eines Kontrollflusses ist ersichtlich, wie die einzelnen Aktionen zusammengefügt sind. Der Kontrollfluss ist somit ein Plan, wie die nicht-funktionalen Zielgrößen der einzelnen Web Services aggregiert werden, um anschließend die optimale Servicekette zu selektieren. Der Kontrollfluss beschreibt die Aktivitäten und die Ausführungsreihenfolge der Web Services, bspw. anhand eines UML-Aktivitätsdiagramm. Er ist zu vergleichen mit einer Aktionskette. Aus dem Kontrollfluss ist ersichtlich, dass bspw. verschiedene Web Services parallel ausgeführt werden müssen oder eine Auswahl zwischen zwei alternativen Pfaden gemacht werden muss. (Heinrich et al. 2011b, S. 89; Strunk 2010, S. 67-68; Berbner 2007, S. 28-31)

Zur Veranschaulichung zeigt das folgende UML-Aktivitätsdiagramm (Abbildung 3-2) einen vereinfachten Geschäftsprozess einer Reisebuchung (Berbner 2007, S. 31), wie er in einem Reisebüro stattfinden kann. Daraus ist ersichtlich, dass im Anschluss an die Erfassung der Reisedaten die Flugsuche, Hotelsuche und die Mietwagensuche parallel abläuft. Nach der Auswahl eines geeigneten Reisepaketes kann das Angebot entweder angenommen und somit gebucht werden oder das Angebot wird verworfen bzw. abgebrochen. (Berbner 2007, S. 31-32)

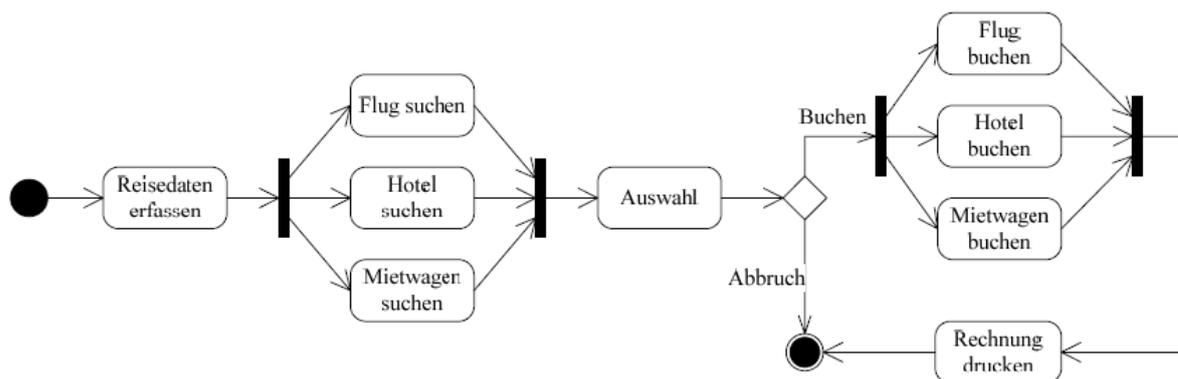


Bild 3-2 UML-Aktivitätsdiagramm Reisebuchung (Berbner 2007, S. 31)

In welcher Art und Weise einzelne Web Services miteinander in Verbindung stehen können, zeigt Abbildung 3-3. Web Services können in sequentieller Abfolge in Verbindung stehen, in einer UND Verbindung, wo alle nachfolgenden Web Services gleichzeitig ausgeführt werden und in einer XOR Verbindung wo anhand einer bestimmten Wahrscheinlichkeit p ein nachfolgender Web Service ausgeführt wird. In einer Schleife wird derselbe Web Service höchstens K -mal hintereinander ausgeführt. (Yu et al. 2007, S. 5)

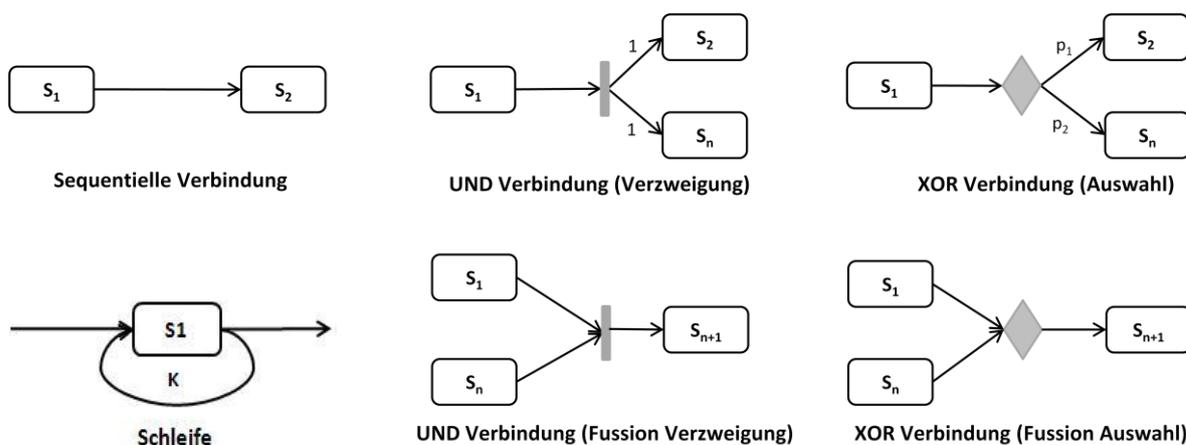


Bild 3-3 Verbindungsstrukturen in Anlehnung an Yu et al. (2007, S. 5)

In der Tabelle 3-1 sind die Formeln aufgeführt, welche zur Aggregation der einzelnen nicht-funktionalen Zielgrößen verwendet werden. Zu den nicht-funktionalen Zielgrößen Verfügbarkeit, Zuverlässigkeit (beides Multiplikative Parameter), Ausführungszeit und Ausführungskosten (beides Additive Parameter) werden die Aggregationsformeln für die Konstrukte Sequenz, Verzweigung, Schleife und Auswahl (XOR) vorgestellt. (Strunk 2010, S. 67)

Nicht-funktionale Zielgröße	Sequenz	Verzweigung	Schleife	Auswahl
Verfügbarkeit	$\prod_{i=1}^m A(s_i)$	$\prod_{i=1}^p A(s_i)$	$A(s)^k$	$\prod_{i=1}^n p_i * A(s_i)$
Zuverlässigkeit	$\prod_{i=1}^m R(s_i)$	$\prod_{i=1}^p R(s_i)$	$R(s)^k$	$\prod_{i=1}^n p_i * R(s_i)$
Ausführungszeit	$\sum_{i=1}^m T(s_i)$	$Max(T(s_i)_{i \in \{1...p\}})$	$k * T(s)$	$\sum_{i=1}^n p_i * T(s_i)$
Ausführungskosten	$\sum_{i=1}^m P(s_i)$	$\sum_{i=1}^p P(s_i)$	$k * P(s)$	$\sum_{i=1}^n p_i * P(s_i)$

Tabelle 3-1 Aggregationsformeln zur Berechnung des Gesamtpräferenzfunktionswerts in Anlehnung an Strunk (2010, S. 68)

3.1.4 Herausforderungen der Selektion - Performance der Selektionsverfahren

Damit eine Web Service Selektion den Wünschen und Präferenzen des Nutzers entspricht, dürfen nicht nur ausschließlich die in Kapitel 2.3 genannten nicht-funktionalen Zielgrößen betrachtet werden. Beim Prozess der Selektion müssen einige weitere Kriterien und Leistungen berücksichtigt werden bzw. müssen vom Selektionsverfahren erbracht werden.

3.1.4.1 Selektionszeit/ Echtzeitselektion

Canfora et al. (2005a, S. 1070) etwa stellen fest, dass Web Service Kompositionen möglichst schnell erstellt werden müssen. Insbesondere wenn interaktive Systeme betroffen sind, stellen große Verzögerungen ein unakzeptables Problem dar. Als Beispiel führen die Autoren die Suche nach dem günstigsten Flugticket in einem Flugticketportal an. Wenn ein Nutzer hierbei minutenlange Wartezeiten in Kauf nehmen muss während das System nach Web Service Kandidaten sucht die Flugtickets anbieten, nur um ein paar Cent beim Buchungspreis zu spa-

ren, wird ihn das enttäuschen. (Canfora et al. 2005a, S. 1070) Eine noch viel größere Bedeutung kann eine Echtzeitselektion in einem medizinischen Informationssystem haben. Z. B. wenn etwa in einer Notaufnahme ein Patientendossier aus einem anderen Krankenhaus angefordert werden muss. (Yu et al. 2007, S. 2) Eine schnelle Erstellung einer Web Service Komposition ist außerdem bezüglich des Re-plannings wichtig. Für den Fall, dass sich die vorhergesagten nicht-funktionalen Zielgrößen eines Web Services von den tatsächlich eintreffenden unterscheiden, oder im Falle des Ausfalls eines Web Services während der Ausführung der optimalen Servicekette muss eine schnelle Komposition einer alternativen Servicekette erfolgen. Eine langandauernde Erstellung einer alternativen Servicekette wirkt sich in auf die Gesamtausführungszeit einer Web Servicekomposition aus und muss, um Verletzungen der SLA zu vermeiden, daher möglichst gering gehalten werden. (Canfora et al. 2005a, S. 1070) Aufgrund der Bedeutung der Selektionszeit beziehen bspw. Yu und Lin (2005a) die Selektionszeit als eines der Qualitätskriterien in ihre Evaluierungen der Selektionsverfahren bzw. der Algorithmen mit ein.

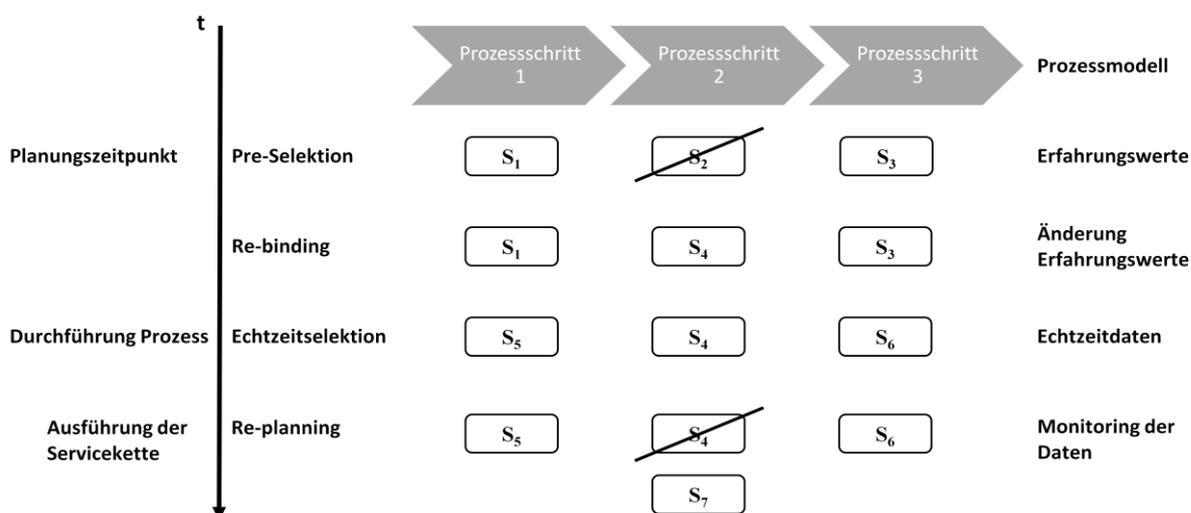


Bild 3-4 Zeitpunkte der Selektion

Abbildung 3-4 verdeutlicht die zeitlichen Stationen der Web Service Selektion. Zum Planungszeitpunkt der Servicekette wird lediglich eine Pre-Selektion durchgeführt. Diese kann zeitlich weit vor der eigentlichen Ausführung der Servicekette liegen. Hierbei wird durch ein Selektionsverfahren für jeden Prozessschritt ein Web Service anhand der dokumentierten nicht-funktionalen Zielgrößen (Erfahrungswerte) zugeordnet. Da diese Servicekette nicht zum Zeitpunkt der Durchführung selektiert worden ist, können sich bis zur Ausführung der Servicekette die nicht-funktionalen Zielgrößen ändern (Änderung der Erfahrungswerte). Sobald sich eine nicht-funktionale Zielgröße eines bereits selektierten Web Services ändert und somit bspw. eine Verletzung der SLA mit sich bringen würde, wird ein Re-binding durchgeführt. In Abbildung 3-4 wird der betroffene Web Services S₂ somit gestrichelt und ein alternativer Web Service S₄ wird für den Prozessschritt selektiert. (Canfora et al. 2008, S. 1757) Eine neue Zuordnung der Web Services kann sich während der Durchführung des Prozesses ergeben. Hier

wird eine Echtzeitselektion durchgeführt, d. h. die selektierten Web Services der Servicekette können komplett andere sein als nach der Pre-Selektion. Eine Echtzeitselektion ist mittels heuristischen Verfahren möglich und wird bspw. in den Selektionsverfahren von Yu et al. (2007) und Wada et al. (2008) angesprochen. Aufgrund der NP-Schwere des Optimierungsproblems benötigen exakte Verfahren hohe Rechenzeiten für die Web Service Selektion. Exakte Verfahren sind daher nicht in der Lage eine Echtzeitselektion durchzuführen. (Berbner 2007, S. 76) Bei einer Änderung der nicht-funktionalen Zielgrößen während der Ausführung der Servicekette kommt ein Re-planning Verfahren zum Einsatz. Ein Re-planning wird durchgeführt, wenn bspw. ein Web Service während der Ausführung ausfällt oder eine alternative Servicekette durch geringere Ausführungskosten eine Verbesserung der nicht-funktionalen Zielgrößen erzielen würde. In beiden Fällen können sich die nicht-funktionalen Zielgrößen dahingehend ändern, dass eine Neubewertung der Servicekette oder ein Einsatz einer alternativen Servicekette notwendig ist. Das Ziel ist es, eine dynamische Anpassung der Servicekette, welche auch während der Ausführung der Servicekette greift, zu erreichen. (Heinrich et al. 2011b, S. 89-93) In Abbildung 3-4 fällt S_4 aus und wird im Re-planning Verfahren durch den funktional gleichen Web Service S_7 ersetzt. Eine Echtzeitselektion verringert jedoch die Wahrscheinlichkeit ein Re-planning durchführen zu müssen.

3.1.4.2 Web Service Abhängigkeiten und Konflikte

Eine weitere Herausforderung der Web Service Selektion ist die, die Abhängigkeit von Web Services und deren nicht-funktionalen Zielgrößen zu anderen Web Services und deren nicht-funktionalen Zielgrößen zu beachten. Die klassische Web Service Selektion nimmt an, dass zwischen den einzelnen Web Services keine Abhängigkeiten und Konflikte bestehen. Realistischerweise bestehen jedoch oftmals zwischen einzelnen Web Services in einer Servicekette Beziehungen wie bspw. technischer Art oder durch eine Partnerschaft mit dem Web Service Anbieter. (Strunk 2010, S. 72)

Da Web Services in der Regel fremdbezogen werden, sind diese ein Teilgebiet des Business Activity Outsourcing. Unternehmen sind deshalb durch die zunehmende Verwendung von Web Services von der Leistungsfähigkeit der externen IT-Dienstleister (Web Service Anbieter) immer mehr abhängig. Da in einer Web Servicekomposition bspw. bei einem Kreditvergabeprozess (siehe Kapitel 3.2.1) aus ökonomischen Gründen mehrere Web Services eines einzigen Anbieters integriert sein können, stehen diese Web Services in einer Abhängigkeit bzw. können nicht als voneinander komplett unabhängig bezeichnet werden. So kann bspw. ein technischer Ausfall des Serversystems des Web Service Anbieters das Web Service Angebot des IT-Dienstleisters und in Folge dessen die in der Servicekette verwendeten Web Services nicht verfügbar machen. (Braunwarth und Heinrich 2008, S. 98-99 + 107) Die Herausforderung ist daher die, Risiken durch Abhängigkeiten vom Web Service Anbieter im IT-Service-Management eines Unternehmens zu berücksichtigen und steuerbar zu machen. (Braunwarth und Heinrich 2008, S. 98)

Grundlegend sind im IT-Service-Management Abhängigkeiten in zwei Typen einteilbar. Zum einen ist das die Abhängigkeit von der technischen Leistungsfähigkeit des IT-Lieferanten und

zum anderen besteht eine Abhängigkeit von der Innovationskraft des IT-Lieferanten. Die erste Gruppe von Abhängigkeiten betrifft die Qualität der zu erbringenden Leistung. So kann aufgrund von Serverproblemen beim IT-Lieferanten ein Web Service ausfallen oder durch mangelnde Organisation Web Services nicht termingerecht bereitgestellt werden. Die Anzahl der Gründe hierfür ist sehr groß und kann vom IT-Lieferanten manchmal kaum bis überhaupt nicht beeinflusst werden, bspw. bei einer Naturkatastrophe. Die zweite Gruppe von Abhängigkeiten betrifft die Flexibilität des IT-Lieferanten. Da sich die IT stetig weiterentwickelt (Stichwort Moorsches Gesetz), ist Innovationskraft in der IT-Branche sehr wichtig. IT-Lieferanten müssen in der Lage sein neue Technologien schnell zu adaptieren um diese den Kunden zur Verfügung zu stellen. Eine mangelnde Flexibilität hierbei wirkt sich nicht nur negativ auf den Erfolg des IT-Lieferanten aus, sondern auch auf den des Kunden. (Braunwarth und Heinrich 2008, S. 99)

Der Vorteil von Web Service Kompositionen liegt in ihrer losen Kopplung und der Möglichkeit, Web Services verschiedener Web Service Anbieter darin zu integrieren. Das Risiko von Ausfällen kann somit reduziert werden. Je weniger zwei Web Services miteinander korreliert sind, desto höher sind die Diversifikationsvorteile und desto geringer auch mögliche Risiken oder Konflikte. Die Verteilung eines Prozesses auf mehrere externe IT-Dienstleister bietet bei einer nicht vollständigen Korrelation ein geringeres Risiko als die Summe der einzelnen Risiken der Web Services. (Braunwarth und Heinrich 2008, S. 107)

Neben den Abhängigkeiten zwischen nicht-funktionalen Zielgrößen existieren auch funktionale Abhängigkeiten. Die funktionale Web Service Abhängigkeit in einer Servicekette lässt sich anhand folgenden Beispiels darstellen: Für die Komposition eines Reisebuchungs-Web Service soll ein Web Service für die Reiseversicherung gewählt werden, welcher für die Bezahlung auch Kreditkarten von Master Card akzeptiert. Für die Bezahlung der Reisebuchung muss demzufolge ein Web Service integriert werden, welcher auch Kreditkarten von Master Card akzeptiert. Zudem ist weiter zu beachten, dass, sollte für den Web Service für die Bezahlung der Reisebuchung ein Web Service gewählt werden, der auch Bezahlungen mit Visa Card ermöglicht, der Web Service für die Reiseversicherung ebenfalls Visa Card akzeptieren muss. Würde der Web Service der Reiseversicherung bspw. keine Visa Card akzeptieren, der Web Service zur Bezahlung der Reise jedoch schon, würden bei einer Bezahlung einer Buchung mit Visa Card unweigerlich Konflikte auftreten. (Tang und Ai 2010, S. 268)

3.1.4.3 Zusammenpassen der Schnittstellen (Interface-Matching)

Bei klassischen Web Service Selektionsverfahren wird davon ausgegangen, dass alle Web Services bzgl. ihrer Schnittstelle mit anderen Web Services kooperieren können. Jedoch passen nicht alle funktionell identischen Web Services zu allen anderen Web Services, können also mit diesen operieren (Interface-Matching Problem). Funktional gleiche Web Services können syntaktisch und semantisch unterschiedliche Schnittstellen haben und somit in einer Servicekette nicht integrierbar sein. In einem beschränkten Umfang berücksichtigen einige Verfahren wie Gao et al. (2005) und Wan et al. (2008) in ihren Selektionsverfahren das Interface-Matching Problem. (Strunk 2010, S. 72)

Der Grund für die Heterogenität bzgl. der Schnittstellen ist eine uneinheitliche Entwicklung der Web Services. Diese beginnt schon bei der Namensgebung für die Funktionalität des Web Services. So kann bspw. ein Web Service „Flugticket Buchung“ oder „Flugschein Buchung“ heißen. (Gao et al. 2006, S. 366) Da momentan die Web Service Suche in den Web Service Verzeichnissen und bei Web Service Anbieter stichwortbasiert ist, ist es die Aufgabe der Web Service Selektion, solche Schnittstellenprobleme zu erkennen und zu lösen. (Barros und Dumas 2006, S. 34)

3.2 Verfahren zur Selektion von Web Services

Sind für die jeweiligen Aktionen im Prozessmodell verschiedene Web Services gefunden worden, muss nun die optimale Servicekette mittels eines geeigneten Verfahrens selektiert werden. Es ist davon auszugehen, dass für jede Aktion im Prozessmodell mehrere funktional gleiche Web Services gefunden und zugeordnet werden. Daher muss hinsichtlich der Nutzerpräferenzen der beste Pfad, d. h. die optimale Servicekette gesucht werden. Mögliche Restriktionen können bspw. sein, dass die Servicekette die Gesamtausführungszeit von zwei Minuten nicht überschreiten darf und die Kosten möglichst gering gehalten werden müssen und eine vorgegebene Grenze nicht überschreiten dürfen. (Bebner 2007, S. 69-70; Zhang et al. 2007, S. 1084)

Bevor in Kapitel 3.2.3 verschiedene exakte und heuristische Selektionsverfahren beschrieben werden, wird in diesem Kapitel zunächst anhand eines einfach gehaltenen Beispiels ein konkretes Selektionsverfahren vorgestellt. Anschließend wird auf Vereinfachungen und Annahmen eingegangen, welcher sich einige Selektionsverfahren bedienen, die in der Literatur und hier vorgestellt werden. Im Anschluss an die Vorstellung expliziter Selektionsverfahren werden diese in Kapitel 3.2.4 miteinander verglichen und bewertet.

3.2.1 Verfahrensbeispiel

Wie bereits in 3.1.2 erwähnt, wird die global optimale Servicekette durch einen Gesamtpräferenzfunktionswert bestimmt. Die hierzu relevanten nicht-funktionalen Zielgrößen müssen im Schritt 4 (siehe Abbildung 3-1) für jeden funktional passenden Web Service erfasst werden. Diese können, soweit vorhanden, aus Dienstverzeichnissen für Web Services entnommen werden, über Web Service Ausschreibungen verlangt werden oder beim Einholen von Web Service Angeboten vom Anbieter gefordert werden. (Heinrich et al. 2011b, S. 94)

Eine Kreditwürdigkeitsprüfung ist bei Finanzdienstleistern im Privatkundenbereich ein häufig durchgeführter Prozess. Da versucht wird, solche Prozesse aufgrund der Möglichkeit zur Kostensenkung weitgehend zu standardisieren, werden IT-gestützte Prozesse momentan intensiv diskutiert und spielen in der strategischen Ausrichtung der IT in vielen Unternehmen eine wichtige Rolle. Serviceorientierte Anwendungen und Architekturen bieten die Möglichkeit Prozesse zu verbessern und deren Umsetzung mittels Web Service Kompositionen zu gestalten. Im Folgenden wird anhand eines Beispiels „Kreditwürdigkeitsprüfung“ aus dem Privat-

kundenbereich einer Bankfiliale dargestellt, wie ein solcher Prozess durch den Einsatz von Web Services verbessert werden kann. Ausgehend vom SZG wird hierbei die optimale Servicekette selektiert. (Heinrich et al. 2011b, S 93)

Für den Prozess der Kreditwürdigkeitsprüfung sind zwei alternative Aktionsketten identifiziert worden, die SCHUFA-Auskunft und ein kombiniertes Verfahren (siehe Abbildung 3-5 Prozessmodell Kreditwürdigkeitsprüfung). Das SCHUFA-Auskunft-Verfahren beinhaltet neben der Aktionen 1 „Kreditfähigkeit prüfen“ und 7 „Kreditwürdigkeit beurteilen“, welche in beiden Aktionsketten integriert sind, die Aktion 2 „Autorisierung SCHUFA einholen“ und die Aktion 3 „SCHUFA Auskunft einholen“. Das kombinierte Verfahren beinhaltet neben Aktion 1 und 7, die Aktionen 4 „GeoCoding einholen“, die Aktion 5 „Scoring einholen“ und die Aktion 6 „Bonität prüfen“. (Heinrich et al. 2011b, S. 95)

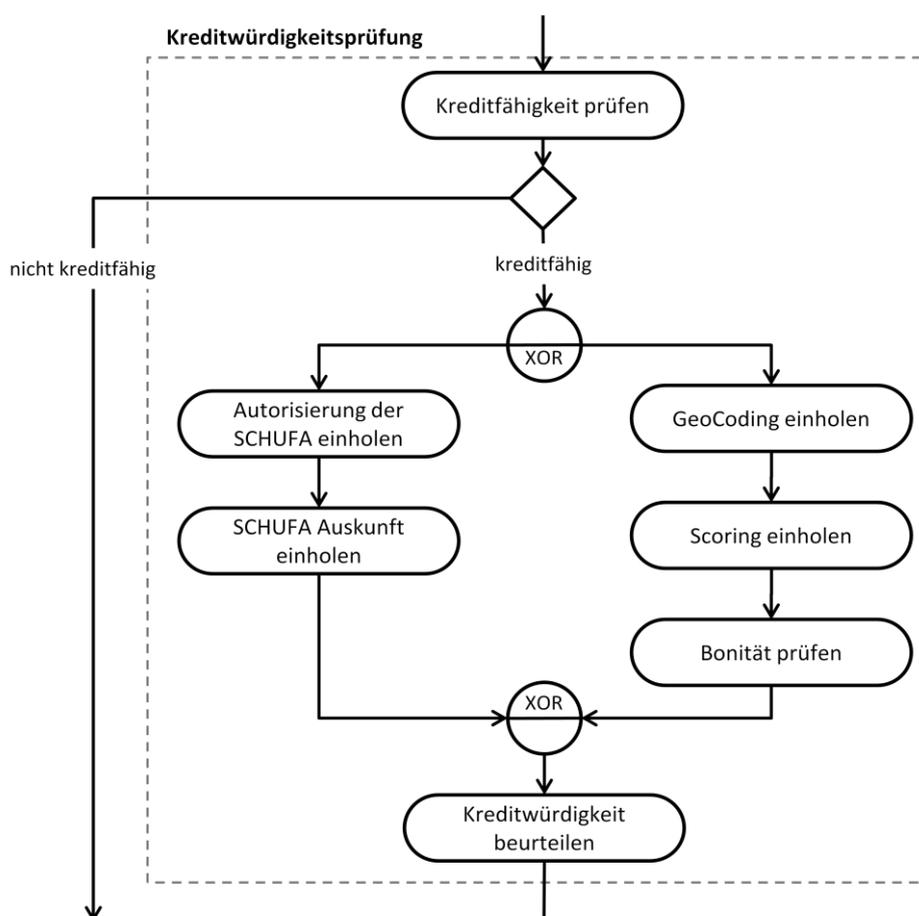


Bild 3-5 Prozessmodell Kreditwürdigkeitsprüfung in Anlehnung an Heinrich et al. (2011b, S. 93-96)

Die Abbildung 3-6 zeigt den SZG zum genannten Beispiel und davon ausgehend eine selektierte Servicekette, welche jedoch weder eine lokal optimale noch die global optimale darstellen muss. Aus dem SZG sind die zwei alternativen Aktionsketten [1-2-3-7] und [1-4-5-6-7] ersichtlich. Die Anzahl der möglichen Serviceketten für die Aktionskette [1-2-3-7] sind 36 (3*2*2*3) und für die Aktionskette [1-4-5-6-7] 54 (3*3*2*1*3). Insgesamt können für diesen

kleinen Prozessausschnitt, welcher nur sieben Aktionen und insgesamt 16 zugewiesene Web Services enthält, 90 Serviceketten gefunden werden. Daraus ist zu schließen, dass bei einem großen Prozessmodell mit mehreren Aktionsketten und funktional passenden Web Services möglicherweise viele Tausend Serviceketten selektierbar und damit zu berücksichtigen sind. (Heinrich et al. 2011b, S. 95)

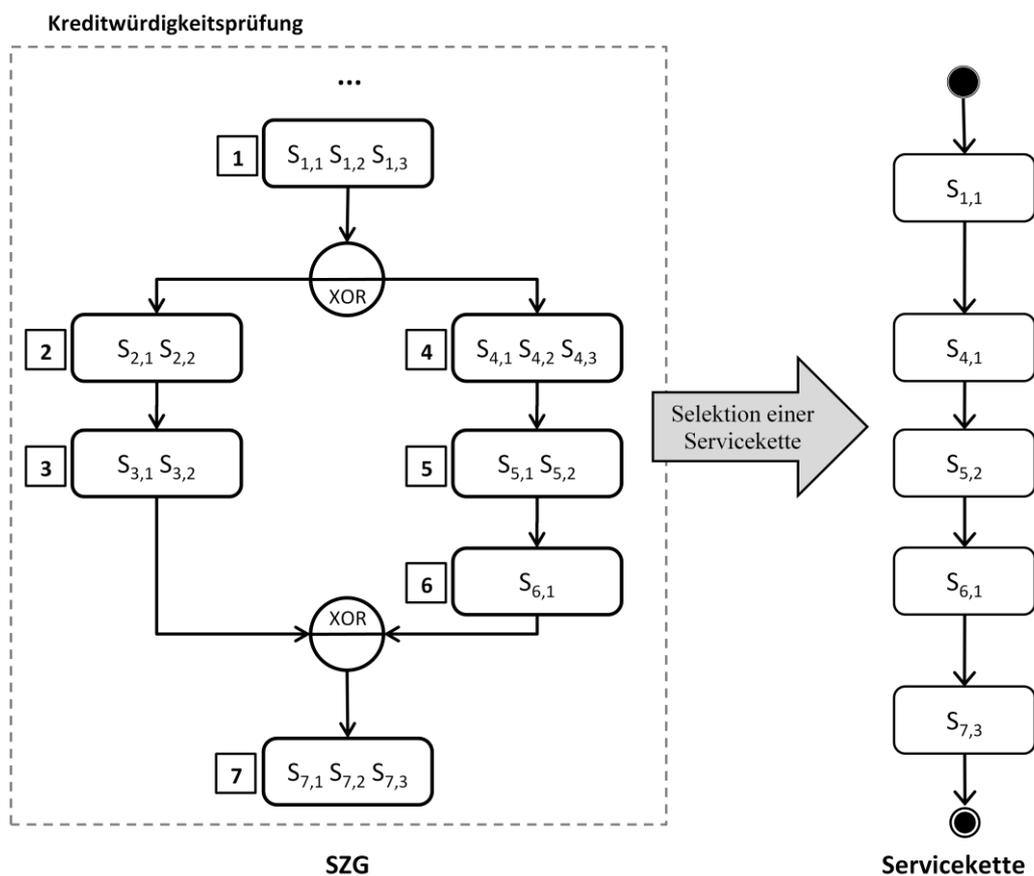


Bild 3-6 Selektion einer Servicekette des Beispiels Kreditwürdigkeitsprüfung, in Anlehnung an Heinrich et al. (2011b, S. 90-95)

Nachdem alle möglichen Serviceketten bestimmt und erfasst wurden muss nun die optimale Servicekette selektiert werden. Dabei werden in diesem Beispiel die nicht-funktionalen Zielgrößen Ausfallwahrscheinlichkeit, Ausführungskosten und durchschnittliche Ausführungszeit herangezogen. Für alle funktional passenden Web Services werden die drei nicht-funktionalen Zielgrößen durch folgenden Bewertungsvektor (Heinrich et al. 2011b, S. 94) erfasst:

$$VectorService (S_{i,j}) = \begin{pmatrix} \text{Ausfallwahrscheinlichkeit} \\ \text{Ausführungskosten} \\ \text{Ausführungszeit} \end{pmatrix} = \begin{pmatrix} p(S_{i,j}) \\ c(S_{i,j}) \\ t(S_{i,j}) \end{pmatrix}$$

Die Selektion wird in diesem Beispiel von einer Software technisch unterstützt. Die nicht-funktionalen Zielgrößen der einzelnen Web Services müssen dabei zunächst aggregiert werden und anschließend für jede Servicekette zu einem Präferenzfunktionswert errechnet werden. Die Ausführungszeit und die Ausführungskosten werden durch Summation bestimmt, diese sind somit adaptive Parameter. Die Ausfallwahrscheinlichkeit ist ein multiplikativer Parameter und wird durch Multiplikation der einzelnen Ausfallwahrscheinlichkeiten bestimmt. Bei allen Web Services wird der Einfachheit halber eine Unabhängigkeit voneinander vorausgesetzt. (Heinrich et al. 2011b, S. 94-96; Berbner 2007, S. 71)

Mit der folgenden Formel (Heinrich et al. 2011b, S. 96) kann der Präferenzfunktionswert Φ errechnet werden. In dieser Formel werden die Aggregationsfunktionen für die Ausfallwahrscheinlichkeit, für die Ausführungskosten und die Ausführungszeit addiert.

$$\Phi = 1,0 * (1 - (\prod_{i=1}^N (1 - p(S_{i,j})))) + 0,6 * \sum_{i=1}^N c(S_{i,j}) + 0,3 * \sum_{i=1}^N t(S_{i,j})$$

Diese Formel repräsentiert den Präferenzfunktionswert für eine Servicekette. In ihr werden auch die Präferenzen des Nutzers bzw. in diesem Beispiel die des Finanzdienstleisters berücksichtigt. Jeder nicht-funktionalen Zielgröße wurde eine bestimmte Gewichtung gegeben. Die Gewichtung der einzelnen nicht-funktionalen Zielgrößen ist abhängig von der Wichtigkeit der nicht-funktionalen Zielgröße für den Nutzer des Web Services. So ist in diesem Beispiel für die Bankfiliale die Ausfallwahrscheinlichkeit das wichtigste Kriterium und wird mit dem höchsten Wert 1,0 gewichtet. Die Kosten wurden mit 0,6 gewichtet, da auf dem Kreditmarkt ein hoher Kostendruck herrscht und Privatkunden eine hohe Preissensitivität aufweisen. Eine Gewichtung von 0,3 wurde der durchschnittlichen Ausführungszeit zugewiesen. Da eine kürzere Ausführungszeit nur wenige Minuten Zeiteinsparung ausmachen würde, ist dieses Kriterium bzgl. des Filialgeschäfts nicht wettbewerbsrelevant und kann einen niedrigen Wert zugewiesen bekommen. Da in diesem Beispiel die enthaltenen Zielgrößen zu minimieren sind, ist der optimale Gesamtpräferenzfunktionswert der Geringste aller errechneten Präferenzfunktionswerte. In diesem Beispiel wurde die global optimale Servicekette in der Aktionskette [1-4-5-6-7] gefunden und hat einen Präferenzfunktionswert von 0,64. Hierzu wurden vom Planer alle lokal optimalen Serviceketten (in diesem Beispiel die jeweils lokal optimale für die Aktionskette SCHUFA-Auskunft und der Aktionskette des kombinierten Verfahren) verglichen und die global optimale mit dem geringeren Präferenzfunktionswert generiert. (Heinrich et al. 2011b, S.96)

Eine solche Selektion ist aufgrund der ständigen Veränderung der Rahmenbedingungen regelmäßig durchzuführen. Unter Umständen ist die Selektion sogar vor jeder Prozessausführung notwendig oder wird durch eine Echtzeitselektion tatsächlich gemacht. Bspw. können sich die Ausführungskosten der bis dahin global optimalen Servicekette (aus dem kombinierten Verfahren) erhöhen, wonach eine Servicekette der alternativen Aktionskette (SCHUFA-Auskunft) nun die global optimale sein kann. Da in der Regel weitaus größere Prozesse mit

einer Vielzahl möglicher Serviceketten vorliegen, ist eine automatisierte Selektion unverzichtbar. In zeitkritischen Fällen muss zudem auf Heuristiken zurückgegriffen werden. (Heinrich et al. 2011b, S. 96) Heuristische Verfahren werden in diesem Kapitel unter Punkt 3.2.3.2 näher erläutert.

3.2.2 Vereinfachungen der Selektion

Die Selektion der optimalen Servicekette ist aufgrund der NP-Schwere des Optimierungsproblems sehr zeitintensiv und arbeitsaufwändig. Aus einer großen Anzahl möglicher Serviceketten die passende zu finden, ist realistischweise ohne Hilfsmittel bzw. Softwareunterstützung nicht machbar. Daher bedienen sich die im Kapitel 3.2.3 vorgestellten Verfahren einer oder mehrerer der folgenden Vereinfachungen. (Strunk 2010, S. 68-71)

I. Lineare Zielfunktion: Eine Zielfunktion ist eine Aggregationsregel, mit der der Präferenzfunktionswert berechnet wird. Mit einer linearen Zielfunktion wird bspw. die Ausführungszeit durch die Summe aller Web Services berechnet, welche in der Servicekette beinhaltet sind. Beispielhaft dazu dient die folgende Zielfunktion (Strunk 2010, S. 69). Sie zeigt die Ausführungszeit der optimalen Servicekette aus Abbildung 3-5.

$$T = T(S_{1,1}) + T(S_{4,1}) + T(S_{5,1}) + T(S_{6,1}) + T(S_{7,1})$$

Die mathematische Berechnung der Zielfunktion ist abhängig von der Art der nicht-funktionalen Zielgrößen, d. h. ob diese Multiplikative Parameter, Additive Parameter oder Minimaloperator-Parameter sind, und von der Gestaltung der Aktionskette, d. h. ob es bspw. Schleifen oder Verzweigungen gibt. Lineare Zielfunktionen beschränken aber den Umfang der nicht-funktionale Zielgrößen auf jene, die nur linear aggregiert werden können. Beispielhaft für eine nicht-lineare Aggregation ist die Ausführungszeit bei einer Verzweigung, wo mehrere Aktionen gleichzeitig ausgeführt werden (siehe Tabelle 3-1). In solch einem Fall wird zwar die Komposition einer Servicekette ermöglicht, forciert aber gleichzeitig eine Einschränkung der Allgemeinheit. (Strunk 2010, S. 69)

II. Lokale Optimierung der nicht-funktionalen Zielgrößen: Eine lokale Optimierung bedeutet, dass anstelle der nicht-funktionalen Zielgrößen der gesamten globalen Servicekette die nicht-funktionalen Zielgrößen der Web Services zu den einzelnen Aktionen optimiert werden. D. h. jede Aktion wird einzeln betrachtet und der beste verfügbare Web Service wird selektiert. Begründet wird diese Vereinfachung dadurch, dass im Falle, dass für jede einzelne Aktion der optimale Web Service gewählt wurde, folglich auch die gesamte Servicekette optimal ist. Außerdem würde auch die Bewertung der Zielfunktion für die Berechnung der Gesamtausführungskosten wegfallen, da bereits für jede Aktion der optimale Web Service gewählt wurde und damit evaluiert ist. Dies verkürzt die Ausführungszeit und verringert die Komplexität der Komposition. (Strunk 2010, S. 69; Zeng et al. 2004, S. 322; Ardagna und Mirandola 2010, S. 1513) Die Nachteile der lokalen Optimierung sind die nicht mögliche globale Optimierung

der gesamten Web Service Komposition und der fehlende Ausgleich zwischen zwei Aktionen in einer parallelen Ausführung. So ist es etwa bei der Aktionskette in Abbildung 3-2 aus kostenwirtschaftlichen Gründen sinnvoll, die Ausführungszeiten der Aktionen „Flug suchen“, „Hotel suchen“ und „Mietwagen suchen“ an die längste von allen anzupassen. D. h. sollte die minimale Ausführungszeit der Aktion „Flug suchen“ bei 50 Millisekunden liegen und damit stets länger sein als die Ausführungszeiten der weiteren Aktionen, ist es sinnvoll die zwei weiteren Aktionen hinsichtlich der Ausführungszeit nicht zu optimieren, sondern an die der Aktion „Flug suchen“ anzupassen. Damit ist es möglich für die zwei weiteren Aktionen Web Services mit geringeren Ausführungskosten zu selektieren. Im Zuge der lokalen Optimierung ist es außerdem nicht möglich mehrere nicht-funktionale Zielgrößen zu betrachten. So ist es zwar möglich den Web Service mit den geringsten Ausführungskosten oder der kleinsten Ausführungszeit zu selektieren, es ist aber nicht möglich beide nicht-funktionalen Zielgrößen zu optimieren. Da beide nicht-funktionalen Zielgrößen negativ miteinander korrelieren, ist es unmöglich einen Web Service mit den geringsten Ausführungskosten und gleichzeitig der kleinsten Ausführungszeit zu selektieren. (Zeng et al. 2004, S. 322)

III. Globale Restriktionen des Nutzers nicht beachten: Restriktionen der Nutzer können bspw. sein, dass eine minimale Verfügbarkeit von 90% gegeben sein muss oder die Zuverlässigkeit mindestens 95% betragen muss. Verfahren, die solche Restriktionen der Nutzer nicht mit einbeziehen, sind in der Lage die Selektion zeiteffizienter zu gestalten. Jedoch hat diese Vereinfachung der Selektion den entscheidenden Nachteil, dass die nicht-funktionalen Zielgrößen der optimalen Servicekette nicht kontrolliert werden können. So ist es dabei nicht möglich bspw. eine Obergrenze für die Ausführungskosten zu definieren und die Ausführungskosten der optimalen Servicekette können unter Umständen das Budget des Nutzers übersteigen. (Strunk 2010, S. 69)

VI. Optimierung nur anhand einer nicht-funktionalen Zielgröße: In der Praxis ist ein Web Service Nutzer daran interessiert mehrere nicht-funktionale Zielgrößen einer Servicekette zu optimieren. So dürfen bspw. die Kosten ein bestimmtes Maximum nicht überschreiten, die Ausführungszeit einen bestimmten Wert nicht übersteigen und die Reputation sollte möglichst hoch sein. Jedoch entstehen bei Beachtung vieler nicht-funktionalen Zielgrößen Konflikte bzgl. der Optimierung. So kann die Ausführungszeit nicht gleichzeitig mit den Kosten minimal sein, da in der Regel die Kosten umso mehr steigen, je geringer die Ausführungszeit eines Web Services ist. Wie in Kapitel 3.1.2 bereits erwähnt, existiert in der Regel keine absolut optimale Servicekette, bei welcher alle nicht-funktionalen Zielgrößen optimal sind. Vielmehr existiert eine Reihe an sogenannten pareto-optimalen Serviceketten. Pareto-optimale Lösungen sind jene, die hinsichtlich aller nicht-funktionalen Zielgrößen nicht mehr verbessert werden können. D. h. bei einer Lösungsmöglichkeit kann keine nicht-funktionale Zielgröße verbessert werden, ohne eine andere schlechter zu stellen. In anderen Worten, es wird eine Reihe an Kompromisslösungen gefunden. Da die Selektion aller pareto-optimaler Serviceketten ungleich schwerer ist, als die Selektion einer Servicekette mit nur einer nicht-funktionalen Zielgröße, werden in vielen Selektionsverfahren nicht-funktionale Zielgrößen zu einer Zielgröße (Präferenzfunktionswert) zusammengefasst. Dabei werden die gewichteten einzelnen nicht-funktionalen Zielgrößen in einer globalen Zielfunktion aggregiert. Zur Berechnung der globa-

len nicht-funktionalen Zielgröße müssen dabei zuerst alle verwendeten nicht-funktionalen Zielgrößen normalisiert und anschließend per Gewichtung zusammengefasst werden. Der dabei entstandene Wert, welcher umso besser ist, je größer er ist, stellt eine globale nicht-funktionale Zielgröße dar. Die folgenden zwei Gleichungen (Strunk 2010, S. 70) definieren die Normalisierung der nicht-funktionalen Zielgrößen, die minimiert werden müssen (1) und die nicht-funktionalen Zielgrößen, die maximiert werden müssen (2). In der Gleichung stellt q den aktuellen Wert, q^{max} den maximalen Wert und q^{min} den minimalen Wert der nicht-funktionalen Zielgröße dar. Jede nicht-funktionale Zielgröße muss separat normalisiert werden. (Strunk 2010, S. 70)

$$q_{norm}^- = \begin{cases} \frac{q^{max} - q}{q^{max} - q^{min}} & \text{when: } q^{max} - q^{min} \neq 0 \\ 1 & \text{when: } q^{max} - q^{min} = 0 \end{cases} \quad (1)$$

$$q_{norm}^+ = \begin{cases} \frac{q - q^{min}}{q^{min} - q^{max}} & \text{when: } q^{min} - q^{max} \neq 0 \\ 1 & \text{when: } q^{min} - q^{max} = 0 \end{cases} \quad (2)$$

Nach der Normalisierung der nicht-funktionalen Zielgrößen werden diese zu einer globalen nicht-funktionalen Zielgröße zusammengefasst. Mit der folgenden Gleichung (3) (Strunk 2010, S. 70) werden die normalisierten gewichteten minimalen nicht-funktionalen Zielgrößen durch die normalisierten gewichteten maximalen nicht-funktionalen Zielgrößen dividiert. Der dabei entstandene Quotient Q ist die globale nicht-funktionale Zielgröße einer Servicekette. (Strunk 2010, S. 70)

$$Q = \frac{\sum_{i=1}^{n^-} w_i^- * q_{norm,j}^-}{\sum_{i=1}^{n^+} w_i^{+-} * q_{norm,j}^+} \quad (3)$$

Der Wert w_i stellt die Gewichtung dar, inwieweit eine einzelne nicht-funktionale Zielgröße auf die globale nicht-funktionale Zielgröße der Servicekette Einfluss hat. Die Summe der Gewichtungen w_i muss dabei exakt 1 entsprechen. Beispielhaft wird angenommen, dass die drei nicht-funktionale Zielgrößen Ausführungszeit, Verfügbarkeit und Zuverlässigkeit normalisiert worden sind. Die Gewichte verteilen sich der Reihenfolge nach mit 0,5, 0,25 und 0,25. Die Gleichung (4) (Strunk 2010, S. 70) zeigt die Berechnung der globalen nicht-funktionalen Zielgröße Q . Sie ergibt sich aus 0,5 multipliziert mit der normalisierten Ausführungszeit (az)

dividiert durch die Summe der mit 0,25 gewichteten normalisierten Verfügbarkeit (v) und Zuverlässigkeit (z). (Strunk 2010, S. 70)

$$Q = \frac{0,5 * az_{norm}}{0,25 * v_{norm} * 0,25 * z_{norm}} \quad (4)$$

Die Vereinfachung, die nicht-funktionalen Zielgrößen einer Servicekette zu einer globalen nicht-funktionalen Zielgröße zusammenzufassen, hat jedoch einen entscheidenden Nachteil. Die Qualität der optimalen Servicekette ist stark abhängig von den Gewichtungen der einzelnen normalisierten nicht-funktionalen Zielgrößen. Die große Herausforderung dabei ist, diese Gewichtungen richtig zu bestimmen. Des Weiteren ist eine Normalisierung nur möglich, wenn die nicht-funktionale Zielgröße einen minimalen und einen maximalen Wert hat. Da dies nicht immer der Fall ist müssen die minimalen und maximalen Werte im Vorhinein definiert werden. (Strunk 2010, S. 69-70)

V. Selektion einer suboptimalen Servicekette: Bei der Verwendung von heuristischen Verfahren wird nicht primär die Selektion der absolut optimalen Servicekette, sondern die Selektion einer nahezu optimalen Servicekette angestrebt. Da mit dem Anstieg der Größe des Optimierungsproblems der Rechenaufwand der Web Service Selektion exponentiell ansteigt (NP-Schwere des Optimierungsproblems), werden vor allem bei praxisrelevanten Problemstellungen Heuristiken zur Selektion eingesetzt. Zwar stellen Heuristiken die Selektion der optimalen Servicekette nicht sicher, aber dafür sind sie weniger rechenintensiv. (Mertens et al. 2001, S. 226) Konkret ermöglicht diese Vereinfachung geringere Selektionszeiten und Selektionskosten und macht daher heuristische Verfahren wesentlich effizienter als exakte Selektionsverfahren. Der Nachteil heuristischer Verfahren ist, dass aufgrund ihrer Verfahrensweise in den meisten Fällen nicht die optimale Servicekette selektiert wird. Diese kann während der Selektion nicht also solche erkannt und als Lösung abgelehnt worden sein, oder es wurde schon vorher ein Abbruchkriterium erreicht. (Strunk 2010, S. 70) Heuristische Selektionsmethoden und konkrete heuristische Web Service Selektionsverfahren werden in Kapitel 3.2.3.2 nochmals genauer betrachtet.

3.2.3 Explizite Selektionsverfahren - ein Optimierungsproblem

Die Verfahren zur Selektion von Web Services, basierend auf den vom Servicenutzer geforderten nicht-funktionalen Zielgrößen, lassen sich grundsätzlich in zwei Kategorien gliedern. Diese sind exakte Optimierungsmodelle und heuristische Optimierungsmodelle. (Strunk 2010, S. 70) Ein Optimierungsmodell bildet dabei ein Optimierungsproblem formal ab. Das Optimierungsproblem bezeichnet alle Entscheidungsprobleme, bei welchen hinsichtlich des angestrebten Ziels die bestmögliche Realisierung des Problems erreicht werden kann. In diesen Modellen enthalten sind Ursache-Wirkungs-Zusammenhänge, welche im Hinblick auf die Entscheidungsfindung zu berücksichtigen sind, und Zielrelationen, welche zur Bewertung und Auswahl von Handlungsmöglichkeiten erforderlich sind. Die Darstellung des Optimierungs-

problems erfolgt mittels eines mathematischen Modells. In diesem mathematischen Modell werden alle Aspekte bzgl. des realen Entscheidungsproblems mittels kardinal messbaren Größen beschrieben. So bezeichnen Variable die Elemente des realen Systems und werden mit Hilfe einer Gleichung oder Ungleichung ausgedrückt. (Berbner 2007, S. 74) Im Folgenden werden exakte und heuristische Web Service Selektionsverfahren dargestellt und deren Eigenheiten beschreiben.

3.2.3.1 Exakte Selektionsverfahren

Ein exaktes Selektionsverfahren ist in der Lage die optimale Lösung des Optimierungsproblems zu finden. D. h. gemessen an den Präferenzen des Nutzers wird die bestmögliche Servicekette selektiert.

Im Folgenden werden wesentliche und in der Literatur diskutierte, exakte Verfahren näher erläutert. Alle exakten Verfahren werden anschließend in Kapitel 3.2.4 zum Vergleich und zur Bewertung in den Tabellen 3-2, 3-3 und 3-4 aufgelistet.

Zeng et al. (2003) und Zeng et al. (2004) präsentieren die Middleware-Plattform AgFlow. Durch AgFlow wird die Komposition von Web Services anhand nicht-funktionaler Zielgrößen ermöglicht. Die Autoren stellen im Rahmen von AgFlow ein Selektionsverfahren vor, welches die Komposition bzw. Selektion auf globaler Ebene ausführt. Zum Vergleich stellt Zeng et al. (2004) ein lokales Selektionsverfahren vor, bei welchem jede Aktion einzeln betrachtet wird und ein optimaler Web Service selektiert wird. Beim lokalen Selektionsverfahren können im Gegensatz zum globalen Selektionsverfahren keine globalen Nutzerrestriktionen beachtet werden. Außerdem kann bspw. bei parallelen Abläufen kein Ausgleich gemacht werden. Hier sollte im Sinne der Kostenwirtschaftlichkeit etwa bei einer stets längeren Ausführungszeit von Web Service 1 der Aktion 1 die Ausführungszeit des Web Services 2 der parallelen Aktion 2 an Web Service 1 angepasst werden und daher nicht lokal optimiert werden. (Zeng et al. 2004, S. 322; Ma und Zhang 2008, S. 1095) Der verwendete Algorithmus ist bei beiden Verfahren die lineare Integer Programmierung. Dabei wird die optimale Servicekette anhand eines Präferenzfunktionswertes selektiert. Die Berechnung des Präferenzfunktionswertes erfolgt mit einer linearen Aggregationsfunktion, wobei die Ausführungskosten, Ausführungszeit, Reputation, erfolgreiche Ausführungsrate und die Verfügbarkeit aggregiert werden. (Zeng et al. 2004, S. 316-321) Während der Ausführung der optimalen Servicekette werden die nicht-funktionalen Zielgrößen der aufgerufenen Web Services einem Monitoring unterzogen. Im Falle einer signifikanten Abweichung der nicht-funktionalen Zielgrößen vom Sollwert wird ein Re-planning durchgeführt. (Zeng et al. 2004, S. 320-321; Berbner 2007, S. 130) Die Evaluierung der Selektionsverfahren von Zeng et al. (2004) zeigt, dass der Einsatz des globalen Selektionsverfahrens im Gegensatz zum lokalen Selektionsverfahren zu geringeren Ausführungskosten und Ausführungszeiten führt. Des Weiteren haben Messungen der Autoren ergeben, dass durch den globalen Optimierungsansatz Serviceketten mit besseren nicht-funktionalen Zielgrößen selektiert werden, sprich niedrigere Ausführungskosten und kürzere Ausführungszeiten haben. (Zeng et al. 2004, S. 326) Einschränkungen und Mängel des Verfahrens zeigen sich bei großen Serviceketten, d. h. bei Serviceketten, wo zu den jewei-

ligen Aktionen viele Web Services verfügbar sind, wie bspw. bei einer Aktionskette der Größe von 10^{15} oder mehr. Hier sind für eine Servicekette mit 15 Aktionen je Aktion 10 Web Services verfügbar. Das vorgestellte Verfahren stößt bei solch großen Kompositionen an seine Grenzen, da der Selektionsaufwand dabei zu viele Ressourcen benötigt und daher für zeitkritische Anwendungen nicht anwendbar ist. (Zeng et al. 2004, S. 326; Berbner 2007, S. 130; Yu et al. 2007, S. 24-25; Su et al. 2007, S. 286) Bei einer eingeschränkten Anzahl von Aktionen und dazu verfügbaren Web Services und einer limitierten Anzahl an Restriktionen jedoch ist die lineare Integer Programmierung bspw. gegenüber genetischen Algorithmen (siehe Kapitel 3.2.3.2) zu bevorzugen, da in diesem Fall die lineare Integer Programmierung kürzere Selektionszeiten benötigt. Wenn kein Bedarf besteht eine nicht-lineare Aggregationsfunktion zu verwenden, wie bspw. in Canfora et al. (2005a) angewendet, so bietet die lineare Integer Programmierung durchaus Vorteile bzgl. der Selektionszeit. (Canfora et al. 2005a, S. 1073)

Im Rahmen des METEOR-S Projekts (University of Georgia Computer Science Department 2002) stellen Aggarwal et al. (2004a) ein Web Service Selektionsverfahren vor. Aggarwal et al. (2004a) und Aggarwal et al. (2004b) verwenden zur Web Service Selektion ebenso wie Zeng et al. (2004) lineare Integer Programmierung. Gleich wie bei Zeng et al. (2004) der Fall ist dieses Verfahren für umfangreiche Aktionsketten nicht brauchbar, da die Komplexität der linearen Integer Programmierung mit dem Umfang des Optimierungsproblems exponentiell ansteigt. (Yu et al. 2007, S 24-25) Die vom Verfahren beachteten nicht-funktionalen Zielgrößen sind die Kosten, Ausführungszeit, Verfügbarkeit, Zuverlässigkeit und eine domainspezifische nicht-funktionale Zielgröße. (Aggarwal et al. 2004b) Laut Canfora et al. (2006) ist die Arbeit von Aggarwal et al. (2004a) die erste, in welcher es möglich ist, domainspezifische nicht-funktionale Zielgrößen zu definieren. Jedoch wird in der Arbeit nicht erwähnt, wie diese aggregiert werden können. (Canfora et al. 2006, S. 151) Ebenfalls präsentieren die Autoren in ihren Arbeiten keine genaue Evaluierung der Verfahren oder ein Re-planning Verfahren. (Berbner 2007, S. 133)

Yu und Lin (2005a; 2005c) stellen zwei Web Service Optimierungsmodelle vor, ein Graphen-Modell und ein kombinatorisches Modell. Mit beiden Modellen wird die optimale Servicekette anhand nur einer nicht-funktionalen Zielgröße selektiert, der Ausführungszeit. Der für das kombinatorische Verfahren verwendete Algorithmus von Pisinger zeigt sehr gute Laufzeiten und ist besonders bei großen Aktionsketten schneller als der CSP Algorithmus, welcher für das Graphen-Verfahren eingesetzt wird. Beim kombinatorischen Modell jedoch muss im Falle von mehreren Aktionsketten der Pisinger Algorithmus jede Aktionskette durchlaufen. Anschließend wird die Servicekette mit dem höchsten Nutzen gewählt. (Yu und Lin 2005a; Berbner 2007, S. 131) In Yu und Lin (2005b) werden beide Optimierungsmodelle erweitert, indem die Selektion anhand mehrerer nicht-funktionaler Zielgrößen ermöglicht wird. Wie auch in den vorhergehenden Arbeiten jedoch ist die Serviceselektion nur für sequentiell auszuführende Aktionsketten möglich. Das bedeutet es werden bspw. Schleifen oder Verzweigungen in sequentielle Abläufe konvertiert. (Yu und Lin 2005a, S. 26; Yu und Lin 2005b, S. 133) Das kombinatorische Modell definiert das Optimierungsproblem als Multi-Dimension

Multi-Choice 0-1 Knapsack Problem (MMKP) und das Graphen-Modell als Multi-Constraint Optimal Path (MCOP). Zur Selektion der optimalen Servicekette wird für das kombinatorische Modell der Branch and Bound Algorithmus (BBLP) und für das Graphen-Modell der MCSP Algorithmus eingesetzt. Die Evaluierung beider Modelle bzw. Algorithmen wird in der Arbeit von Yu und Lin (2005b) und Yu et al. (2007) durch Vergleiche mit heuristischen Verfahren gemacht. Dabei ist erkennbar, dass der MSCP Algorithmus bei Tests mit zwei nicht-funktionalen Zielgrößen und einer kleinen Anzahl an Aktionen und Web Service Kandidaten pro Aktion eine bessere Ausführungszeit der Selektion aufweist als der vorgestellte heuristische Algorithmus. Bei steigender Anzahl von Aktionen und verfügbaren Web Services pro Aktion sollte das heuristische Verfahren bevorzugt werden. Hier weisen die beiden Algorithmen BBLP und MCSP deutliche Schwächen bzgl. der Selektionszeit auf. (Yu und Lin 2005b, S. 140; Yu et al. 2007, S. 9) In Yu et al. (2007), einer verbesserten Fassung der bisherigen Arbeiten, werden beide Optimierungsmodelle dahingehend erweitert, dass nun Aktionsketten mit Schleifen, Verzweigungen und Auswahl selektiert werden können. (Yu et al. 2007, S. 2) Eine Echtzeitselektion ist mit den vorgestellten exakten Verfahren nicht möglich. (Yu et al. 2007, S. 9)

In Huang et al. (2009b) und Jiang et al. (2010) wird von den Autoren ein exaktes Selektionsverfahren vorgestellt, welches im ersten Schritt durch einen Filteralgorithmus die Anzahl der in Frage kommenden Web Services reduziert. D. h. der Suchraum für die Selektion der optimalen Servicekette wird anfangs eingeschränkt, indem erstens nichtausführbare Web Services und zweitens bzgl. der nicht-funktionalen Zielgrößen ungeeignete Web Services aussortiert werden. Im zweiten Schritt wird anhand der Ausführungszeit und des Durchsatzes die optimale Servicekette selektiert. (Huang et al. 2009b, S. 519-521) In der Evaluierung zeigt der Selektionsalgorithmus eine gute Leistungsfähigkeit. Veränderungen in der Anzahl der Web Services haben nur einen geringen Einfluss auf die Selektionszeit, was den Algorithmus stabil und skalierbar macht. (Huang et al. 2009b, S. 522) Das Verfahren selektiert die optimale Servicekette in sehr kurzer Zeit und laut Jiang et al. (2010, S. 48) ist sogar eine Echtzeitselektion möglich. Ebenso ist bei einer großen Anzahl von potentiellen Web Services die Selektionszeit im akzeptablen Bereich. (Bartalos und Bielíková 2010, S. 345) Das Selektionsverfahren von Huang et al. (2009b) wurde bei der Web Service Challenge 2009 in Wien mit dem ersten Platz in der Kategorie Performance Results ausgezeichnet. (Georgetown University 2009) Der Fokus des Web Service Challenge 2009 war Methoden zu entwickeln, die eine automatische Web Service Komposition anhand nicht-funktionaler Zielgrößen vornehmen. (Bartalos und Bielíková 2010, S. 345)

Ein weiteres exaktes Selektionsverfahren stellen Gao et al. (2006) in ihrer Arbeit über die Selektion optimaler Web Services durch dynamische Programmierung vor. Sie transformieren dabei das Problem der Web Service Selektion in die Suche nach dem längsten Pfad aus einem gewichteten mehrstufigen Graphen. Der genannte Graph ist zu vergleichen mit einem Servicekandidatengraph, wie er auch beim Graphen Modell in Yu et al. (2007) verwendet wird. Die Selektion erfolgt dynamisch durch einen Algorithmus, welcher auf dynamischer Prog-

rammierung basiert. Die nicht-funktionalen Zielgrößen werden zu einem Präferenzfunktionswert aggregiert. (Gao et al. 2006, S. 368-370; Wan et al. 2008, S. 473) Welche nicht-funktionalen Zielgrößen speziell verwendet werden, wird in der Arbeit nicht angegeben. Jedoch werden in der Arbeit u. a. die Ausführungszeit, Ausführungskosten und Zuverlässigkeit als Beispiele für nicht-funktionale Zielgrößen genannt. (Gao et al. 2006, S. 367) Außerdem werden keine Angaben darüber gemacht, inwieweit Konstrukte wie Schleifen und Verzweigungen vom Selektionsverfahren berücksichtigt werden können. Obwohl das Interface-Matching Problem nur auf die Beziehung zwischen zwei Web Services limitiert ist, ist hervorzuheben, dass Gao et al. (2005) die ersten waren, welche sich dem Interface-Matching Problem gewidmet haben. (Strunk 2010, S. 72; Wan et al. 2008, S. 473) Bezüglich der Leistungsfähigkeit des Selektionsverfahrens hat die Evaluierung der Autoren ergeben, dass die Selektion sehr zeitaufwendig ist. So werden bei einer größer werdenden Anzahl an Aktionen und einer großen Zahl verfügbarer Web Service Kandidaten die Selektionszeiten immer länger. (Gao et al. 2006, S. 369)

Wan et al. (2008) stellen in ihrer Arbeit ein Verfahren vor, das die gesamte Web Service Komposition, d. h. die Servicekette in mehrere kleinere Teilketten gliedert, um anschließend das Optimierungsproblem zu lösen. Angestrebt wird das Optimierungsproblem in einer Echtzeitselektion zu lösen. Hierzu wird von den Autoren ein rekursiver Branch and Bound Algorithmus eingesetzt. Zur Selektion der optimalen Servicekette werden die Ausführungskosten, Ausführungszeit, Zuverlässigkeit und Verfügbarkeit zu einem Gesamtpräferenzfunktionswert zusammengefasst. (Wan et al. 2008, S. 467-470) Die Evaluierung der Autoren zeigt, dass die Selektionszeit mit der Größe der Servicekette und Anzahl der verfügbaren Web Services pro Aktion linear ansteigt. (Wan et al. 2008, S. 472-473) Außerdem präsentieren Wan et al. (2008) als Weiterführung zu der Arbeit von Gao et al. (2006) ein 3-Schichten Web Service Optimierungsmodell (WS3LM), welches eine verbesserte Lösung des Interface-Matching Problems darstellen soll. (Wan et al. 2008, S. 468-469)

Viel Beachtung erhielt die Arbeit von Ardagna und Pernici (2007). Sie stellen darin ein Selektionsverfahren vor, welches in die MAIS (Multichannel Adaptive Information Systems) Architektur integriert ist. MAIS ist eine Plattform, die es ermöglicht, Prozesse mit Hilfe von Web Services flexibel zu gestalten und auszuführen. Das Optimierungsproblem wird als „Mixed Integer Programming (MILP)“ Problem betrachtet. (Ardagna und Pernici 2007, S. 370-371) Mittels einer Präferenzfunktion werden die Ausführungszeit, Verfügbarkeit, Ausführungskosten, Reputation und Datenqualität aggregiert und zur Selektion der Web Services verwendet. Die Datenqualität bezeichnet die Güte des Ausführungsergebnisses des Web Services, d. h. wie exakt das gelieferte Ergebnis des Web Services den Vorgaben des Nutzers entspricht. Jedoch können nur lineare Aggregationen durchgeführt werden, nichtlineare nicht-funktionale Zielgrößen wie Verfügbarkeit werden linearisiert. (Ardagna und Pernici 2007, S. 372+375) Eine Besonderheit des Verfahrens von Ardagna und Pernici (2007) ist, dass im Falle, dass keine optimale Servicekette gefunden worden ist, ein Verhandlungsprozess zwischen dem Serviceanbieter und Servicebroker angestoßen wird. Durch eine Verhandlungsmethode,

basierend auf einem Algorithmus, welcher in Faratin et al. (1998) und Comuzzi und Pernici (2005) vorgestellt wird, können der Preis und die Datenqualität mit dem Serviceanbieter verhandelt und angepasst werden. Bei erfolgreicher Verhandlung wird der Selektionsprozess erneut durchgeführt. (Ardagna und Pernici 2007, S. 372 + 376-377) Die Selektion der optimalen Servicekette erfüllt globale Restriktionen und ist außerdem laut den Autoren in Echtzeit möglich. (Ardagna und Pernici 2007, S. 382)

Eine Weiterentwicklung zu der Arbeit von Haddad et al. (2008), in der nur elementare und keine zusammengesetzten Web Services in eine Web Service Komposition eingesetzt werden konnten, wird in Haddad et al. (2010) vorgestellt. Bei Haddad et al. (2010) werden in den Selektionsverfahren neben den nicht-funktionalen Zielgrößen auch transaktionale Zielgrößen mit einbezogen, die eine zuverlässige Ausführung der Servicekette sicherstellen. Da auch eine optimale Web Service Komposition keine zuverlässige Ausführung garantiert, sind aus Sicht der Autoren Haddad et al. (2010) transaktionale Zielgrößen mit einzubeziehen. Das Selektionsverfahren bedient sich einer lokalen Optimierungsstrategie anhand der Ausführungszeit, Reputation, erfolgreichen Ausführungsrate, Verfügbarkeit und des Ausführungspreises. (Haddad et al. 2010, S. 75 + 80)

Ein weiteres exaktes Selektionsverfahren wird in Aiello et al. (2009) vorgestellt. Das von den Autoren vorgeschlagene Selektionsverfahren RuGQoS basiert auf einem „Backward Breadth First“ Algorithmus. Die Selektion wird mittels des Durchsatzes und der Ausführungszeit vollzogen. Das Selektionsverfahren hat an der Web Service Challenge 2009 in Wien teilgenommen und wurde mit dem vierten Platz in der Kategorie Performance Results ausgezeichnet. (Georgetown University 2009)

Cardellini et al. (2007) formulieren die Selektion der optimalen Servicekette als lineares Programmierungs-Optimierungsproblem. Die dabei verwendeten nicht-funktionalen Zielgrößen sind Ausführungszeit, Kosten und Verfügbarkeit. (Cardellini et al. 2007, S. 346)

In Huang et al. (2009a) wird ein Selektionsverfahren präsentiert, welches die Ausführungskosten, Ausführungszeit, Verfügbarkeit und Zuverlässigkeit zur Selektion der optimalen Servicekette verwendet. Das Selektionsverfahren stützt sich auf Multiple Criteria Decision Making (MCDM) und Integer Programmierung. (Huang et al. 2009a, S. 3309-3311)

3.2.3.2 Heuristische Verfahren

In der Praxis existieren in einem Prozessmodell oftmals sehr viele alternative Aktionsketten und zu den einzelnen Aktionen werden viele funktional gleiche Web Services angeboten. Folglich können daraus viele Serviceketten selektiert werden. Aus diesem Grund geht bei NP-schweren Problemen die optimale Lösungsfindung mit hohen Rechenzeiten einher. Um die-

sem Problem entgegenzuwirken, sucht man in der Praxis nicht wie bei exakten Verfahren immer nach der optimalen Lösung, sondern versucht dieser möglichst exakt zu entsprechen, d. h. dieser möglichst nahe zu kommen (siehe Vereinfachungen in Kapitel 3.2.2). Die Möglichkeit dafür bieten heuristische Verfahren (Heuristiken). Heuristiken geben im Gegensatz zu exakten Verfahren keine Garantie, dass eine optimale Servicekette gefunden wird, jedoch bieten sie einen enormen Laufzeitvorteil gegenüber den exakten Lösungsmethoden. In anderen Worten, sie bieten einen Kompromiss zwischen der benötigten Selektionszeit und der erzielten Lösungsgüte. (Berbner 2007, S. 76) In der Web Selektion eingesetzte mathematische Selektionsmethoden sind Simulated Annealing (SA), Tabu Search oder genetische Algorithmen (GA). Die genetischen Algorithmen zählen zu den Evolutionsstrategien. Die Evolutionsstrategien, Simulated Annealing, Tabu Search und weitere Optimierungsverfahren wie Ameisenalgorithmen oder die Neuronale Netze bilden die Klasse der naturanalogen Suchverfahren. Naturanaloge Netze sind ein Gebiet innerhalb der heuristischen Suchverfahren. (Mertens et al. 2001, S. 210)

3.2.3.2.1 Individuelle Selektionsverfahren

In Yu und Lin (2005b) modellieren die Autoren die Web Service Selektion einmal durch ein kombinatorisches Modell und einmal durch ein graphisches Modell. Das kombinatorische Modell definiert das Optimierungsproblem durch ein Multi-Dimension Multi-Choice 0-1 Knapsack Problem (MMKP). Das graphische Modell definiert das Optimierungsproblem als einen Multi-Constraint Optimal Path (MCOP). Zu beiden Modellen stellen die Autoren heuristische und exakte Verfahren vor. Die vorgestellten heuristischen Verfahren sind WS_HEU, WFlow und MCSP-K. Die Verfahren können mehrere nicht-funktionale Zielgrößen beachten, globale Nutzerrestriktionen beachten und sind in der Lage Aktionsketten mit Schleifen, Verzweigungen und Auswahl mit zu berücksichtigen. (Yu und Lin 2005b, S. 133; Yu et al. 2007, S. 2) Die Evaluierung der Algorithmen zeigt, dass die heuristischen Algorithmen eine sehr gute Laufzeit haben und der Nutzen und die Güte der Ergebnisse sehr nahe dem Optimum sind. Insbesondere bei umfangreichen Aktionsketten wird der Vorteil der heuristischen Verfahren deutlich. Hinsichtlich der Selektionszeit in Relation zur Güte des Ergebnisses übertreffen die vorgestellten heuristischen Verfahren exakte Verfahren deutlich. In diesem Zusammenhang wird auch eine Echtzeitselektion angesprochen. (Yu et al. 2007, S. 24) Die nicht-funktionalen Zielgrößen bei den Messungen sind die Ausführungskosten, Ausführungszeit, Verfügbarkeit und Zuverlässigkeit. (Yu et al. 2007, S 18-23; Berbner 2007, S. 131) In der Evaluierung der heuristischen Verfahren in Yu et al. (2007) ist zu bemängeln, dass nicht dargestellt wird, wie sich die Laufzeiteigenschaften und die Güte der Ergebnisse bei einer hohen Restriktionsstärke gestalten. (Berbner 2007, S. 131)

In seiner Dissertation stellt Berbner (2007, S. 83-109) ein heuristisches Selektionsverfahren vor, welches das Optimierungsproblem durch ein Eröffnungsverfahren (H1_RELAX_IP), bestehend aus zwei Schritten und einem anschließenden Verbesserungsverfahren (H2_SWAP oder H3_SIMUL_ANNEAL), löst. Im ersten Schritt des Eröffnungsverfahrens wird durch

bspw. den Simplex-Algorithmus das relaxierte Optimierungsproblem exakt gelöst. Im zweiten Schritt wird durch einen Backtracking-Algorithmus eine zulässige Servicekette selektiert. Weicht der nun selektierte Zielfunktionswert der gefundenen Servicekette zu weit von einer vorher definierten oberen Schranke ab, wird angenommen, dass eine nicht ausreichend gute Servicekette selektiert wurde. In diesem Fall wird durch das Verbesserungsverfahren auf lokaler Ebene die Optimierung fortgesetzt. D. h. der für die jeweilige Aktion zuvor definierte Web Service wird gegen einen anderen funktional gleichen ersetzt. Dieses Verbesserungsverfahren wird fortgeführt, bis sich eine Verbesserung des Zielfunktionswertes ergibt, anschließend wird der neue Web Service in die selektierte Servicekette übernommen. (Berbner 2007, S. 83-93) Einschränkungen des Selektionsverfahrens ergeben sich durch die Vereinfachung, dass zur Verringerung der Komplexität nur sequentielle Ausführungen und keine parallele Ausführung oder Schleifen beachtet werden. (Berbner 2007, S. 70; Berbner et al. 2006, S. 74) Die im Verfahren verwendeten nicht-funktionalen Zielgrößen sind die Ausführungszeit, Verfügbarkeit, Ausführungskosten und der Durchsatz. (Berbner 2007, S. 106) In der Arbeit von Berbner (2007, S. 110-129) wird auch ein Re-planning Verfahren vorgestellt.

Ye und Mounla (2008) präsentieren ein Selektionsverfahren, welches bereits existierende Serviceketten wiederverwendet, um die Selektionszeit zu verbessern. Die Web Service Selektion erfolgt durch eine Kombination aus Integer Programmierung, Case-based Reasoning (CBR) und einem genetischen Algorithmus. Muss eine komplett neue, bisher noch nicht existierende Servicekette selektiert werden, wird diese mittels Integer Programmierung selektiert. Anderenfalls wird durch das CBR aus einem Speicherort eine Reihe an geeigneten bereits existierenden Serviceketten selektiert. Aus diesen Serviceketten wird anschließend durch einen genetischen Algorithmus eine hinreichende Servicekette selektiert. (Ye und Mounla 2008, S. 62-63) Die Evaluierung des Verfahrens zeigt, dass das vorgestellte Selektionsverfahren weniger Selektionszeit benötigt als eine Selektion mittels reiner Integer Programmierung. (Ye und Mounla 2008, S. 67-69) Ausstehend bleibt jedoch ein Vergleich des Verfahrens mit anderen Selektionsverfahren wie bspw. Web Service Selektion durch ausschließlich genetische Algorithmen.

Laut Ardagna und Mirandola (2010) präsentieren Alrifai und Risse (2009) momentan eines der modernsten Selektionsverfahren, die „top performing state-of-the-art techniques“ (Ardagna und Mirandola 2010, S. 1513). Die Autoren stellen ein Verfahren vor, welches eine globale Optimierung mit einer lokalen Web Service Selektion verbindet. Das Ziel des Verfahrens ist es, globale Restriktionen (Ebene der gesamten Servicekette) in dekomponierte lokale Restriktionen (Aktionsebene) zu unterteilen und anschließend auf lokaler Ebene geeignete Web Services zu selektieren. Damit sollen die Vorteile der lokalen Optimierung, nämlich eine schnelle Selektionszeit (Echtzeitselektion) zu erzielen, und die Vorteile der globalen Optimierung, globale Restriktionen zu beachten, zusammengeführt werden. Um die optimale Dekomposition der globalen Restriktionen zu finden, wird der erste Schritt durch Misch-Integer Programmierung ausgeführt. Damit auf lokaler Ebene, gemessen an den nun lokalen Restriktionen, der bestmögliche Web Service selektiert werden kann, wird der zweite Schritt durch eine verteilte

lokale Selektion ausgeführt, welche den Web Service anhand des Nutzens wählt. (Alrifai und Risse 2009, S. 881+886-887; Alrifai et al. 2010, S. 12) Die Web Service Selektion soll erst bei der Anfrage des Nutzers nach einer Web Service Komposition erfolgen. Explizit wird damit von den Autoren eine Echtzeitselektion der Web Services angestrebt. Eine Echtzeitselektion soll auch in der zukünftigen Forschung unter anderem als Fokus gesetzt werden. (Alrifai und Risse 2009, S. 889; Alrifai et al. 2010, S. 15) Das Selektionsverfahren der Autoren zieht jedoch keine möglichen Abhängigkeiten oder Korrelationen von nicht-funktionalen Zielgrößen in Betracht. Dies kann auf der Ebene der lokalen Restriktionen besonders im Falle von sehr engen Restriktionen zu Dekompositionen führen, für welche keine passenden Web Services gefunden werden können, obwohl eventuell welche existieren. (Alrifai et al. 2010, S. 16) Dieses Problem wird in Alrifai et al. (2010) behoben. Außerdem verbessern die Autoren das Selektionsverfahren dahingehend, dass nur noch eine eingeschränkte Anzahl an Web Services pro Aktion betrachtet wird, die sogenannten Skyline Services. Skyline Services sind jene Web Services, welche hinsichtlich ihrer nicht-funktionalen Zielgrößen von keinem anderen funktional identischen Web Service dominiert werden. Trotz dieser ersten Verringerung der Web Service Kandidaten ist bei einer ursprünglichen Größe des Optimierungsproblems von bspw. 100⁵ (für fünf Aktionen existieren je 100 Web Service Kandidaten) die Anzahl der verbliebenen Web Service Kandidaten immer noch sehr groß. Dies liegt daran, dass einige nicht-funktionale Zielgrößen wie bspw. die Ausführungskosten und die Ausführungszeit negativ miteinander korreliert sind. Denn im Zuge der Web Service Selektion wird angenommen, dass die Ausführungszeit umso geringer ist, je höher der Ausführungspreis ist. Aus diesem Grund wird in einem weiteren Schritt nur noch eine Teilmenge der Skyline Services betrachtet und als Basis der Web Service Selektion verwendet. (Alrifai et al. 2010, S. 12 + 16-17)

Ein heuristisches Verfahren, das Tabu Search und Simulated Annealing verbindet, stellen Ko et al. (2008) vor. Die nicht-funktionalen Zielgrößen für die Web Service Selektion sind die Ausführungskosten, Ausführungszeit, Verfügbarkeit, erfolgreiche Ausführungsrate, Reputation und die Ausführungshäufigkeit. (Ko et al. 2008, S. 2082)

In Fan et al. (2011) wird ein Web Service Selektionsverfahren vorgestellt, das einen kooperativen Evolutions-Algorithmus einsetzt, der aus einer stochastischen Teilmengen Optimierung (Stochastic particle swarm optimization (SPSO)) und Simulated Annealing (SA) zusammengesetzt ist. Die Zusammensetzung der Selektionsmethoden hat den Vorteil, dass die Nachteile der Methoden von der jeweils anderen ausgeglichen werden. Konkret wird der Nachteil der SPSO, zu schnell ein nur lokales Optimum zu finden, und der langsamen Annäherungsgeschwindigkeit von SA jeweils von der anderen Methode aufgehoben. Der vorgestellte kooperative Evolutions-Algorithmus ist jedoch nur für die Web Service Selektion mit einer Zielvorgabe effektiv. Daher wird aufgrund der Notwendigkeit bei der Web Service Komposition, mehrere Zielvorgaben zu erreichen und die nicht-funktionalen Zielgrößen zu optimieren, die SPSO zu einer sogenannten Multi-objective SPSO erweitert. Die Multi-objective SPSO selektiert eine Reihe an pareto-optimalen Serviceketten. (Fan et al. 2011, S. 9736-9740) Die für die

Web Service Selektion verwendeten nicht-funktionalen Zielgrößen werden von den Autoren nicht exakt angegeben, jedoch verweisen sie auf eine Reihe nicht-funktionaler Zielgrößen wie Ausführungskosten, Ausführungszeit, Durchsatz, Verfügbarkeit, Zuverlässigkeit und Reputation. (Fan et al. 2011, S. 9736)

3.2.3.2.2 *Genetische Verfahren*

Die derzeit vielversprechendsten heuristischen Selektionsverfahren für Web Services sind Verfahren basierend auf genetischen Algorithmen. Das Prinzip der genetischen Algorithmen gleicht der biologischen Evolution (Vererbungslehre). Ausgangspunkt der Selektion ist eine zufällige Menge an Chromosomen (Individuen), auch bezeichnet als Ausgangspopulation. Aus dieser Ausgangspopulation werden anhand der Fitness gute Chromosomen selektiert. Durch genetische Operationen wie Rekombinierung und Mutation entsteht nun eine neue Population, welche eine neue Generation und die Basis für die nächste Selektion bildet. Jede Selektion bzw. Rekombinierung und Mutation verbessert die Qualität der folgenden Generation. Die guten Chromosomen einer jeden Population entsprechen einer immer besseren Lösung des Optimierungsproblems bzw. kommen dem Optimum immer näher. (Strunk 2010, S. 71; Mertens et al. 2001, S. 209)

Übertragen auf die Web Service Selektion sind Chromosomen die Serviceketten und die Selektion anhand der Fitness ist die Selektion anhand nicht-funktionaler Zielgrößen. Das genetische Selektionsverfahren wird so lange ausgeführt, bis ein vorher definiertes Abbruchkriterium erfüllt wird. Die Lösung des genetischen Algorithmus ist das während des Evolutionsprozesses erzeugte Chromosom mit dem besten Zielfunktionswert. (Strunk 2010, S. 71; Berner 2007, S. 132; Mertens et al. 2001, S. 209)

Die ersten Autoren, welche die Selektion durch genetische Algorithmen vorgeschlagen haben, waren Canfora et al. (2005a). (Strunk 2010, S. 71) Sie stellen eine Optimierungsmethode basierend auf genetischen Algorithmen vor, die im Selektionsverfahren die nicht-funktionale Zielgrößen Ausführungszeit, Ausführungskosten, Verfügbarkeit und Zuverlässigkeit berücksichtigt. (Canfora et al. 2005a, S. 1072) Des Weiteren ist die Möglichkeit gegeben, domain-spezifische nicht-funktionale Zielgrößen zu verwenden, wie sie in Kapitel 2.3 beschreiben werden. (Canfora et al. 2005a, S. 1071-1072) Das Selektionsverfahren ist fähig nicht-funktionale Zielgrößen zu berücksichtigen, die eine nicht lineare Aggregationsfunktion besitzen. Außerdem werden die Konstrukte Schleife, Verzweigung und Auswahl im Kontrollfluss beachtet. (Canfora et al. 2005a, S. 1072-1075) Um die Vorteile und Schwächen genetischer Algorithmen zu verdeutlichen, wird in der Arbeit Canfora et al. (2005a) mittels Experimenten der genetische Algorithmus mit der linearen Integer Programmierung verglichen, der bis dato meist angenommenen Methode. In der Analyse der Experimente zeigt sich, dass bei wenigen konkreten Web Service Kandidaten pro Aktion (bis ca. 17 Web Services) lineare Integer Programmierung vorzuziehen ist. Bei steigender Anzahl an konkreten Web Service Kandidaten jedoch bleibt die Selektionszeit des genetischen Algorithmus nahezu konstant, während die Selektionszeit bei der linearen Integer Programmierung exponentiell ansteigt. (Canfora et

al. 2005a, S. 1073-1075) Die Selektionszeit des genetischen Algorithmus wird vorwiegend durch die Anzahl der Aktionen und die Struktur der Aktionskette beeinflusst und nicht durch die Anzahl der konkreten Web Services pro Aktion. (Bebner 2007, S. 132) Eine Erweiterung des Selektionsverfahrens wird in Canfora et al. (2008) durch ein Re-binding vorgestellt. Um auf die ständige Veränderung der nicht-funktionalen Zielgrößen oder bspw. auf das Auftreten eines neuen Web Services zu reagieren, wird darin vorgeschlagen die nicht-funktionalen Zielgrößen der Servicekette immer neu zu berechnen, falls eine Änderung der nicht-funktionalen Zielgrößen der einzelnen Web Services vorliegt oder ein neuer Web Service erkannt wird. Damit bei einer einflusslosen Änderung der nicht-funktionalen Zielgrößen oder bei einer zu geringen Änderung kein Re-binding ausgelöst wird, findet eine Neuberechnung erst beim Überschreiten einer bestimmten Abweichung der nicht-funktionalen Zielgrößen vom vorherigen Wert statt. Bspw. wird ein Re-binding erst bei einer Änderung von mehr als 1,5% zum ursprünglichen Wert durchgeführt. (Canfora et al. 2008, S. 1757)

Eine signifikante Verbesserung der Selektionszeit gegenüber dem Verfahren von Canfora et al. (2005a) erreichen Gao et al. (2007) mit der Verwendung eines Tree-coding Genetic Algorithm (TGA) anstelle eines One-dimension-coding Genetic Algorithm, wie er in Canfora (2005a) vorgestellt wird. Dabei wird der Servicezuordnungsgraph durch eine Baumcodierung anstatt einer Reihencodierung ver- und entschlüsselt. Diese Weiterentwicklung bewirkt eine um 60% schnellere Selektionszeit der Web Services. (Gao et al. 2007, 365-366) Von diesem Wert abweichend wird in der Arbeit von Strunk (2010, S. 71-72) eine Verbesserung von nur 40% angegeben. Dies ist jedoch falsch, da in der Arbeit von Gao et al. (2007, S. 366) nicht von einer 40%igen Verbesserung gesprochen wird, sondern von einer Selektionszeit, die nur noch 40% der bisherigen Zeit benötigt, also um 60% schneller ist. Dies kann zudem auch aus der Darstellung 4 in der Arbeit von Gao et al. (2007, S. 366) entnommen werden. In Gao et al. (2007) werden der Tree-coding Genetic Algorithmus und der One-dimension-coding Genetic Algorithmus evaluiert, dabei wird unter anderem die Selektionszeit verglichen. (Gao et al. 2007, S. 368-370)

Abhängigkeiten und Konflikte von Web Services zu anderen Web Services (siehe Kapitel 3.1.4.2) wurden erstmals von Ai und Tang (2008) in einem Selektionsverfahren berücksichtigt. Die Autoren erweiterten darin den genetischen Algorithmus von Canfora et al. (2005a) zu einem strafbasierenden genetischen Algorithmus, welcher Abhängigkeiten zwischen Web Services berücksichtigt. (Ai und Tang 2008, S. 738; Stunk 2010, S. 72) Zur Web Service Selektion verwenden die Autoren die nicht-funktionalen Zielgrößen Ausführungszeit, Ausführungskosten, Reputation, Verfügbarkeit und Zuverlässigkeit, welche zu einem Gesamtpräferenzfunktionswert aggregiert werden. (Ai und Tang 2008, S. 738-740) Die Evaluierung des Verfahrens zeigt, dass der genetische Algorithmus effizient, effektiv und skalierbar ist. Die Skalierbarkeit wird dadurch deutlich, dass die Selektionszeit mit der Zunahme von Aktionen oder Restriktionen linear ansteigt und durch die Anzahl an konkreten Web Services pro Aktion nicht beeinflusst wird. Des Weiteren ist das Verfahren erweiterbar, ohne Veränderungen

vornehmen zu müssen, d. h. es können problemlos weitere nicht-funktionale Zielgrößen aufgenommen werden. (Ai und Tang 2008, S. 742-743)

Zhang et al. (2006a) und Zhang et al. (2006b) präsentieren ebenfalls eine Weiterentwicklung der Optimierungsverfahren basierend auf genetischen Algorithmen. Dabei wird für die Selektion ein neuartiger genetischer Algorithmus mit einem Zuordnungsmatrixzahlenschlüssel aus Chromosomen (Relation Matrix Coding Scheme of Chromosomes) verwendet. Erweiternd hinzu kommen ein spezieller Kreuzungsoperator und ein spezieller Mutationsoperator. (Zhang et al. 2006a, S. 224-225) In Zhang et al. (2006a, S. 230) und Su et al. (2007, S. 285) wird bemängelt, dass eindimensionale genetische Algorithmen, welche ein One-dimension-coding scheme verwenden, zu früh ein vermeintliches Optimum finden, d. h. glauben eine nahezu optimale Servicekette gefunden zu haben. Es wird nicht bedacht, dass die gefundene Servicekette unter Umständen nicht die bestmögliche Lösung, sondern nur eine lokal optimale Servicekette ist. Zudem können eindimensionale genetische Algorithmen ein Re-planning nicht ausreichend darstellen. Das Relation Matrix Coding Scheme ist im Gegensatz zum One-dimension-coding scheme in der Lage, gleichzeitig simultan alle möglichen Serviceketten zu selektieren und alle möglichen Konstrukte wie Schleifen oder parallele Ausführung zu beachten. (Su et al. 2007, S. 285; Ma und Zhang 2008; S. 1094-1097) In Zhang et al. (2008) stellen die Autoren CoDiGa vor, eine Weiterentwicklung des Genetischen Algorithmus und der bisher vorgestellten Arbeiten. CoDiGA zeigt eine bessere Selektionszeit und eine bessere Stabilität. Bei kleinen Aktionsketten ist das Verfahren jedoch nicht zu empfehlen, da keine bessere Selektionszeit erzielt werden kann und keine besseren Ergebnisse gefunden werden können als mit alternativen Selektionsverfahren. (Ma und Zhang 2008; S. 1094-1095) Diese Einschränkung des CoDiGA Web Service Selektionsverfahren geht aus der fehlenden Anpassbarkeit der Leistungsfähigkeit hervor. (Ma und Zhang 2008; S. 1103).

Den ersten multiobjektiven genetischen Algorithmus stellen Wada et al. (2008) mit dem Verfahren E^3 vor. Das Selektionsverfahren kann simultan mehrere SLAs beachten und selektiert alle zulässigen pareto-optimalen Serviceketten. (Strunk 2010, S. 72; Wada et al. 2008, S. 368) Die nicht-funktionalen Zielgrößen Latenz, Ausführungskosten und Durchsatz werden durch Wada et al. (2008) nicht zu einem Gesamtpräferenzfunktionswert zusammengefasst, sondern die einzelnen nicht-funktionalen Zielgrößen werden entsprechend dem Ausführungsplan aggregiert. (Wada et al. 2008, S. 368-370) So können Zielkonflikte wie bspw. die Minimierung der Ausführungskosten und die Maximierung des Durchsatzes vermieden werden. (Wada et al. 2008, S. 370) Besonders hervorzuheben ist das Verfahren von Wada et al. (2008) deshalb, weil es das einzige Verfahren ist, welches keine der in Kapitel 4.2.2.1 genannten Vereinfachungen benutzt. (Strunk 2010, S. 70-71)

Einen genetischen Algorithmus mit einer variablen Länge der Chromosomen (Ausgangspopulation), um damit eine Reihe an verschiedenen Aktionsketten darzustellen, präsentieren Jiang et al. (2011). Die Selektion wird anhand der Ausführungskosten, Ausführungszeit, Verfüg-

barkeit und Reputation durchgeführt, welche zu einem Gesamtpräferenzfunktionswert aggregiert werden. (Jiang et al. 2011, S. 113-116)

3.2.4 Strukturierter Vergleich und Bewertung der Verfahren zur Selektion von Web Services basierend auf nicht-funktionalen Zielgrößen

Die Tabellen 3-2, 3-3, 3-4 und 3-5 zeigen eine strukturierte Übersicht über alle in Kapitel 3.2.3 vorgestellten Selektionsverfahren. Sollen die Selektionsverfahren bewertet werden, muss zunächst definiert werden, was ein optimales Selektionsverfahren ist. Das perfekte Musterbeispiel wäre ein Selektionsverfahren, welches alle nicht-funktionalen Zielgrößen berücksichtigt, alle in Kapitel 3.1.4 genannten Herausforderungen bewältigen kann und keine der in Kapitel 3.2.2 genannten Vereinfachungen anwendet. Da jedoch ein solches Verfahren in der Praxis nicht existiert bzw. dies momentan noch nicht möglich ist, muss, um die Selektionsverfahren bewerten zu können, eine Gewichtung der Kriterien gemacht werden, welche ein Selektionsverfahren erfüllen muss.

3.2.4.1 Leistungsvermögen der Selektionsverfahren

Eine wesentliche Rolle in der Web Service Selektion spielen nicht-funktionalen Zielgrößen. Je mehr nicht-funktionale Zielgrößen ein Verfahren berücksichtigt, desto qualitativer kann die Selektion erfolgen. Jedoch ist eine zu große Zahl nicht-funktionaler Zielgrößen nicht effizient. Es ist sinnvoller eine bestimmte Anzahl an nicht-funktionalen Zielgrößen effizient zu optimieren als möglichst viele effektiv. Eine Reihenfolge nach Bedeutung der nicht-funktionalen Zielgrößen ist nicht direkt machbar. Abhängig vom Einsatzgebiet und Zweck des Web Services kann jedoch eine Menge an nicht-funktionalen Zielgrößen genannt werden, welche zwingend berücksichtigt werden müssen, sozusagen die „must have“. Des Weiteren können eventuell gewünschte nicht-funktionale Zielgrößen hinzugenommen und berücksichtigt werden. Die Wahl der einzubindenden nicht-funktionalen Zielgrößen ist aber in erster Linie abhängig von der Funktionalität des Web Services und den Präferenzen des Web Service Nutzers. Dennoch sind die Wahl und eine Einteilung der nicht-funktionalen Zielgrößen erforderlich. (Malan und Bredemeyer 2001, S. 7) Umgesetzt werden kann dies in ähnlicher Form wie die Gewichtung der nicht-funktionalen Zielgrößen im Verfahrensbeispiel in Kapitel 3.2.1. Dort wird, wie auch beschrieben in Yu et al. (2007), Yu und Lin (2005c) und Berbner (2007, S. 52), durch eine Gewichtung die Stärke des Einflusses einer nicht-funktionalen Zielgröße auf den Nutzen eines Web Services und somit auch auf die Web Service Auswahl festgelegt. (Berbner 2007, S. 52; Yu und Lin 2005c, S. 112) In gleicher Form können benötigte oder nicht benötigte nicht-funktionale Zielgrößen gefunden werden.

Da Web Services als Teil des Business Activity Outsourcing Geschäftsprozesse in Unternehmen durchführen und Geschäftsprozesse in erster Linie aus Kostengründen ausgelagert werden, sind die Ausführungskosten eines Web Services eine der wichtigsten nicht-funktionalen

Zielgrößen. (Braunwarth und Heinrich 2008, S. 107) Da Unternehmen ihre Prozesse zeitlich punktgenau durchführen müssen, nehmen wir als weitere erfolgskritische nicht-funktionale Zielgrößen für Geschäftsprozesse die Ausführungszeit und die Verfügbarkeit. Bei den exakten Verfahren werden diese nicht-funktionalen Zielgrößen von den Verfahren von Zeng et al. (2004), Aggarwal (2004a und 2004b), Yu und Lin (2005a, 2005b und 2005c), Yu et al. (2007), Wan et al. (2008), Ardagna und Pernici (2007), Haddad et al. (2010), Cardellini et al. (2007) und durch das Verfahren von Huang et al. (2009a) beachtet. Die heuristischen Verfahren, welche die genannten wichtigen nicht-funktionalen Zielgrößen beachten, sind die Selektionsverfahren von Yu und Lin (2005b) und Yu et al. (2007), Berbner (2007), Berbner et al. (2006), Alrifai und Risse (2009), Alrifai et al. (2010), Ko et al. (2008), Canfora et al. (2005a), Canfora et al. (2008), Gao et al. (2007), Ai und Tang (2008), Su et al. (2007) und Ma und Zhang (2008) und das Verfahren von Jiang et al. (2011). Die meisten nicht-funktionalen Zielgrößen werden in den Selektionsverfahren von Alrifai und Risse (2009), Alrifai et al. (2010) und Ko et al. (2008) mit einbezogen.

Bezüglich der Vereinfachungen der Selektion (Kapitel 3.2.2) ist aus den Tabellen 3-2, 3-3, 3-4 und 3-5 herauszulesen, dass nur zwei heuristische Verfahren und kein exaktes Verfahren sich keiner der genannten Vereinfachungen bedienen. Außer den heuristischen Selektionsverfahren E^3 von Wada et al. (2008) und dem Selektionsverfahren von Fan et al. (2011) nutzen alle genannten Selektionsverfahren eine oder mehrere der genannten Vereinfachungen.

Die Herausforderungen der Web Service Selektion aus Kapitel 3.1.4 sind bereits von einigen Autoren aufgegriffen worden. Eine Echtzeitselektion wird bspw. von Jiang et al. (2010, S. 48) und Ardagna und Pernici (2007, S. 382) konkret angesprochen. Erwähnenswert ist hierbei, dass es sich bei beiden Verfahren um exakte Selektionsverfahren handelt. Wie bereits bei den Vereinfachungen in Kapitel 3.2.2 genannt, ist aufgrund der Tatsache, dass heuristische Verfahren eine Zeitersparnis durch Selektion einer suboptimalen Servicekette erzielen können, besonders viel Potential zur Echtzeitselektion gegeben. Eine Echtzeitselektion zum Zeitpunkt der Ausführung des Prozesses zu erreichen wird von den Autoren heuristischer Verfahren jedoch vielfach nur angesprochen. So wird bei Alrifai und Risse (2009 S. 889) und Alrifai et al. (2010, S. 15) erwähnt, dass die Web Service Selektion zum Zeitpunkt des Aufrufes der Servicekette erfolgt. Konkrete Aussagen und ein Beweis durch eine Evaluierung, dass Echtzeitselektionen möglich sind, werden von den Autoren nicht gemacht.

Abhängigkeiten zwischen nicht-funktionalen Zielgrößen sind von Ai und Tang (2008) erstmals berücksichtigt worden. Jedoch wurde bisher in keinem weiteren Verfahren die Abhängigkeiten in die Web Service Selektion mit einbezogen, d. h. es gibt bis dato auch kein Verfahren, das eine zufriedenstellende Lösung der Herausforderung bietet.

Das Problem des Zusammenpassens der Schnittstellen (Interface-Matching) wird ebenfalls nur von wenigen Autoren angesprochen. Erstmals von Gao et al. (2005) in einem Selektionsverfahren berücksichtigt, hat Wan et al. (2008) das Interface-Matching aufgegriffen und versucht es weiter zu optimieren.

3.2.4.2 Entwicklung der Selektionsverfahren

Eine Weiterentwicklung der Web Service Selektionsverfahren ist im Verlauf der vergangenen Jahre deutlich zu beobachten. So ist der genetische Algorithmus von Canfora et al. (2005a) von vielen Autoren weiterentwickelt worden. Gao et al. (2007) bspw. haben eine wesentliche Verbesserung der Selektionszeit erreicht. Generell ist Folgendes zu beobachten: Um eine bessere Leistungsfähigkeit der Selektionsverfahren zu erzielen, sind heuristische Verfahren im Moment die bessere Lösung als exakte Verfahren. Besonders Selektionsverfahren, welche sich eines genetischen Algorithmus bedienen, sind derzeit die vielversprechendsten Ansätze. (Strunk 2010, S. 71) Dies zeigt sich vor allem dadurch, dass in jüngerer Zeit viele Autoren Selektionsverfahren veröffentlichten, welche die Web Service Selektion mit genetischen und weiterentwickelten genetischen Algorithmen durchführen. Erfunden wurden genetische Algorithmen in den frühen siebziger Jahren. (Srinivas und Patnaik 1994, S. 26) Genetische Algorithmen basieren auf der Evolutionstheorie von Darwin. (Su et al. 2007, S. 285) Sie eignen sich für die Web Service Selektion, weil die Evolutionstheorie aus Sicht von Mathematikern, Ingenieuren und Informatikern ein außerordentlich leistungsstarkes Optimierungsverfahren ist. (Schöneburg et al. 1994, Seite 13) Jedoch werden das Verhalten und die Performance genetischer Algorithmen wesentlich durch deren Design beeinflusst. Insbesondere das Design der verwendeten Parameter ist essentiell. (Roja et al. 2002, S. 31)

Eine Übersicht über die Entwicklung der Web Service Selektionsverfahren stellen Ardagna und Mirandola (2010, S. 1513-1514) vor. Die Autoren teilen darin die Selektionsverfahren in drei Generationen ein. Die erste Generation betrifft lokale Optimierungsverfahren wie die von Zeng et al. (2004), Yu et al. (2007) und Ardagna und Pernici (2007). Lokale Optimierungsverfahren sind sehr simpel und können nur lokale Restriktionen beachten, d. h. die Optimierung findet auf Aktionsebene statt. Einfacher ausgedrückt wird zu jeder Aktion derjenige Web Service selektiert, welcher den Anforderungen des Nutzers am besten entspricht. Bspw. wird aus denjenigen Web Services, welche unterhalb einer bestimmten Preisgrenze liegen, der Beste selektiert. (Ardagna und Mirandola 2010, S. 1513)

Die zweite Generation der Selektionsverfahren beschreibt globale Optimierungsverfahren. In Verfahren wie dem von Zeng et al. (2004) oder Canfora et al. (2008) werden die Serviceketten anhand globaler Restriktionen selektiert. Das erste globale Selektionsverfahren wurde in Zeng et al. (2004) vorgestellt, wo das Optimierungsproblem durch lineare Integer Programmierung gelöst wurde. Im Zusammenhang mit der globalen Optimierung sind auch Selektionsverfahren entwickelt worden, welche genetische Algorithmen verwenden, bspw. in Canfora et al. (2008). Selektionsverfahren der zweiten Generation erfordern durch die Lösung des NP-schweren Optimierungsproblems einen wesentlich höheren Selektionsaufwand. Um daher die Komplexität zu verringern, bedienen sich Verfahren der zweiten Generation Vereinfachungen und Einschränkungen (siehe Kapitel 3.2.2). So werden bei Zeng et al. (2004) globale Restriktionen nur für den kritischen Pfad garantiert, also jenen mit der höchsten Ausführungszeit. Oder es werden Schleifen sozusagen aufgefaltet, damit eine Schleife mit maximal einem vorher definierten Wert durchlaufen wird. (Ardagna und Mirandola 2010, S. 1513-1514)

Die dritte Generation der Selektionsverfahren versucht die Einschränkungen und Vereinfachungen der ersten und zweiten Generation zu überwinden. So ist bspw. die Vereinfachung,

eine Schleife aufzufalten, wie in Zeng et al. (2004), durch andere Autoren wie Ardagna und Pernici (2007) verbessert worden, indem Schleifen geschält werden, indem die erwartete Anzahl der Durchläufe der Schleife und nach jedem Durchlauf die Wahrscheinlichkeit, dass die Schleife verlassen wird, beachtet werden. (Canfora et al. 2008, S. 1767) Oder, sollte eine machbare Lösung des Optimierungsproblems nicht gefunden werden, empfehlen Ardagna und Pernici (2007) einen Verhandlungsprozess anzustoßen, mit dem eine Lösung gefunden werden soll. (Ardagna und Mirandola 2010, S. 1514)

Die folgenden Tabellen 3-2, 3-3, 3-4 und 3-5 zeigen eine Übersicht über alle in Kapitel 3.2.3 beschriebenen Web Service Selektionsverfahren. Die Tabelle verdeutlicht unter anderem, anhand welcher nicht-funktionalen Zielgrößen selektiert wird und welche Algorithmen dabei eingesetzt werden. Außerdem ist angegeben, ob eine Echtzeitselektion möglich ist oder diese angesprochen wird und ob das Selektionsverfahren ein Re-planning beinhaltet.

Verfahren	Zielfunktion	Optimierung	Optimierung nicht-funktionaler Zielgrößen	Nicht-funktionale Zielgrößen	Globale Restriktionen	Angewendeter Algorithmus	Echtzeit-selektion	Re-planning
Zeng et al. (2004)	Linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Reputation, erfolgreiche Ausführungsrate, Verfügbarkeit	Werden beachtet	Lineare Integer Programmierung	Nicht möglich	Wird beachtet
Aggarwal (2004a) (2004b)	Linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Verfügbarkeit, Zuverlässigkeit, domainspezifische nicht-funktionale Zielgröße	Werden beachtet	Lineare Integer Programmierung	Nicht möglich, wird aber angesprochen	Wird nicht beachtet
Yu und Lin (2005a), (2005b), (2005c) Yu et al. (2007)	Linear	Global	Präferenzfunktion	Ausführungszeit, Verfügbarkeit, Ausführungskosten. Evaluierung durch 2 bis 5 nicht-funktionalen Zielgrößen	Werden beachtet	Branch and Bound	Nicht möglich	Wird beachtet
Yu und Lin (2005a), (2005b), (2005c) Yu et al. (2007)	Linear	Lokal	Präferenzfunktion	Ausführungszeit, Verfügbarkeit, Ausführungskosten. Evaluierung durch 2 bis 5 nicht-funktionalen Zielgrößen	Werden beachtet	MCSP	Nicht möglich	Wird beachtet
Hunag et al. (2009), Jiang et al. (2010)	Nicht-linear	Lokal	Präferenzfunktion	Ausführungszeit, Durchsatz	Werden nicht beachtet	Modifizierte dynamische Programmierung	Ist möglich	Wird nicht beachtet
Gao et al. (2006)	Linear	Lokal	Präferenzfunktion	Keine exakten Angaben. Ausführungspreis, Ausführungszeit, Zuverlässigkeit etc.	Werden nicht beachtet	Dynamische Programmierung	Wird angesprochen	Wird nicht beachtet

Tabelle 3-2 Übersicht exakte Selektionsverfahren 1 in Anlehnung an Strunk (2010, S. 71)

Verfahren	Zielfunktion	Optimierung	Optimierung nicht-funktionaler Zielgrößen	Nicht-funktionale Zielgrößen	Globale Restriktionen	Angewandeter Algorithmus	Echtzeit-selektion	Re-planning
Wan et al. (2008)	Nicht-linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Zuverlässigkeit, Verfügbarkeit	Werden beachtet	Auf Divide-and-Conquer basierender Algorithmus	Wird angestrebt	Wird nicht beachtet
Ardagna und Pernici (2007)	Linear	Global	Präferenzfunktion	Ausführungszeit, Verfügbarkeit, Ausführungskosten, Reputation, Datenqualität	Werden beachtet	Misch-Integer Programmierung	Ist möglich	Wird beachtet
Haddad et al. (2010)	Linear	Lokal	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Reputation, erfolgreiche Ausführungsrate, Verfügbarkeit	Werden beachtet	Multiple Criteria Decision Making (MCDM)	Wird angesprochen	Wird nicht beachtet
Aiello et al. (2009)	Nicht-linear	Global	Separate Zielfunktionen. Anschließende Kombination	Durchsatz, Ausführungszeit	Werden nicht beachtet	Backward Breadth First	Wird nicht angesprochen	Wird nicht beachtet
Cardellini et al. (2007)	Linear	Global	Präferenzfunktion	Ausführungszeit, Ausführungskosten, Verfügbarkeit	Werden beachtet	Lineare Programmierung	Wird nicht angesprochen	Wird beachtet
Huang et al. (2009a)	Linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Verfügbarkeit, Zuverlässigkeit	Werden beachtet	Multiple Criteria Decision Making (MCDM) und Integer Programmierung	Wird nicht angesprochen	Wird nicht beachtet

Tabelle 3-3 Übersicht exakte Selektionsverfahren 2 in Anlehnung an Strunk (2010, S. 71)

Verfahren	Zielfunktion	Optimierung	Optimierung nicht-funktionaler Zielgrößen	Nicht-funktionale Zielgrößen	Globale Restriktionen	Angewendeter Algorithmus	Echtzeit-selektion	Re-planning
Yu und Lin (2005b) Yu et al. (2007)	Linear	Global	Präferenzfunktion	Ausführungszeit, Verfügbarkeit, Ausführungskosten. Evaluierung durch 2 bis 5 nicht-funktionalen Zielgrößen	Werden beachtet	MSCP-K, WS_HEU, WFlow	Wird angesprochen	Wird nicht beachtet
Berbner (2007) Berbner et al. (2006)	Linear	Global	Präferenzfunktion	Ausführungszeit, Verfügbarkeit, Durchsatz, Ausführungskosten	Werden beachtet	Integer Programmierung + Heuristik	Wird angesprochen	Wird beachtet
Ye und Mounla (2008)	Linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit	Werden beachtet	Kombination aus Integer Programmierung, Case-based Reasoning, genetischer Algorithmus	Wird nicht angesprochen	Wird nicht beachtet
Alrifai und Risse (2009) Alrifai et al. (2010)	Linear	Global	Einzelne Aggregation	Ausführungszeit, Ausführungskosten, Verfügbarkeit, Zuverlässigkeit, Durchsatz, Reputation, domainspezifische nicht-funktionale Zielgrößen	Werden beachtet	Misch-Integer Programmierung	Ist möglich	Wird nicht beachtet
Ko et al. (2008)	Nicht-linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Verfügbarkeit, Erfolgreiche Ausführungsrate, Reputation, Ausführungshäufigkeit	Werden beachtet	Kombination aus Tabu Search und Simulated Annealing	Akzeptable Selektionszeit wird angesprochen	Wird nicht beachtet

Tabelle 3-4 Übersicht heuristische Selektionsverfahren 1 in Anlehnung an Strunk (2010, S. 72)

Verfahren	Zielfunktion	Optimierung	Optimierung nicht-funktionaler Zielgrößen	Nicht-funktionale Zielgrößen	Globale Restriktionen	Angewendeter Algorithmus	Echtzeit-selektion	Re-planning
Fan et al. (2011)	Nicht-linear	Global	Pareto-optimale Serviceketten	Werden nicht speziell genannt	Werden beachtet	Kooperativer Evolutions-Algorithmus	Wird angesprochen	Wird nicht beachtet
Canfora et al. (2005a) Canfora et al. (2008)	Nicht-linear	Global	Präferenzfunktion	Ausführungszeit, Ausführungskosten, Verfügbarkeit, Zuverlässigkeit	Werden beachtet	Genetischer Algorithmus	Wird angesprochen	Wird beachtet
Gao et al. (2007)	Nicht-linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Verfügbarkeit, Zuverlässigkeit	Werden beachtet	Weiterentwickelter genetischer Algorithmus	Wird angesprochen	Wird beachtet
Ai und Tang (2008)	Nicht-linear	Global	Präferenzfunktion	Ausführungszeit, Ausführungskosten, Reputation, Verfügbarkeit, Zuverlässigkeit	Werden beachtet	Genetischer Algorithmus	Wird nicht konkret angesprochen.	Wird nicht beachtet
Su et al. (2007) Ma und Zhang (2008)	Nicht-linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Reputation, Verfügbarkeit	Werden beachtet	Weiterentwickelter genetischer Algorithmus	Wird nicht angesprochen	Wird beachtet
Wada et al.(2008)	Nicht-linear	Global	Pareto-optimale Serviceketten	Durchsatz, Ausführungskosten, Latenz	Werden beachtet	Genetischer Algorithmus	Wird angesprochen	Wird nicht beachtet
Jiang et al. (2011)	Nicht-linear	Global	Präferenzfunktion	Ausführungskosten, Ausführungszeit, Verfügbarkeit Reputation	Werden beachtet	Genetischer Algorithmus	Wird nicht angesprochen	Wird beachtet

Tabelle 3-5 Übersicht heuristische Selektionsverfahren 2 in Anlehnung an Strunk (2010, S. 72)

3.3 Re-planning

Wie bereits in Kapitel 3.1.2 erwähnt, müssen Ausfälle oder Laufzeitveränderungen der Web Services während der Ausführung berücksichtigt werden, da in der Praxis der Einsatz von Web Services oft nicht verläuft, wie ursprünglich in der Planung vorgesehen. Beinahe immer unterscheiden sich die vorhergesagten nicht-funktionalen Zielgrößen von den tatsächlich eintreffenden. (Canfora et al. 2005b, S. 121) Das Laufzeitverhalten der Web Services ist von vielen Umweltfaktoren abhängig, welche sich durch nicht-funktionale Zielgrößen ausdrücken lassen. So sind etwaige Gründe für verändertes Laufzeitverhalten ein Serverausfall beim Web Service Anbieter oder die nicht hundertprozentige Sicherheit einer störungsfreien Internetnutzung. Ausgedrückt werden solche Unsicherheiten durch die Ausfallwahrscheinlichkeit, wobei viele Risiken wie etwa ein schwankendes Verkehrsaufkommen und Serverauslastung oder Probleme seitens des Telekommunikationsanbieters kaum bis überhaupt nicht in die Größe Ausfallwahrscheinlichkeit mit einberechnet werden können. (Berbner 2007, S. 110) Hat bspw. der Web Service $S_{6,1}$ aus dem Verfahrensbeispiel aus 3.2.1 eine Ausfallwahrscheinlichkeit von 0,2 %, so kann dieser jederzeit ausfallen, auch während der Ausführung einer Servicekette. Anders ausgedrückt fällt der Web Service $S_{6,1}$ pro 500 Ausführungen der Wahrscheinlichkeit nach einmal aus. Fällt nun dieser oder auch ein anderer Web Service während der Ausführung der optimalen Servicekette aus, kommt ein Re-planning-Mechanismus zum Einsatz.

3.3.1 Definition Re-planning

Damit ein Prozess auch während der Ausführung, also während der Laufzeit, ausführbar bleibt, die vorher definierten Restriktionen des Nutzers eingehalten werden und die Präferenzen des Nutzers weiter berücksichtigt werden, wird durch Re-planning auf Abweichungen zwischen den geplanten Laufzeitverhalten und dem tatsächlichen Laufzeitverhalten reagiert. Kurz gesagt stellt Re-planning sicher, dass ein Web Service Prozess ausführbar bleibt, wenn bspw. ein einzelner Web Service während der Ausführung des Prozesses ausfällt. (Berbner 2007, S. 111) Konkret hat Re-planning mehrere Aufgaben, die in den folgenden Kapiteln genauer erläutert werden.

3.3.1.1 Sicherstellung der Ausführbarkeit

Der Ausfall eines einzelnen Web Services hätte ein Scheitern des gesamten Prozesses zur Folge. Damit ein solches Szenario ausbleibt, wird sozusagen zur Beobachtung der Web Services ein Monitoring-Mechanismus eingesetzt. Ein Monitor-Mechanismus erkennt, wenn ein Web Service nicht mehr ausgeführt werden kann bzw. abgebrochen wurde, und leitet ein Re-planning Verfahren ein. Im einfachsten Fall wird durch Re-planning der ausgefallene Web Service einer Aktion durch einen alternativen funktional gleichen Web Service ersetzt. (Berbner 2007, S. 111)

Für ein Re-planning Verfahren existieren zwei Arten von Optimierungsstrategien: eine lokale Optimierungsstrategie und eine globale Optimierungsstrategie. Wird eine lokale Optimierungsstrategie verfolgt, wird der ausgefallene Web Service durch denjenigen ersetzt, der die zweitbeste Bewertung erhalten hat, sprich den nächstbesten funktional gleichen Web Service. Wird jedoch eine globale Optimierungsstrategie verfolgt, kann nicht einfach auf den nächstbesten Web Service zurückgegriffen werden. Mit einer globalen Optimierungsstrategie kann dieses Problem auf zwei Lösungswegen behoben werden. Lösung 1: Hier muss der noch nicht ausgeführte Teil des Prozesses anhand der vorher definierten Präferenzen und Restriktionen und durch Anpassung der Nebenbedingungen erneut berechnet werden. Dies kann unter Umständen auch zu einer erneuten kompletten Selektion des noch nicht ausgeführten Teils der Aktionskette führen. (Berbner 2007, S. 111; Heinrich et al. 2011b, S. 93) Lösung 2: Alternativ kann auch eine andere Aktionskette gestartet werden. Bereits ausgeführte Web Services und deren nicht-funktionale Zielgrößen wie etwa die Ausführungskosten sind für beide Lösungen zum Entscheidungszeitpunkt nicht mehr relevant. Sollte ein bereits ausgeführter Web Service Teil einer alternativen Aktionskette sein, muss diese neu berechnet werden. In der Realität ist dies häufig der Fall, da alternative Aktionsketten gleiche Aktionen beinhalten und folglich gleiche Web Services beinhalten können. (Heinrich et al. 2011b, S. 93)

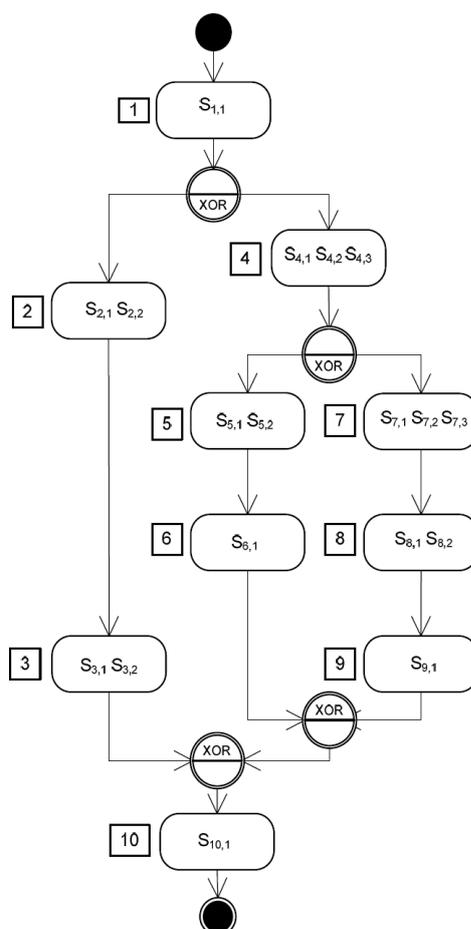


Bild 3-7 Beispiel für die Notation des SZG (Heinrich et al. 2011b, S. 92)

Derartige Überschneidungen lassen sich anhand der Abbildung 3-7 verdeutlichen. Aus der Abbildung 3-7 ist ersichtlich, dass die Aktionen 1 und 4 sowohl in der Aktionskette [1-4-5-6-10] als auch in der Aktionskette [1-4-7-8-9-10] zu finden sind. Demzufolge ist mindestens Web Service $S_{1,1}$ und einer der Web Services $S_{4,1}$, $S_{4,2}$ oder $S_{4,3}$ bereits ausgeführt worden. Fällt bspw. der Web Service $S_{9,1}$ der Aktion 9 aus, muss demnach der Web Service $S_{1,1}$ und der selektierte Web Service der Aktion 4 bei der Durchführung der alternativen Aktionskette [1-4-5-6-10] nicht mehr erneut ausgeführt werden. Wird die Aktionskette [1-4-7-8-9-10] weitergeführt, müssen die Aktionen 9 und 10 erneut berechnet werden. Bei Ausführung der alternativen Aktionskette [1-2-3-10] muss die Aktion 1 nicht mehr in die erneute Berechnung mit eingezogen werden. (Heinrich et al. 2011b, S. 93)

3.3.1.2 Gewährleistung der Gültigkeit des Prozesses hinsichtlich der Restriktionen des Nutzers

Wie bereits eingangs unter Kapitel 3.3 Re-planning erwähnt wurde, kann es bspw. aufgrund von starken Verkehrsaufkommen auf einem Server oder einer hohen Auslastung einer Internetverbindung zu Verzögerungen bei der Ausführung eines Web Service kommen. Diese Verzögerung wirkt sich negativ auf die Gesamtantwortzeit aus und kann zu einer Überschreitung der zulässigen Gesamtantwortzeit führen. Ein Monitoring-Mechanismus erkennt auch solche Probleme und leitet ein Re-planning ein. Der noch nicht ausgeführte Teil der Aktionskette wird unter Beachtung der definierten Restriktionen, d. h. im Anbetracht des zugrunde liegenden Optimierungsproblems und unter veränderten Nebenbedingungen neu bewertet. Genauer gesagt werden für die noch nicht ausgeführten Aktionen alternative Web Services selektiert, welche eine kürzere Ausführungszeit haben. Damit wird sichergestellt, dass nach Ausführung die vorher definierte Gesamtausführungszeit nicht überschritten wurde. (Bebner 2007, S. 111)

3.3.1.3 Gewährleistung des optimalen Prozesses hinsichtlich der Präferenzen des Nutzers

Durch den Monitoring-Mechanismus werden auch positive Laufzeitveränderungen erfasst. So kann der Fall eintreten, dass das tatsächliche Laufzeitverhalten der Web Services besser ist als vorher errechnet. Zwar werden in diesem Fall von der Servicekette weiterhin alle Restriktionen eingehalten, aber der Ausführungsplan ist nicht mehr unbedingt optimal hinsichtlich der Nutzerpräferenzen. Anhand des folgenden Beispiels wird versucht, diesen Sachverhalt zu verdeutlichen: Die Restriktion der Gesamtantwortzeit ist gegeben und darf nicht überschritten werden. Der Nutzer der Web Services zieht weiterhin eine möglichst kostengünstige Ausführung des Prozesses vor. Außerdem wird angenommen, dass Web Services mit einer höheren Ausführungszeit geringere Ausführungskosten verursachen. Da nun das Laufzeitverhalten der Web Services des ursprünglich berechneten Ausführungsplans besser ist als vorhergesagt, ist dieser nicht mehr optimal hinsichtlich der Nutzerpräferenzen. Es können im verbleibenden Teil der Aktionskette zu den einzelnen Aktionen Web Services mit einer höheren Ausfüh-

rungszeit, jedoch mit geringeren Ausführungskosten gewählt werden, ohne die Restriktionen des Nutzers zu verletzen. Durch Re-planning werden nun die noch nicht ausgeführten Web Services durch kostengünstigere ersetzt. So entsteht für den Nutzer durch das verbesserte Laufzeitverhalten eine Einsparung durch den Einsatz von kostengünstigeren Web Services und gleichzeitig wird die Restriktion der Gesamtausführungszeit nicht verletzt. Das Re-planning stellt somit stets sicher, dass hinsichtlich der Nutzerpräferenzen stets der optimale Prozess ausgeführt wird. (Berbner 2007, S. 111-112)

3.3.2 Re-planning Beispiel

Im Folgenden wird das Beispiel Kreditwürdigkeitsprüfung aus Kapitel 3.2.1 aufgegriffen und mit einem Re-planning weitergeführt. In dem Verfahrensbeispiel ist als optimale Servicekette eine Servicekette aus der Aktionskette [1-4-5-6-7] des kombinierten Verfahrens selektiert worden. Tritt nun der Fall auf, dass nach dem Aufruf des kombinierten Verfahrens der Web Service für die Aktion „Sourcing einholen“ ausfällt (siehe Abbildung 3-5), muss ein Re-planning durchgeführt werden. Bei einer lokalen Optimierungsstrategie würde es genügen, den nächstbesten Web Service der Aktion 5 aufzurufen, wenn dieser die Restriktionen des Nutzers nicht verletzt. Jedoch wenn wie in diesem Fall eine globale Optimierungsstrategie angewendet wird, muss hier die noch nicht ausgeführte Aktionskette neu berechnet werden. Es muss festgestellt werden, ob der Einsatz eines alternativen Web Services für Aktion 5 und demzufolge eine erneute Selektion des noch nicht ausgeführten Teils der Aktionskette gemacht wird oder ob gegebenenfalls der Start der alternativen Aktionskette (SCHUFA-Verfahren) einen größeren Vorteil bringt. Eine Überschneidung liegt durch die Aktion 1 „Kreditwürdigkeit prüfen“ vor, da diese Aktion aber schon ausgeführt wurde, ist sie nicht mehr relevant. Ebenso wenig relevant sind die bereits entstandenen Ausführungskosten, Ausführungszeiten und die Ausfallwahrscheinlichkeit der bereits ausgeführten Web Services für die Aktionen 1, 4 und 5. Durch eine Berechnung im Rahmen des Re-plannings wurde herausgefunden, dass durch den Aufruf des Web Services $S_{5,2}$ die Ausführung des kombinierten Verfahrens mit einem Präferenzfunktionswerts von 0,49 weiterhin vorteilhafter ist als die Ausführung der alternativen Aktionskette (Präferenzfunktionswert von 0,6 für SCHUFA-Verfahren). In Folge dessen ist es nur notwendig den Web Service $S_{5,2}$ anstelle des Web Service $S_{5,1}$ aufzurufen sowie die restlichen Web Services der Aktion „Bonität prüfen“ und der Aktion „Kreditwürdigkeit beurteilen“ auszuführen. Um die Werte der nicht-funktionalen Zielgrößen aktuell zu halten, wird der Ausfall des zuvor selektierten Web Services für Aktion 5 zugleich auch im Dienstverzeichnis erfasst, was zu einem neuen höheren Wert der Ausfallwahrscheinlichkeit dieses Web Services führt. (Heinrich et al. 2011b, S. 96)

Eine Übersicht über Web Service Selektionsverfahren, die Re-planning miteinbeziehen oder ansprechen, ist den Tabellen 3-2, 3-3, 3-4 und 3-5 im Kapitel 3.2.4 zu entnehmen.

4 Detailliertes Verfahrensbeispiel

In diesem Kapitel werden nochmals die von Yu et al. (2007) vorgestellten Selektionsverfahren aufgegriffen. Anhand selbst aufgestellter Angaben zu Ausführungszeit und Ausführungskosten sollen die Vorgehensweisen der vorgestellten Algorithmen durch deren Berechnung verständlich und nachvollziehbar gemacht werden.

Grundsätzlich gliedern Yu et al. (2007) die Selektion in zwei Kategorien, nämlich in die Selektion von Prozessen sequentieller Struktur und in die Selektion von Prozessen genereller Struktur. Prozesse genereller Struktur können im Gegensatz zur rein sequentiellen Struktur auch Schleifen, Auswahl oder Verzweigungen enthalten. Yu et al. (2007) stellen zu diesen Kategorien je zwei Selektionsmodelle vor, das Kombinatorische Modell und das Graphenmodell. Das Selektionsproblem wird im kombinatorischen Modell als ein „Multidimension Multichoice Knapsack Problem (MMKP)“ definiert. Das Graphenmodell definiert das Selektionsproblem als „Multiconstrained Optimal Path (MCOP) Problem“. Zu beiden Modellen werden von den Autoren exakte und heuristische Web Service Selektionsverfahren vorgestellt. (Yu et al. 2007, S. 2+7ff.)

Im Folgenden wird jeweils eine Web Service Selektion anhand der Algorithmen BBPL (exaktes Verfahren), WS_HEU und MCSP-K (beide heuristische Verfahren) aus der Arbeit von Yu et al. (2007) für eine Aktionskette sequentieller Struktur berechnet.

4.1 Allgemeine Angaben

Für die Berechnung der Algorithmen für Prozesse sequentieller Struktur werden die folgenden Angaben verwendet. Sie gelten für alle Algorithmen.

Die Abbildung 4-1 zeigt die Aktionskette des auszuführenden Prozesses.

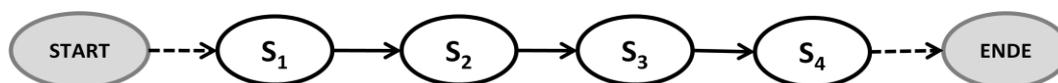


Bild 4-1 Aktionskette sequentieller Struktur

Zu jeder Aktion S_1 bis S_4 muss ein konkreter Web Service s_{ij} aus den jeweiligen Web Service Klassen S_i selektiert werden. Die konkreten Web Services sind inklusive der Werte zu ihren nicht-funktionalen Zielgrößen in Tabelle 4-1 aufgeführt.

Die globalen Restriktionen der zu selektierenden Servicekette sind:

Ausführungszeit ≤ 700 Zeiteinheiten

Ausführungskosten ≤ 350 Geldeinheiten

Serviceklasse	Web Service	Ausführungszeit	Ausführungskosten
S ₁	s _{1,1}	100	50
	s _{1,2}	190	40
	s _{1,3}	170	30
S ₂	s _{2,1}	200	50
	s _{2,2}	160	100
	s _{2,3}	180	80
S ₃	s _{3,1}	150	70
	s _{3,2}	120	110
S ₄	s _{4,1}	280	70
	s _{4,2}	200	90
	s _{4,3}	160	180

Tabelle 4-1 Übersicht der Web Services und der Werte der nicht-funktionalen Zielgrößen

Die Selektionsparameter und Notationen der Funktionen werden in der folgenden Tabelle 4-2 definiert.

Parameter	Definition
S _i	Serviceklasse i. Beinhaltet funktional gleiche Web Services mit unterschiedlichen nicht-funktionalen Zielgrößen.
s _{ij}	Abstrakter Web Service j der Serviceklasse i.
F	Nutzenfunktion eines jeden Servicekandidaten v, welche durch nicht-funktionale Zielgrößen definiert ist.
q _{ij}	Vektor der nicht-funktionalen Zielgrößen des abstrakten Web Services s _{ij} , q _{ij} = [q ¹ _{ij} , ..., q ⁿ _{ij}].
Q _c	Restriktionen der nicht-funktionalen Zielgrößen.

Tabelle 4-2 Parameter und Notationen (Yu et al. 2007, S. 6)

Die Aggregationsformeln der nicht-funktionalen Zielgrößen sind aus der Tabelle 3-1 zu entnehmen.

Anhand der Formel der Nutzenfunktion (1) (Yu et al. 2007, S. 6) wird der Servicenutzen F eines einzelnen Web Services berechnet. In der Tabelle 4-3 ist der berechnete Servicenutzen F für jeden Web Service der Serviceklassen eins bis vier aufgelistet.

$$F_{ij} = \sum_{\alpha=1}^x w_{\alpha} * \left(\frac{q_{ij}^{\alpha} - \mu^{\alpha}}{\sigma^{\alpha}} \right) + \sum_{\beta=1}^y w_{\beta} * \left(1 - \frac{q_{ij}^{\beta} - \mu^{\beta}}{\sigma^{\beta}} \right) \quad (1)$$

Mathematisch wird das Serviceselektionsproblem wie folgt formuliert:

$$\begin{aligned} & \text{Max} \sum_{i=1}^N \sum_{j \in S_i} F_{ij} x_{ij} \quad (2) \\ & \text{In Abhängigkeit von} \sum_{i=1}^N \sum_{j \in S_i} q_{ij} x_{ij} \leq Q_C^{\alpha} \quad (\alpha = 1, \dots, m) \\ & \sum_{j \in S_i} x_{ij} = 1 \\ & x_{ij} \in \{0,1\} \quad i = \dots, N, j \in S_i \end{aligned}$$

Wird ein konkreter Web j Service aus der Serviceklasse S_i gewählt, wird x_{ij} auf 1 gesetzt, falls nicht auf 0. Der Parameter $q_{ij} = [q_{ij}^1, \dots, q_{ij}^n]$ stellt die benötigten Ressourcen pro konkretem Web Service s_{ij} dar. (Yu et al. 2007, S. 8)

Die Tabelle 4-3 zeigt für den jeweiligen Web Service der Serviceklassen eins bis vier den Mittelwert, die Standardabweichung und den Servicenutzen. Der Servicenutzen (fett gedruckter Wert) wurde nach der Formel (1) berechnet.

Web Service	Mittelwert		Standardabweichung		Servicenutzen
	Ausführungszeit	Ausführungskosten	Ausführungszeit	Ausführungskosten	
s _{1,1}	153,3333	40,0000	38,5861	8,1650	0,4832
s _{1,2}	153,3333	40,0000	38,5861	8,1650	0,5124
s _{1,3}	153,3333	40,0000	38,5861	8,1650	1,2544
s _{2,1}	180,0000	76,6667	16,3299	20,5480	1,0927
s _{2,2}	180,0000	76,6667	16,3299	20,5480	0,4884
s _{2,3}	180,0000	76,6667	16,3299	20,5480	0,6689
s _{3,1}	135,0000	90,0000	15,0000	20,0000	1,0000
s _{3,2}	135,0000	90,0000	15,0000	20,0000	0,5000
s _{4,1}	213,3333	113,3333	49,8888	47,8423	0,8688
s _{4,2}	213,3333	113,3333	49,8888	47,8423	1,0607
s _{4,3}	213,3333	113,3333	49,8888	47,8423	0,3205

Tabelle 4-3 Berechnung Servicenutzen F

4.2 Berechnung exaktes Verfahren – BBLP Algorithmus

Zur Selektion einer optimalen Servicekette schlagen Yu et al. (2007) ein Branch and Bound Verfahren (BBLP Algorithmus) vor. Dieser findet durch die wiederholte Generierung eines Suchbaumes die optimale Lösung des Optimierungsproblems. In diesem Suchbaum werden aufeinanderfolgende Ebenen mit Knoten gebildet, welche je Ebene zusammen eine Serviceklasse darstellen. Zu jeder Serviceklasse wird ein Web Service selektiert, also der Knoten, welcher den größten Upper Bound und zugleich eine zulässige Lösung darstellt. Ausgehend von diesem Knoten wird der Suchbaum weitergeführt, indem neue Knoten der nächsten Serviceklasse generiert werden. (Yu et al. 2007, S. 8)

Die Abbildung 4-2 zeigt in Form des Suchbaumes die Berechnung des Optimierungsproblems durch das Branch and Bound Verfahren.

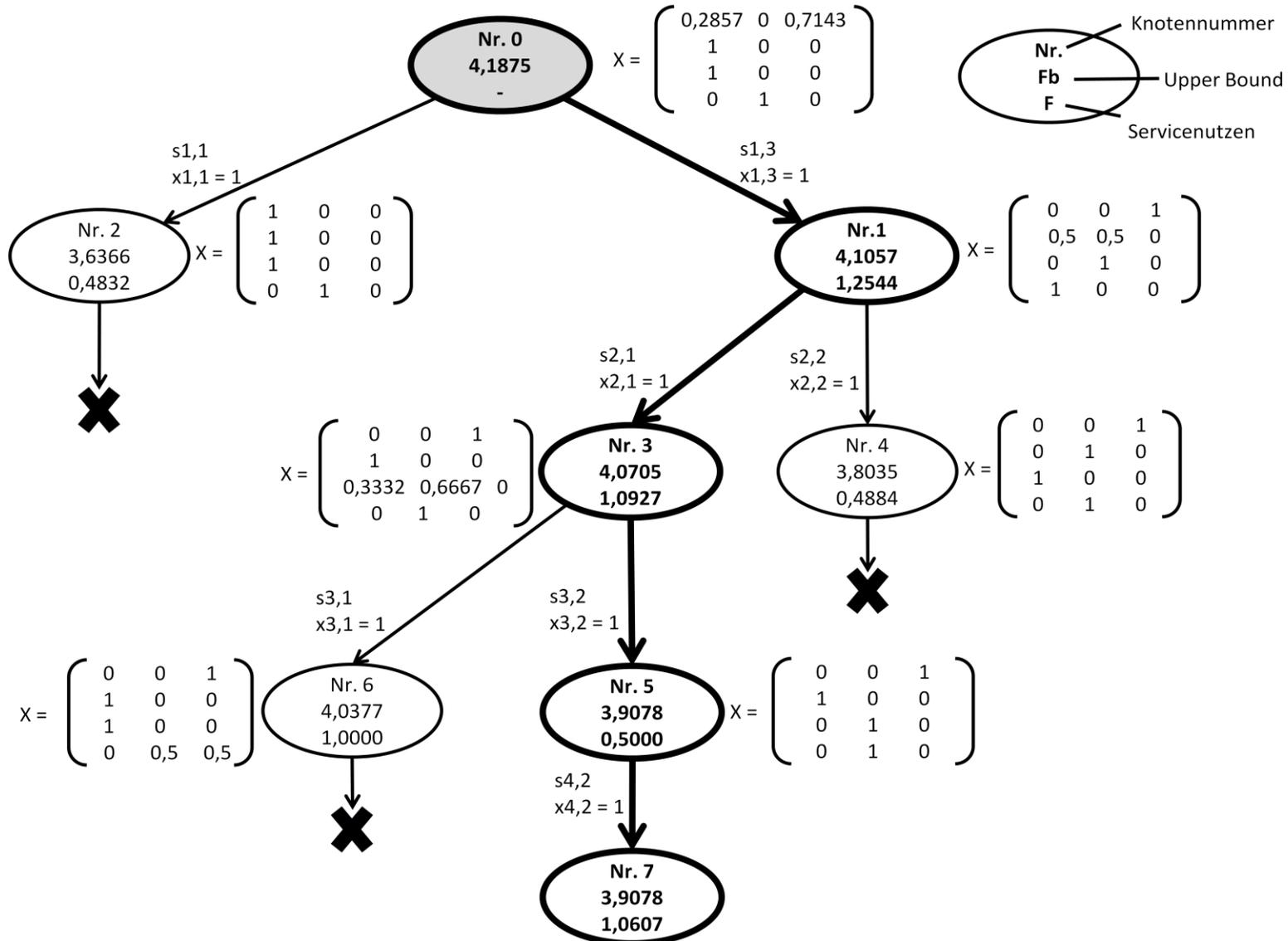


Bild 4-2 Berechnung BBLP Algorithmus

Aus dem Knoten Nr. 0 der Abbildung 4-2 ist zu entnehmen, dass der maximale Nutzen des relaxierten Problems 4,1875 ist. Dieser Wert stellt die oberste Schranke dar. Auf den jeweiligen Verbindungspfeilen sind die Web Service Kandidaten angegeben und ob dieser Pfad auf 1 oder 0 gesetzt wurde. Zur Verdeutlichung ist nicht nur jeweils ein Pfad auf 1 gesetzt, sondern alle möglichen Pfade, nach welchen verzweigt werden kann. In der vorliegenden Berechnung sind $x_{1,1}$ und $x_{1,3}$ auf 1 gesetzt, um zu zeigen, dass der Ast $x_{1,1}$ eine unzulässige Lösung darstellt. Aus den X-Matrizen der Knoten sind die relaxierten Werte der möglichen Äste zu entnehmen.

Die X-Matrix des Knoten 0 für den Ast $x_{1,1}$ zeigt einen x -Wert von 0,2857 und für den Ast $x_{1,3}$ einen x -Wert von 0,7143. Der Ast $x_{1,2}$ hat den x -Wert Null und wird daher nicht beachtet, also nicht verzweigt. Da der x -Wert des Astes $x_{1,3}$ den größeren Wert als der des Astes $x_{1,2}$ besitzt, d. h. den Wert, welcher näher bei 1 ist, wird die Berechnung zunächst immer von Web Service 1,3 weitergeführt. Zur Verdeutlichung, aber auch zur Überprüfung, ob nicht in diesem Ast eine bessere Lösung erzielt werden kann, ist auch der Upper Bound des Astes $x_{1,1}$ zum Knoten Nr. 2 dargestellt. Der Upper Bound des Knoten Nr. 2 ist aber mit dem Wert von 3,6366 geringer als die unterste Schranke mit dem Wert von 3,9078 im Ast $x_{3,2}$ und liefert zudem schon jetzt ausschließlich binäre x -Werte in der X-Matrix. Da dies eine unzulässige Lösung ist, wird der Ast hier nicht weiter betrachtet. Vom Knoten Nr. 1 wird nun weiter verzweigt. Die X-Matrix des Knoten Nr. 1 zeigt für die Äste $x_{2,1}$ und $x_{2,2}$ jeweils die Werte 0,5 an. Knoten Nr.4 im Ast $x_{2,2}$ liefert aber wieder nur binäre Werte, dieser Knoten wird damit ab hier nicht weiter verzweigt. Es wird nun Knoten Nr. 3 betrachtet. Die X-Matrix zeigt die Äste $x_{3,1}$ und $x_{3,2}$ als mögliche Verzweigungen an. Die X-Matrizen der nun generierten Knoten Nr. 5 und Nr. 6 dürfen nur noch binäre Werte enthalten, da sich die Servicekette hier am Ende befindet und eine zulässige Lösung erzeugt werden muss. Die X-Matrix des Knoten Nr. 6 zeigt nicht ausschließlich binäre Werte an. Da dies damit keine zulässige Lösung darstellt, wird der Knoten hier nicht mehr weiter verzweigt. Die X-Matrix des Knoten Nr. 5 enthält nur binäre Werte und zeigt, dass nur noch der Ast $x_{4,2}$ verzweigt werden kann. Somit wird im Ast $x_{4,2}$ der letzte Knoten Nr. 7 erzeugt. Der Suchbaum ist damit am Ende, es wurde eine zulässige Lösung gefunden.

Der fett gedruckte Ast in der Abbildung 4-2 stellt die durch den BBLP Algorithmus selektierte optimale Servicekette dar. Sie lautet:

$$\text{Optimale Servicekette} = s_{13} + s_{21} + s_{32} + s_{42}$$

Prüfung der Restriktionen:

Ausführungszeit prüfen: $Q = 170 + 200 + 120 + 200 = 690 \rightarrow Q \leq Q_c$

Ausführungskosten prüfen: $Q = 30 + 50 + 110 + 90 = 280 \rightarrow Q \leq Q_c$

Die Ausführungszeit der optimalen Servicekette ist 690 Zeiteinheiten (Restriktion 700 Zeiteinheiten) und die Ausführungskosten belaufen sich auf 280 Geldeinheiten (Restriktion 350 Geldeinheiten). Gleich dem Upper Bound in Ast $x_{4,2}$ ist der Nutzen der einzeln addierten Web Services der Servicekette 3,9078. Der jeweilige Upper Bound der Knoten wurde mit dem Simplex Verfahren unter Zuhilfenahme der Optimierungssoftware IBM ILOG CPLEX Optimization Studio Version 12.3 berechnet.

4.3 Berechnung heuristisches Verfahren – WS_HEU Algorithmus

Da das exakte Verfahren BBPL durch die Verwendung des Branch and Bound Verfahrens ein sehr zeitaufwändiges Optimierungsverfahren ist, wird von Yu et al. (2007) die Heuristik WS_HEU vorgeschlagen. Eine Heuristik ist in der Lage eine zulässige Lösung mit weniger Rechenaufwand zu finden und ermöglicht damit eine zeitlich effizientere Web Service Selektion. (Yu et al. 2007, S. 9)

4.3.1 Angaben zu WS_HEU

Das Verfahren WS_HEU ist in drei Schritte gegliedert. Im ersten Schritt wird eine zulässige Lösung des Problems gesucht, indem zunächst Web Services mit möglichst wenigen Ressourcen selektiert werden. Ist im ersten Schritt eine nicht zulässige Lösung gefunden, wird versucht die Lösung aufzuwerten (Funktion F2), indem Web Services mit einem noch geringeren Ressourcenverbrauch gesucht werden. Ist nun eine zulässige Lösung gefunden, wird durch die Funktion F3 versucht den Ressourcenverbrauch weiter zu senken, bis keine weitere Web Services, welche den Ressourcenverbrauch noch weiter senken können, vorhanden sind. Wenn keine Web Services mehr vorhanden sind welche den Ressourcenverbrauch senken, wird nun mit der Funktion F4 versucht möglichst viel Nutzen aus mehr verbrauchten Ressourcen zu ziehen. Ist ein solcher Web Service gefunden worden, wird durch die Funktion F3 erneut versucht den Ressourcenverbrauch zu senken. Werden keine zulässigen Aufwertungen mehr gefunden, wird das aktuelle Ergebnis gespeichert. In Schritt 1 und 2 werden die Funktionen immer nur mit einem Web Service durchgeführt. Mit Schritt 3 wird durch eine unzulässige Aufwertung des Nutzens und einer anschließenden Abwertung versucht das Ergebnis mittels zwei verschiedener Web Services zu verbessern.

Es wird jetzt durch eine nicht zulässige Aufwertung (Funktion F5) des bisherigen Ergebnisses versucht die Schranken der Restriktionen zu durchbrechen. Damit die Schranken aber auch wieder eingehalten werden, wird durch eine anschließende Abwertung (Funktion F6) versucht die Schranken wieder einzuhalten. Ist dies möglich, wird das Ergebnis dieses Schrittes mit dem bis dahin gespeicherten Ergebnis verglichen. Wurde eine Verbesserung erzielt, wird nun wieder zu Funktion F3 zurückgegangen und versucht den Ressourcenverbrauch zu senken. Anderenfalls, wenn keine Verbesserung erfolgt oder es keine mögliche nicht zulässige Aufwertung und folgende Abwertung gibt, ist das bis dahin gespeicherte Ergebnis das Endergebnis.

Die folgenden Abbildungen 4-3 und 4-4 zeigen den graphischen Ablauf des Verfahrens und die mathematischen Definitionen der Ablaufschritte des Verfahren WS_HEU.

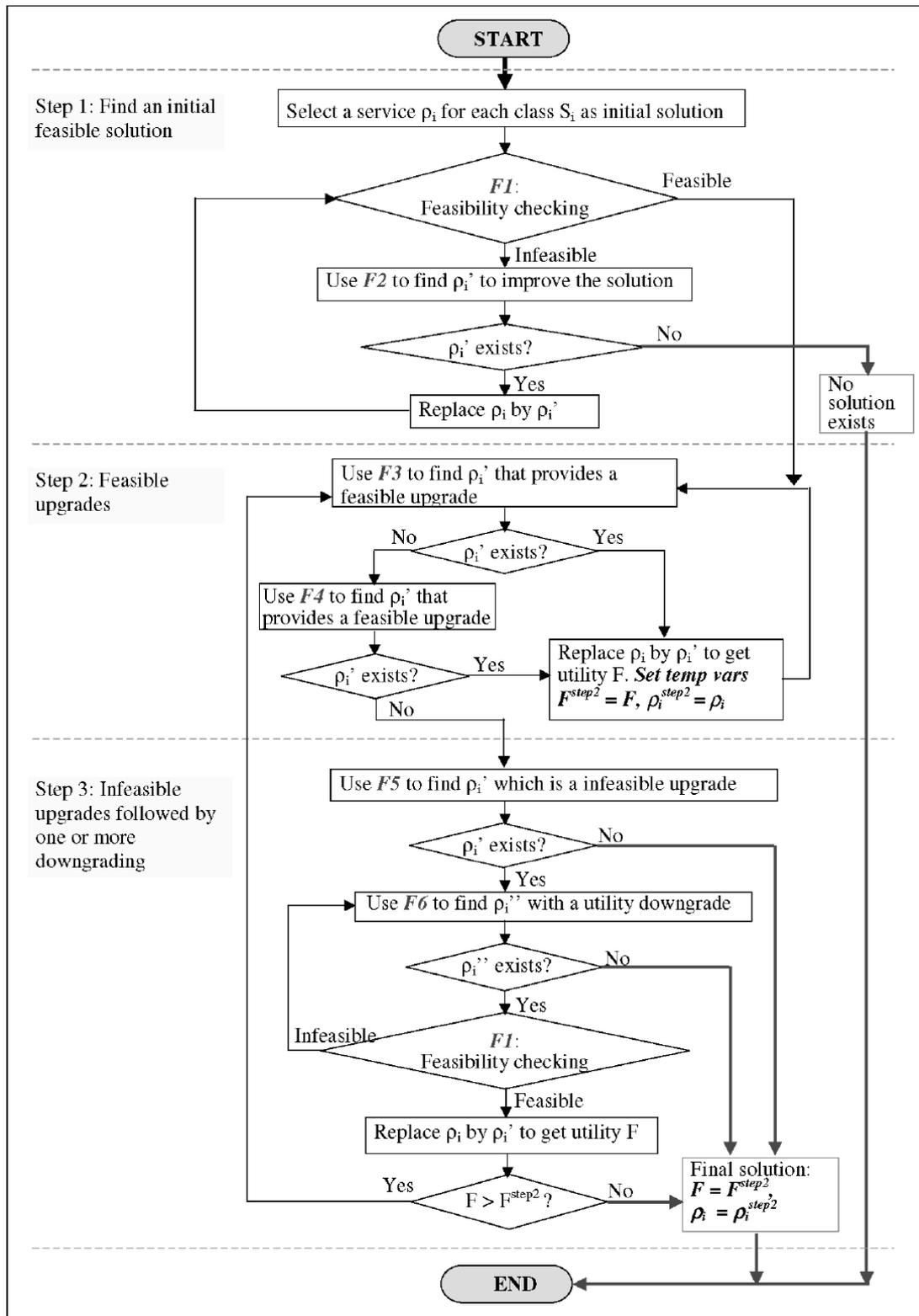


Bild 4-3 Ablauf WS_HEU (Yu et al. 2007, S. 10)

Function	Definition
Notations	$\mathbf{Q} = \sum_{i=1}^N \mathbf{q}_{i\rho_i}$: Current aggregated QoS value; ($\mathbf{Q} = [Q^1, \dots, Q^m]$) $f^\alpha = \frac{Q^\alpha}{Q_c^\alpha}$: feasibility factor of QoS attribute α . ≤ 1 : feasible; > 1 , infeasible; $\Delta a_{ij} = (\mathbf{q}_{i\rho_i} - \mathbf{q}_{ij}) \times \mathbf{Q} / \mathbf{Q} $: saving in aggregated QoS ; $\Delta p_{ij} = (\mathcal{F}_{i\rho_i} - \mathcal{F}_{ij}) / \Delta a_{ij}$: value gain per unit of extra QoS; $\Delta t'_{ij} = (\mathbf{q}_{i\rho_i} - \mathbf{q}_{ij}) / (\mathbf{Q}_c - \mathbf{Q}) = \sum_{\alpha=1}^m (q_{i\rho_i}^\alpha - q_{ij}^\alpha) / (Q_c^\alpha - Q^\alpha)$: savings in total-per-unit-QoS ; $\Delta p'_{ij} = (\mathcal{F}_{i\rho_i} - \mathcal{F}_{ij}) / \Delta t'_{ij}$: value gain per unit of extra total-per-unit-QoS; $\Delta t''_{ij} = (\mathbf{q}_{i\rho_i} - \mathbf{q}_{ij}) / (\mathbf{Q} - \mathbf{Q}_c)$: losses in total-per-unit-QoS; $\Delta p''_{ij} = \Delta t''_{ij} / (\mathcal{F}_{i\rho_i} - \mathcal{F}_{ij})$: value loss per unit of extra total-per-unit-QoS;
F1	$\forall \alpha, Q^\alpha \leq Q_c^\alpha$
F2	$\rho'_i = \max_{i,j} \{\Delta a_{ij}\}$ and satisfy: (1) $f^\beta = \max_\alpha \{f^\alpha\}$, $f_{new}^\beta < f_{old}^\beta$; (2) $f_{new}^\alpha \leq f_{old}^\alpha$, if $f_{old}^\alpha > 1$ and $\alpha \neq \beta$; (3) $f_{new}^\alpha \leq 1$, if $f_{old}^\alpha \leq 1$ and $\alpha \neq \beta$; f_{old} / f_{new} : value before / after service replacement;
F3	$\rho'_i = \max_{i,j} \{\Delta a_{ij}\}$, $\Delta a_{ij} > 0$
F4	$\rho'_i = \max_{i,j} \{\Delta p_{ij}\}$, $\Delta a_{ij} < 0$
F5	$\rho'_i = \max_{i,j} \{\Delta p'_{ij}\}$
F6	$\rho''_i = \min_{i,j} \{\Delta p''_{ij}\}$

Bild 4-4 Definitionen des Ablauf WS_HEU (Yu et al. 2007, S. 11)

4.3.2 Berechnung WS_HEU

Schritt 1:

Man finde eine erste zulässige Lösung, indem aus jeder Serviceklasse S_i der Web Service $\min_j \{ \max_\alpha \{ q_{ij}^\alpha / Q_c^\alpha \} \}$ gewählt wird. Falls keine zulässige Lösung sofort gefunden wurde, wird versucht die unzulässige Lösung zu verbessern.

Aus jeder Serviceklasse der Tabelle 4-4 wird zunächst der Wert mit dem höchsten Wert ρ_i betrachtet. Anschließend wird in der entsprechenden Kategorie der nicht-funktionalen Zielgröße, in welcher der höchste Wert gefunden wurde, der kleinste Wert gesucht. Dieser kleinste Wert ist nun Bestandteil des Web Services der ersten Lösung. Konkret wird in dieser Berechnung aus der Serviceklasse 1 zuerst der höchste Wert betrachtet, welcher der Wert 0,24 in der Kategorie der Ausführungszeit ist. Nun wird aus dieser Kategorie Ausführungszeit der kleinste Wert 0,14 herangezogen, der dem Web Service $s_{1,1}$ entspricht. Dieser Web Service stellt jetzt eine erste Lösung dar. Zweck dieser ersten Selektion ist, eine Lösung mit möglichst wenig Ressourcenverbrauch zu finden. Die selektierten Web Services dieses Schrittes sind in der Tabelle 4-4 fett gedruckt.

Serviceklasse	Web Service	Ausführungskosten	Ausführungszeit
S ₁	s _{1,1}	$\rho_i = \frac{100}{700} = 0,14$	$\rho_i = \frac{50}{350} = 0,14$
	s _{1,2}	$\rho_i = \frac{190}{700} = 0,27$	$\rho_i = \frac{40}{350} = 0,11$
	s _{1,3}	$\rho_i = \frac{170}{700} = 0,24$	$\rho_i = \frac{30}{350} = 0,09$
S ₂	s _{2,1}	$\rho_i = \frac{200}{700} = 0,29$	$\rho_i = \frac{50}{350} = 0,14$
	s _{2,2}	$\rho_i = \frac{160}{700} = 0,23$	$\rho_i = \frac{100}{350} = 0,29$
	s _{2,3}	$\rho_i = \frac{180}{700} = 0,26$	$\rho_i = \frac{80}{350} = 0,23$
S ₃	s _{3,1}	$\rho_i = \frac{150}{700} = 0,21$	$\rho_i = \frac{70}{350} = 0,20$
	s _{3,2}	$\rho_i = \frac{120}{700} = 0,17$	$\rho_i = \frac{110}{350} = 0,31$
S ₄	s _{4,1}	$\rho_i = \frac{280}{700} = 0,40$	$\rho_i = \frac{70}{350} = 0,20$
	s _{4,2}	$\rho_i = \frac{200}{700} = 0,29$	$\rho_i = \frac{90}{350} = 0,26$
	s _{4,3}	$\rho_i = \frac{160}{700} = 0,23$	$\rho_i = \frac{180}{350} = 0,51$

Tabella 4-4 Berechnung der Werte für erste zulässige Lösung

Funktion F1: Zulässigkeit der ersten Lösung prüfen.

Servicekette = s₁₁ + s₂₂ + s₃₁ + s₄₁

Ausführungszeit prüfen: $Q = 100 + 160 + 150 + 280 = 690 \rightarrow Q \leq Q_c$

Ausführungskosten prüfen: $Q = 50 + 100 + 70 + 70 = 290 \rightarrow Q \leq Q_c$

Das bedeutet, dass die erste gefundene Lösung zulässig ist. Dies deckt sich auch mit den Ergebnissen der Messungen der Autoren Yu et al. (2007, S. 19). Dabei wurden bei 98% der untersuchten Fälle im ersten Versuch eine zulässige Lösung gefunden.

Schritt 2

Funktion F3: Man finde einen Web Service ρ'_i , welcher eine zulässige Aufwertung ermöglicht. Mittels Funktion F3 wird versucht einen Web Service zu finden, welcher den Ressourcenverbrauch der Servicekette senkt.

$$Q = \begin{pmatrix} 100 \\ 50 \end{pmatrix} + \begin{pmatrix} 160 \\ 100 \end{pmatrix} + \begin{pmatrix} 150 \\ 70 \end{pmatrix} + \begin{pmatrix} 280 \\ 70 \end{pmatrix} = \begin{pmatrix} 690 \\ 290 \end{pmatrix}$$

$$|Q| = \sqrt{690^2 + 290^2} = 748,47$$

$$\Delta\alpha_{12} = \left[\begin{pmatrix} 100 \\ 50 \end{pmatrix} - \begin{pmatrix} 190 \\ 40 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = -44,09$$

$$\Delta\alpha_{13} = \left[\begin{pmatrix} 100 \\ 50 \end{pmatrix} - \begin{pmatrix} 170 \\ 30 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = -45,55$$

$$\Delta\alpha_{21} = \left[\begin{pmatrix} 160 \\ 100 \end{pmatrix} - \begin{pmatrix} 200 \\ 50 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = -61,59$$

$$\Delta\alpha_{23} = \left[\begin{pmatrix} 160 \\ 100 \end{pmatrix} - \begin{pmatrix} 180 \\ 80 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = -26,19$$

$$\Delta\alpha_{32} = \left[\begin{pmatrix} 150 \\ 70 \end{pmatrix} - \begin{pmatrix} 120 \\ 110 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = 48,50$$

$$\Delta\alpha_{42} = \left[\begin{pmatrix} 280 \\ 70 \end{pmatrix} - \begin{pmatrix} 200 \\ 90 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = 49,43$$

$$\Delta\alpha_{43} = \left[\begin{pmatrix} 280 \\ 70 \end{pmatrix} - \begin{pmatrix} 160 \\ 180 \end{pmatrix} \right] \times \begin{pmatrix} 690 \\ 290 \end{pmatrix} \div 748,47 = 147,90$$

Betrachtet wird nur der maximale Wert von $\Delta\alpha$, welcher größer als Null ist. Das ist der Wert $\Delta\alpha_{43}$. Der Web Service s_{41} wird nun gegen $\rho'_i = s_{43}$ ausgetauscht. Anschließend wird die Zulässigkeit geprüft.

Servicekette = $s_{11} + s_{22} + s_{31} + s_{43}$

Ausführungszeit prüfen: $Q = 100 + 160 + 150 + 160 = 570 \rightarrow Q \leq Q_c$

Ausführungskosten prüfen: $Q = 50 + 100 + 70 + 180 = 400 \rightarrow Q \geq Q_c \rightarrow$ Nicht zulässig!

Da diese Aufwertung nicht zulässig ist, d. h. kein Senken der Ressourcen möglich ist, verbleibt der Web Service s_{41} in der Servicekette. Die aktuelle Servicekette lautet weiterhin:

Servicekette = $s_{11} + s_{22} + s_{31} + s_{41}$

Es wird nun zu Funktion F4 weitergegangen.

Funktion F4: Finde eine Lösung, die eine zulässige Aufwertung ermöglicht. Es wird versucht einen Web Service ρ'_i zu finden, welcher den Nutzen der bisherigen Lösung erhöht. Bei der Berechnung werden nur Web Services betrachtet, deren bei Funktion F3 berechneter $\Delta\alpha$ Wert kleiner als Null ist.

$$\Delta p_{12} = (0,4832 - 0,5124) \div (-44,09) = 0,000662$$

$$\Delta p_{13} = (0,4832 - 1,2544) \div (-45,55) = 0,016931$$

$$\Delta p_{21} = (0,4884 - 1,0927) \div (-61,59) = 0,009812$$

$$\Delta p_{23} = (0,4884 - 0,6689) \div (-26,19) = 0,006892$$

Es wird nun der Web Service mit dem maximalen positiven Wert $\Delta\rho_{ij}$ betrachtet und gegen den bisherigen Web Service der entsprechenden Serviceklasse getauscht. Dies ist der Wert $\Delta\rho_{13}$, welcher dem Web Service s_{13} entspricht und den Web Service s_{11} nun ersetzt.

$$\text{Servicekette} = s_{13} + s_{22} + s_{31} + s_{41}$$

$$\text{Ausführungszeit prüfen: } Q = 170 + 160 + 150 + 280 = 760 \rightarrow Q \geq Q_c \rightarrow \text{Nicht zulässig!}$$

$$\text{Ausführungskosten prüfen: } Q = 30 + 100 + 70 + 70 = 270 \rightarrow Q \leq Q_c$$

Dies ist eine nicht zulässige Lösung. Daher verbleibt der bisherige Web Service s_{11} in der Servicekette. Die aktuelle Servicekette lautet:

$$\text{Servicekette} = s_{11} + s_{22} + s_{31} + s_{41}$$

Schritt 3

In Schritt 3 wird nun durch eine unzulässige Aufwertung und anschließende Abwertung durch einen anderen Web Service versucht den Nutzen der bisherigen Lösung zu erhöhen.

Funktion F5: Für die unzulässige Aufwertung wird der maximale Wert $\Delta p'_{ij}$ herangezogen. Zur Berechnung von $\Delta p'_{ij}$ muss zunächst $\Delta t'_{ij}$ berechnet werden.

$$\Delta t'_{12} = (100 - 190) \div (700 - 690) + (50 - 40) \div (350 - 290) = -8,3334$$

$$\Delta t'_{13} = (100 - 170) \div (700 - 690) + (50 - 30) \div (350 - 290) = -6,6667$$

$$\Delta t'_{21} = (160 - 200) \div (700 - 690) + (100 - 50) \div (350 - 290) = -3,1667$$

$$\Delta t'_{23} = (160 - 180) \div (700 - 690) + (100 - 80) \div (350 - 290) = 1,6667$$

$$\Delta t'_{32} = (150 - 120) \div (700 - 690) + (70 - 110) \div (350 - 290) = 2,3334$$

$$\Delta t'_{42} = (280 - 200) \div (700 - 690) + (70 - 90) \div (350 - 290) = 7,6667$$

$$\Delta t'_{43} = (280 - 160) \div (700 - 690) + (70 - 180) \div (350 - 290) = 10,1667$$

Berechnung $\Delta p'_{ij}$:

$$\Delta p'_{12} = (0,4832 - 0,5124) \div (-8,3334) = 0,0035$$

$$\Delta p'_{13} = (0,4832 - 1,2544) \div (-6,6667) = 0,1157$$

$$\Delta p'_{21} = (0,4884 - 1,0927) \div (-3,1667) = 0,1908$$

$$\Delta p'_{23} = (0,4884 - 0,6689) \div (1,6667) = -0,1083$$

$$\Delta p'_{32} = (1,000 - 0,5000) \div (2,3334) = 0,2143$$

$$\Delta p'_{42} = (0,8688 - 1,0607) \div (7,6667) = -0,0250$$

$$\Delta p'_{43} = (0,8688 - 0,3205) \div (10,1667) = 0,0539$$

Es wird nun derjenige Web Service mit dem maximalen positiven Wert $\Delta p'_{ij}$ betrachtet und gegen den bisherigen Web Service der entsprechenden Serviceklasse getauscht. Dies ist der Wert $\Delta p'_{32}$, welcher dem Web Service s_{32} entspricht und den Web Service s_{31} nun ersetzt.

$$\text{Servicekette} = s_{11} + s_{22} + s_{32} + s_{41}$$

$$\text{Ausführungszeit prüfen: } Q = 100 + 160 + 120 + 280 = 660 \rightarrow Q \leq Q_c$$

$$\text{Ausführungskosten prüfen: } Q = 50 + 100 + 110 + 70 = 330 \rightarrow Q \leq Q_c$$

Dies ist eine zulässige Lösung. Daher verbleibt der bisherige Web Service s_{31} in der Servicekette. Die aktuelle Servicekette lautet:

$$\text{Servicekette} = s_{11} + s_{22} + s_{31} + s_{41}$$

Es wird nun nicht mehr zu Funktion F6 weitergegangen, da keine nicht zulässige Aufwertung durch den maximalen Wert $\Delta p'_{ij}$ erzielt wurde. Eine zulässige Lösung, wie sie hier durch die Aufwertung durch den maximalen Wert $\Delta p'_{32}$ generiert wurde, kann unter Umständen nur ein lokales Optimum schaffen. Daher stellt die Lösung nach Schritt 2 die bestmögliche gefundene Lösung dar. Die selektierte Servicekette des WS_HEU Algorithmus lautet:

$$\text{Finale Servicekette} = s_{11} + s_{22} + s_{31} + s_{41}$$

Die Ausführungszeit der finalen Servicekette benötigt 690 Zeiteinheiten (Restriktion 700 Zeiteinheiten) und die Ausführungskosten belaufen sich auf 290 Geldeinheiten (Restriktion 350 Geldeinheiten). Der Nutzen der einzeln addierten Web Services der Servicekette ist 2,8404. Zum Vergleich: Die optimal Lösung, welche durch den BBLP Algorithmus gefunden wurde, lautet: $s_{13} + s_{21} + s_{32} + s_{42}$. Sie hat eine Ausführungszeit von 690 Zeiteinheiten und Ausführungskosten von 280 Geldeinheiten. Der Nutzen der optimalen Lösung ist 3,9078. Die vom heuristischen Verfahren WS_HEU gefundene Servicekette hat einen um 10 Geldeinheiten höheren Ressourcenverbrauch, ist damit zwar eine zulässige jedoch nicht die optimale Lösung.

4.4 Berechnung heuristisches Verfahren – MCSP-K Algorithmus

MCSP ist der Algorithmus für ein exaktes Web Service Selektionsverfahren, welches ein graphisches Modell darstellt. Indem der Algorithmus MCSP durch den Algorithmus MCSP_RELAX erweitert wird, ergibt sich ein heuristisches Selektionsverfahren, der MCSP-K Algorithmus. Das Web Service Selektionsproblem wird gelöst, indem ein Pfad mit einem möglichst hohen Nutzen unter Beachtung der Nutzerpräferenzen gesucht wird. (Yu et al. 2007, S. 12)

4.4.1 Angaben zu MCSP-K

Die folgenden zwei Abbildungen 4-5 und 4-6 zeigen den Algorithmus MCSP und die Erweiterung des Algorithmus zum heuristischen Verfahren.

Algorithm MCSP ($G = (V, E)$, v_s , v_d , \mathbf{Q}_c)

//Definitions used in the algorithm:
 v_s, v_d : source and target nodes in service candidate graph;
 $\mathbf{q}(\mu, v)$: QoS of edge (μ, v) ; $\mathbf{q}(\mu, v) = [q^1(\mu, v), \dots, q^m(\mu, v)]$
 \mathcal{F} : utility of edge (μ, v) or path p ;
 $\mathbf{Q}(p)$: aggregated QoS of path p ;
 $\mathbf{P}(v)$: list of paths from v_s to node v that satisfy QoS constraints.

```

1  Topologically sort nodes in  $G$ ;
2  For each node  $\mu$ , taken in topological order
3    For each  $v \in adj[\mu]$ 
4      If  $(\mu = v_s)$  then
5         $\mathbf{Q} \leftarrow \mathbf{q}(\mu, v)$ 
6         $\mathcal{F} \leftarrow \mathcal{F}(\mu, v)$ 
7        MCSP_RELAX( $\mu, v, \mathbf{Q}, \mathcal{F}$ ) 8      Else for each  $p \in \mathbf{P}(\mu)$ 
9         $\mathbf{Q} \leftarrow \mathbf{Q}(p) + \mathbf{q}(\mu, v)$ 
10        $\mathcal{F} \leftarrow \mathcal{F}(p) + \mathcal{F}(\mu, v)$ 
11       MCSP_RELAX( $\mu, v, \mathbf{Q}, \mathcal{F}$ )
12      End If
13    End For
14  End For

15  Find  $p^*$ , s.t.  $\forall p \in \mathbf{P}(v_d), \mathcal{F}(p^*) \geq \mathcal{F}(p)$ 

```

Bild 4-5 Algorithmus MCSP (Yu et al. 2007, S. 12)

```

Procedure MCSP_RELAX ( $\mu, v, \mathbf{Q}, \mathcal{F}$ )
1  If ( $\exists \alpha, Q^\alpha > Q_c^\alpha$ ) then
2    return
3  End If
4  For each  $p \in \mathbf{P}(v)$ 
5    If  $\mathcal{F}(p) > \mathcal{F}$  and  $\forall \alpha, Q^\alpha(p) \leq Q^\alpha$  then return
6    End If
7    If  $\mathcal{F}(p) < \mathcal{F}$  and  $\forall \alpha, Q^\alpha(p) \geq Q^\alpha$  then
8      remove  $p$  from  $\mathbf{P}(v)$ 
9    End If
10 End For
11 Add path  $p'$  with attributes  $[\mu, \mathbf{Q}, \mathcal{F}]$  to  $\mathbf{P}(v)$ 

```

Bild 4-6 Algorithmus MCSP-K

Durch die topologische Sortierung aller Knoten (Web Service Kandidaten) muss zunächst ein Servicekandidatengraph erstellt werden. Dieser Servicekandidatengraph beinhaltet auch einen Startknoten v_s und einen Endknoten v_d . Jeder Knoten hat einen Nutzen und nicht-funktionale Zielgrößen. (Yu et al. 2007, S. 10-11) Der entsprechende Graph ist Abbildung 4-7 dargestellt.

Jeder Knoten wird nun nacheinander abgearbeitet. In jedem Knoten werden der bis dahin aggregierte Nutzen und die aggregierten Werte der nicht-funktionalen Zielgrößen der bis dahin durchlaufenen Pfade gespeichert. Jedoch werden die gespeicherten Pfade, welche in einem Knoten von einem anderen Pfad dominiert werden, wieder gelöscht. Das heuristische Verfahren macht die Selektion zeitlich nochmals effizienter, indem in jedem Knoten nur eine Anzahl von K Pfaden gespeichert wird. D. h. wenn die Anzahl von K Pfaden erreicht ist, werden die Pfade gestrichen, welche bis dahin den größten Ressourcenverbrauch aufweisen, sprich die Pfade mit größten g_λ werden nicht mehr weiter beachtet. Die entsprechende Kostenfunktion (1) definieren Yu et al. (2007, S. 12) folgend:

$$g_\lambda(p) \triangleq \left(\frac{q^1(p)}{Q^1}\right)^\lambda + \left(\frac{q^2(p)}{Q^2}\right)^\lambda + \dots + \left(\frac{q^m(p)}{Q^m}\right)^\lambda \quad (1)$$

Die Anzahl der Pfade K , welche in jedem Knoten gespeichert werden, ist in der folgenden Berechnung des Algorithmus 1, d. h. K wird auf 1 gesetzt, die restlichen Pfade pro Knoten werden nicht weiter geführt.

4.4.2 Berechnung MCSP-K

Die Abbildung 4-7 zeigt die topologische Sortierung der Knoten von Knoten v_1 bis v_{11} , welchen den einzelnen Web Services $s_{1,1}$ bis $s_{4,3}$ aus der Tabelle 4-1 entsprechen.

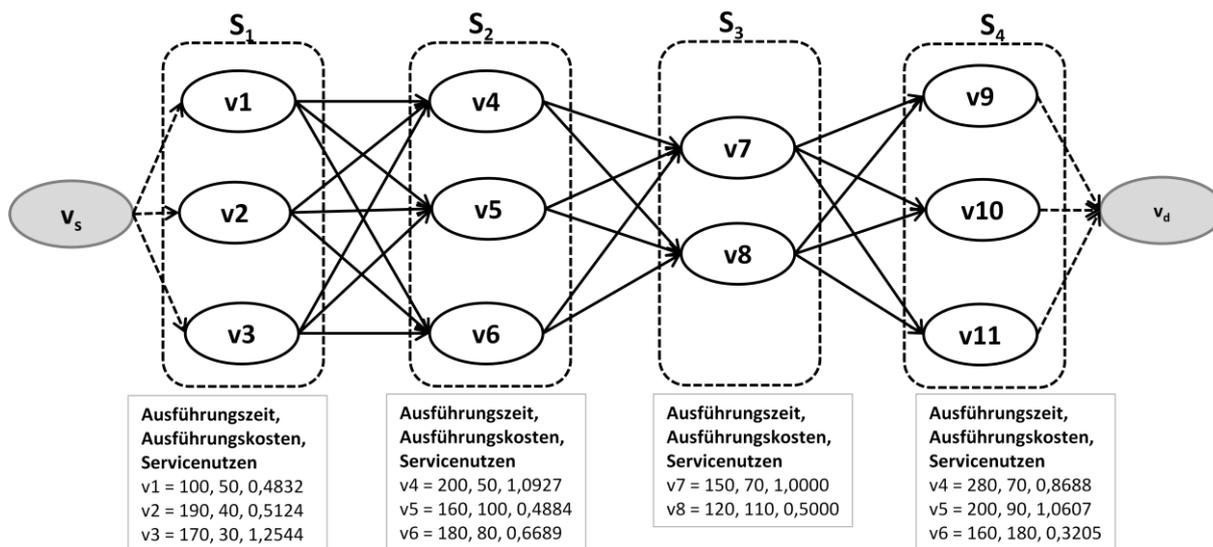


Bild 4-7 Topologische Sortierung der Knoten

Im Folgenden werden alle Knoten und ihre Nachfolger einzeln betrachtet.

1. Knoten $\mu = v_s$:

Nachfolger $v = v_1$:

$$q_{v1} = (100, 50)$$

$$Q_{v1} = (100, 50)$$

$$F_{v_s,v1} = 0,4832$$

$$P_{v_s,v1} = v_s + v_1$$

Nachfolger $v = v_2$:

$$q_{v2} = (190, 40)$$

$$Q_{v2} = (190, 40)$$

$$F_{v_s,v2} = 0,5124$$

$$P_{v_s,v2} = v_s + v_2$$

Nachfolger $v = v_3$:

$$q_{v3} = (170, 30)$$

$$Q_{v3} = (170, 30)$$

$$F_{v_s,v3} = 1,2544$$

$$P_{vs,v3} = v_s + v_3$$

2. Knoten $\mu = v_1$:

Nachfolgeknoten $v = v_4$:

$$Q_{v1,v4} = Q_{v1} + q_{v1,v4} = 100,50 + 200,50 = 300,100$$

$$F_{v1,v4} = 0,4832 + 1,0927 = 1,5759$$

$$P_{v1,v4} = v_s + v_1 + v_4$$

Nachfolgeknoten $v = v_5$:

$$Q_{v1,v5} = Q_{v1} + q_{v1,v5} = 100,50 + 160,100 = 260,150$$

$$F_{v1,v5} = 0,4832 + 0,4884 = 0,9716$$

$$P_{v1,v4} = v_s + v_1 + v_5$$

Nachfolgeknoten $v = v_6$:

$$Q_{v1,v6} = Q_{v1} + q_{v1,v6} = 100,50 + 180,80 = 280,130$$

$$F_{v1,v6} = 0,4832 + 0,6689 = 1,1521$$

$$P_{v1,v6} = v_s + v_1 + v_6$$

3. Knoten $\mu = v_2$:

Nachfolgeknoten $v = v_4$:

$$Q_{v2,v4} = Q_{v2} + q_{v2,v4} = 190,40 + 200,50 = 390,90$$

$$F_{v2,v4} = 0,5124 + 1,0927 = 1,6051$$

$$P_{v2,v4} = v_s + v_2 + v_4$$

→ Dieser Pfad wird von Pfad $P_{vs,v3,v4}$ dominiert, aus diesem Grund wird der Pfad hier nicht mehr weitergeführt.

Nachfolgeknoten $v = v_5$:

$$Q_{v2,v5} = Q_{v2} + q_{v2,v5} = 190,40 + 160,100 = 350,140$$

$$F_{v2,v5} = 0,5124 + 0,4884 = 1,0008$$

$$P_{v2,v5} = v_s + v_2 + v_5$$

- Dieser Pfad wird von Pfad P_{v_s, v_3, v_5} dominiert, aus diesem Grund wird der Pfad hier nicht mehr weitergeführt.

Nachfolgeknoten $v = v_6$:

$$Q_{v_2, v_6} = Q_{v_2} + q_{v_2, v_6} = 190,40 + 180,80 = 370,120$$

$$F_{v_2, v_6} = 0,5124 + 0,6689 = 1,1813$$

$$P_{v_2, v_6} = v_s + v_2 + v_6$$

4. Knoten $\mu = v_3$:

Nachfolgeknoten $v = v_4$:

$$Q_{v_3, v_4} = Q_{v_3} + q_{v_3, v_4} = 170,30 + 200,50 = 370,80$$

$$F_{v_3, v_4} = 1,2544 + 1,0927 = 2,3471$$

$$P_{v_3, v_4} = v_s + v_3 + v_4$$

Nachfolgeknoten $v = v_5$:

$$Q_{v_3, v_5} = Q_{v_3} + q_{v_3, v_5} = 170,30 + 160,100 = 330,130$$

$$F_{v_3, v_5} = 1,2544 + 0,4884 = 1,7428$$

$$P_{v_3, v_5} = v_s + v_3 + v_5$$

- Dieser Pfad wird von Pfad P_{v_s, v_3, v_6} dominiert, aus diesem Grund wird der Pfad hier nicht mehr weitergeführt.

Nachfolgeknoten $v = v_6$:

$$Q_{v_3, v_6} = Q_{v_3} + q_{v_3, v_6} = 170,30 + 180,80 = 350,110$$

$$F_{v_3, v_6} = 1,2544 + 0,6689 = 1,9233$$

$$P_{v_3, v_6} = v_s + v_3 + v_6$$

5. Knoten $\mu = v_4$:

Ab hier haben die betrachteten Knoten mehr als einen Vorgänger. Es werden daher nun durch die Kostenfunktion $g_\lambda(p)$ in den jeweiligen Knoten die Pfade mit dem höchsten Ressourcenverbrauch gelöscht. Es werden in den entsprechenden Knoten nur noch K Pfade weiter betrachtet. K wurde zu Beginn auf 1 gesetzt. Die Kostenfunktion $g_\lambda(p)$ (siehe Formel (1)) zur Berechnung der K Pfade ist in Yu et al. (2007) für den MCSP-K Algorithmus nicht exakt definiert. Laut Korkmaz und Krunkz (2001, S. 836) ist der Parameter λ bei einer linearen Aggre-

gation der nicht-funktionalen Zielgrößen 1, daher wird λ in der vorliegenden Berechnung auf 1 gesetzt. Der Parameter $q^m(p)$ stellt den Wert der aggregierten nicht-funktionale Zielgrößen bis zum entsprechenden Knoten dar (Yu et al. 2007, S. 13). Der Parameter Q^m ist in Yu et al. (2007) nicht beschrieben. In Korkmaz und Krunz (2001, S. 835) wird der Parameter als die minimale, aggregierte Summe der nicht-funktionalen Zielgrößen aller zulässigen Pfade angegeben. Da dies im vorliegenden Fall ein Vektor ist, wird hier für den Wert Q^m die Länge des kürzesten Vektors als Q^m aller zulässigen Pfade verwendet.

Selektion der K Pfade:

$$|Q|_{v_s, v_1, v_4} = \sqrt{300^2 + 100^2} = 316,23 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$|Q|_{v_s, v_3, v_4} = \sqrt{370^2 + 80^2} = 378,55$$

$$\text{Pfad } v_s, v_1, v_4: g_1(p) \triangleq \left(\frac{300}{316,23}\right)^1 + \left(\frac{100}{316,23}\right)^1 = 1,26 \rightarrow \text{Wird weitergeführt}$$

$$\text{Pfad } v_s, v_3, v_4: g_1(p) \triangleq \left(\frac{370}{316,23}\right)^1 + \left(\frac{80}{316,23}\right)^1 = 1,42 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_7$:

$$Q_{v_1, v_7} = Q_{v_1, v_4} + q_{v_4, v_7} = 300,100 + 150,70 = 450,170$$

$$F_{v_1, v_7} = 1,5759 + 1,0000 = 2,5759$$

$$P_{v_1, v_7} = v_s + v_1 + v_4 + v_7$$

Nachfolgeknoten $v = v_8$:

$$Q_{v_1, v_8} = Q_{v_1, v_4} + q_{v_4, v_8} = 300,100 + 120,110 = 420,210$$

$$F_{v_1, v_8} = 1,5759 + 0,5000 = 2,0759$$

$$P_{v_1, v_8} = v_s + v_1 + v_4 + v_8$$

6. Knoten $\mu = v_5$:

Selektion der K Pfade:

$$|Q|_{v_s, v_1, v_5} = \sqrt{260^2 + 150^2} = 300,17 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$|Q|_{v_s, v_3, v_5} = \sqrt{330^2 + 130^2} = 354,68$$

$$\text{Pfad } v_s, v_1, v_5: g_1(p) \triangleq \left(\frac{260}{300,17}\right)^1 + \left(\frac{150}{300,17}\right)^1 = 1,37 \rightarrow \text{Wird weitergeföhrt}$$

$$\text{Pfad } v_s, v_3, v_5: g_1(p) \triangleq \left(\frac{330}{300,17}\right)^1 + \left(\frac{130}{300,17}\right)^1 = 1,53 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_7$:

$$Q_{v_1, v_7} = Q_{v_1, v_5} + q_{v_5, v_7} = 260,150 + 150,70 = 410,220$$

$$F_{v_1, v_7} = 0,9312 + 1,0000 = 1,9312$$

$$P_{v_1, v_7} = v_s + v_1 + v_5 + v_7$$

Nachfolgeknoten $v = v_8$:

$$Q_{v_1, v_8} = Q_{v_1, v_5} + q_{v_5, v_8} = 260,150 + 120,110 = 380,260$$

$$F_{v_1, v_8} = 0,9312 + 0,5000 = 1,4312$$

$$P_{v_1, v_8} = v_s + v_1 + v_5 + v_8$$

7. Knoten $\mu = v_6$:

Selektion der K Pfade:

$$|Q|_{v_s, v_1, v_6} = \sqrt{280^2 + 130^2} = 308,71 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$|Q|_{v_s, v_3, v_6} = \sqrt{350^2 + 110^2} = 366,88$$

$$\text{Pfad } v_s, v_1, v_6: g_1(p) \triangleq \left(\frac{260}{308,71}\right)^1 + \left(\frac{150}{308,71}\right)^1 = 1,33 \rightarrow \text{Wird weitergeföhrt}$$

$$\text{Pfad } v_s, v_3, v_6: g_1(p) \triangleq \left(\frac{330}{308,71}\right)^1 + \left(\frac{130}{308,71}\right)^1 = 1,49 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_7$:

$$Q_{v_1, v_7} = Q_{v_1, v_6} + q_{v_6, v_7} = 280,130 + 150,70 = 430,200$$

$$F_{v_1, v_7} = 1,1521 + 1,0000 = 2,1521$$

$$P_{v_1, v_7} = v_s + v_1 + v_6 + v_7$$

Nachfolgeknoten $v = v_8$:

$$Q_{v_1, v_8} = Q_{v_1, v_6} + q_{v_6, v_8} = 280,130 + 120,110 = 400,240$$

$$F_{v_1,v_8} = 1,1521 + 0,5000 = 1,6521$$

$$P_{v_1,v_8} = v_s + v_1 + v_6 + v_8$$

Die in der Serviceklasse 2 nun weitergeführten Pfade sind folgende:

$$P_{v_1,v_7} = v_s + v_1 + v_4 + v_7$$

$$P_{v_1,v_8} = v_s + v_1 + v_4 + v_8$$

$$P_{v_1,v_7} = v_s + v_1 + v_5 + v_7$$

$$P_{v_1,v_8} = v_s + v_1 + v_5 + v_8$$

$$P_{v_1,v_7} = v_s + v_1 + v_6 + v_7$$

$$P_{v_1,v_8} = v_s + v_1 + v_6 + v_8$$

8. Knoten $\mu = v_7$:

Selektion der K Pfade:

$$|Q|_{v_s,v_1,v_4,v_7} = \sqrt{450^2 + 170^2} = 481,04$$

$$|Q|_{v_s,v_1,v_5,v_7} = \sqrt{410^2 + 220^2} = 465,30 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$|Q|_{v_s,v_1,v_6,v_7} = \sqrt{430^2 + 200^2} = 474,24$$

$$\text{Pfad } v_s, v_1, v_4, v_7: g_1(p) \triangleq \left(\frac{450}{465,30}\right)^1 + \left(\frac{170}{465,30}\right)^1 = 1,33 \rightarrow \text{Wird weitergeföhrt}$$

$$\text{Pfad } v_s, v_1, v_5, v_7: g_1(p) \triangleq \left(\frac{410}{465,30}\right)^1 + \left(\frac{220}{465,30}\right)^1 = 1,35 \text{ Wird gestrichen}$$

$$\text{Pfad } v_s, v_1, v_6, v_7: g_1(p) \triangleq \left(\frac{430}{465,30}\right)^1 + \left(\frac{200}{465,30}\right)^1 = 1,35 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_9$:

$$Q_{v_1,v_9} = Q_{v_1,v_7} + q_{v_7,v_9} = 450,170 + 280,70 = 730,240$$

$$F_{v_1,v_9} = 2,5759 + 0,8688 = 3,4447$$

$$P_{v_1,v_9} = v_s + v_1 + v_4 + v_7 + v_9$$

Da in diesem Pfad die Restriktion der Ausführungszeit (≤ 700) verletzt wurde, wird dieser Pfad nicht weiter betrachtet.

Nachfolgeknoten $v = v_{10}$:

$$Q_{v_1, v_{10}} = Q_{v_1, v_7} + q_{v_7, v_{10}} = 450,170 + 200,90 = 650,260$$

$$F_{v_1, v_{10}} = 2,5759 + 1,0607 = 3,6366$$

$$P_{v_1, v_{10}} = v_s + v_1 + v_4 + v_7 + v_{10}$$

Nachfolgeknoten $v = v_{11}$:

$$Q_{v_1, v_{11}} = Q_{v_1, v_7} + q_{v_7, v_{11}} = 450,170 + 160,180 = 610,350$$

$$F_{v_1, v_{11}} = 2,5759 + 0,3205 = 2,8964$$

$$P_{v_1, v_{11}} = v_s + v_1 + v_4 + v_7 + v_{11}$$

9. Knoten $\mu = v_8$:

Selektion der K Pfade:

$$|Q|_{v_s, v_1, v_4, v_8} = \sqrt{420^2 + 210^2} = 469,57$$

$$|Q|_{v_s, v_1, v_5, v_8} = \sqrt{380^2 + 260^2} = 460,43 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$|Q|_{v_s, v_1, v_6, v_8} = \sqrt{400^2 + 240^2} = 466,48$$

$$\text{Pfad } v_s, v_1, v_4, v_8: g_1(p) \triangleq \left(\frac{420}{460,43}\right)^1 + \left(\frac{210}{460,43}\right)^1 = 1,37 \rightarrow \text{Wird weitergeföhrt}$$

$$\text{Pfad } v_s, v_1, v_5, v_8: g_1(p) \triangleq \left(\frac{380}{460,43}\right)^1 + \left(\frac{260}{460,43}\right)^1 = 1,39 \text{ Wird gestrichen}$$

$$\text{Pfad } v_s, v_1, v_6, v_8: g_1(p) \triangleq \left(\frac{400}{460,43}\right)^1 + \left(\frac{240}{460,43}\right)^1 = 1,39 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_9$:

$$Q_{v_1, v_9} = Q_{v_1, v_8} + q_{v_8, v_9} = 420,210 + 280,70 = 700,280$$

$$F_{v_1, v_9} = 2,0759 + 0,8688 = 2,9447$$

$$P_{v_1, v_9} = v_s + v_1 + v_4 + v_8 + v_9$$

Nachfolgeknoten $v = v_{10}$:

$$Q_{v_1, v_{10}} = Q_{v_1, v_8} + q_{v_8, v_{10}} = 420,210 + 200,90 = 620,300$$

$$F_{v_1, v_{10}} = 2,0759 + 1,0607 = 3,1366$$

$$P_{v1,v10} = v_s + v_1 + v_4 + v_8 + v_{10}$$

Nachfolgeknoten $v = v_{11}$:

$$Q_{v1,v11} = Q_{v1,v8} + q_{v8,v11} = 420,210 + 160,180 = 580,390$$

$$F_{v1,v11} = 2,0759 + 0,3205 = 2,3964$$

$$P_{v1,v11} = v_s + v_1 + v_4 + v_8 + v_{11}$$

Da in diesem Pfad die Restriktion der Ausführungskosten (≤ 350) verletzt wurde, wird dieser Pfad nicht weiter betrachtet.

Die in der Serviceklasse 3 nun weitergeführten Pfade sind folgende:

$$P_{v1,v10} = v_s + v_1 + v_4 + v_7 + v_{10}$$

$$P_{v1,v11} = v_s + v_1 + v_4 + v_7 + v_{11}$$

$$P_{v1,v9} = v_s + v_1 + v_4 + v_8 + v_9$$

$$P_{v1,v10} = v_s + v_1 + v_4 + v_8 + v_{10}$$

10. Knoten $\mu = v_9$:

Nachfolgeknoten $v = v_d$:

$$Q_{v1,vd} = Q_{v1,v4,v8,v9,vd} = 700,280$$

$$F_{v1,vd} = F_{v1,v4,v8,v9,vd} = 2,9447$$

$$P_{v1,v9} = v_s + v_1 + v_4 + v_8 + v_9 + v_d$$

11. Knoten $\mu = v_{10}$:

Selektion der K Pfade:

$$|Q|_{vs,v1,v4,v7,v10} = \sqrt{650^2 + 260^2} = 700,07$$

$$|Q|_{vs,v1,v4,v8,v10} = \sqrt{620^2 + 300^2} = 688,77 \rightarrow \text{Kleinsten Wert} = Q^m$$

$$\text{Pfad } vs, v1, v4, v7, v10: g_1(p) \triangleq \left(\frac{650}{688,77}\right)^1 + \left(\frac{260}{688,77}\right)^1 = 1,32$$

\rightarrow Wird weitergeführt

$$\text{Pfad } vs, v1, v4, v8, v10: g_1(p) \triangleq \left(\frac{620}{688,77}\right)^1 + \left(\frac{300}{688,77}\right)^1 = 1,34 \text{ Wird gestrichen}$$

Nachfolgeknoten $v = v_d$:

$$Q_{v_1,v_d} = Q_{v_1,v_4,v_7,v_{10},v_d} = 650,260$$

$$F_{v_1,v_d} = F_{v_1,v_4,v_7,v_{10},v_d} = 3,6366$$

$$P_{v_1,v_{10}} = v_s + v_1 + v_4 + v_7 + v_{10} + v_d$$

12. Knoten $\mu = v_{11}$:

Nachfolgeknoten $v = v_d$:

$$Q_{v_1,v_d} = Q_{v_1,v_4,v_7,v_{11},v_d} = 610,350$$

$$F_{v_1,v_d} = F_{v_1,v_4,v_7,v_{11},v_d} = 2,8964$$

$$P_{v_1,v_{11}} = v_s + v_1 + v_4 + v_7 + v_{11}$$

12. Knoten $\mu = v_d$:

Nach Zeile 15 der Beschreibung des Algorithmus MCSP (siehe Abbildung 4-5) wird nun der Pfad mit dem höchsten Nutzen gewählt. Dieser ist folgender:

$$Q_{v_1,v_d} = Q_{v_1,v_4,v_7,v_{10},v_d} = 650,260$$

$$F_{v_1,v_d} = F_{v_1,v_4,v_7,v_{10},v_d} = 3,6366$$

$$P_{v_1,v_{10}} = v_s + v_1 + v_4 + v_7 + v_{10} + v_d$$

Die abschließend vom heuristischen Algorithmus MCSP-K selektierte Servicekette lautet:

$$P_{v_1,v_{10}} = v_s + v_1 + v_4 + v_7 + v_{10} + v_d$$

Sie entspricht der Servicekette: Finale Servicekette = $s_{11} + s_{21} + s_{31} + s_{42}$.

Prüfung der Restriktionen:

Ausführungszeit prüfen: $Q = 100 + 200 + 150 + 200 = 650 \rightarrow Q \leq Q_c$

Ausführungskosten prüfen: $Q = 50 + 50 + 70 + 90 = 260 \rightarrow Q \leq Q_c$

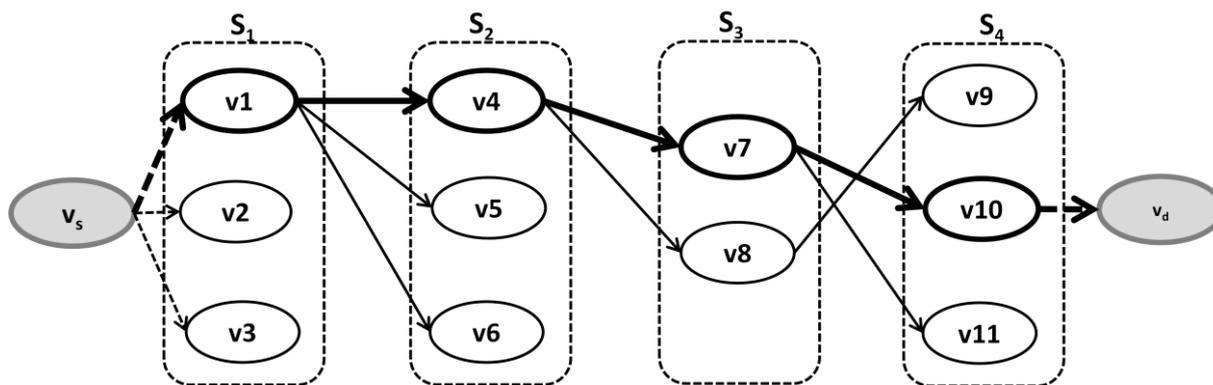


Bild 4-8 *Selektierte Servicekette durch MCSP-K*

Die Abbildung 4-8 zeigt in Anlehnung an Abbildung 4-7 die selektierte Servicekette (fett gedruckter Pfad). Des Weiteren sind die Pfade entfernt, welche von anderen Pfaden dominiert wurden oder im Zuge der Kostenberechnung gestrichen wurden. Die Ausführungszeit der durch den Algorithmus MCSP-K selektierten Servicekette ist 650 Zeiteinheiten (Restriktion 700 Zeiteinheiten) und die Ausführungskosten belaufen sich auf 260 Geldeinheiten (Restriktion 350 Geldeinheiten). Der Nutzen der selektierten Servicekette ist 3,6366. Zum Vergleich: Die optimale Lösung, welche durch den BBLP Algorithmus gefunden wurde, lautet: $s_{13} + s_{21} + s_{32} + s_{42}$. Sie hat eine Ausführungszeit von 690 Zeiteinheiten und Ausführungskosten von 280 Geldeinheiten. Der Nutzen der optimalen Lösung ist 3,9078.

Eine effizientere Art solch ein graphisches Modell zu berechnen bietet die dynamische Programmierung. Die dynamische Programmierung wird bspw. beim Web Service Selektionsverfahren von Gao et al. (2006) angewendet.

5 Monitoring zur Bestimmung der Qualität von Web Services bei Web Service Anbietern und in Dienstgüterezeichnissen

In diesem Kapitel soll gezeigt werden, inwieweit einzelne nicht-funktionale Zielgrößen in Web Service Dienstverzeichnissen und bei Web Service Anbietern dokumentiert werden. Dabei wird speziell dargestellt, welche nicht-funktionalen Zielgrößen dokumentiert sind, über welchen Zeitraum sich die Dokumentation erstreckt und für wie viele der angebotenen Web Services Werte vorliegen. Als Basis für eine qualitative Web Service Selektion sind diese Informationen entscheidend. Eine langfristige und fundierte Dokumentation möglichst vieler nicht-funktionalen Zielgrößen ist ein wichtiges Kriterium zur Entscheidungsfindung des Web Service Nutzers darüber, welche konkreten Web Services für die jeweiligen Aktionen eingesetzt werden. Außerdem ist die Dokumentation maßgeblich an der Entscheidung beteiligt, welche Web Service Anbieter verwendet werden und in einen Prozess integriert werden.

5.1.1 Web Service Dienstverzeichnisse

XMethods: <http://www.xmethods.net>

XMethods stellt ein Dienstverzeichnis zur Verfügung, auf welchem Web Service Anbieter ihre Web Services anbieten können. Neben einer Beschreibung des jeweiligen Web Services werden aber keine Informationen zu nicht-funktionalen Zielgrößen gemacht.

RemoteMethods: <http://www.remotemethods.com>

Ein Dienstverzeichnis für Web Services stellt auch RemoteMethods zur Verfügung. Die darin aufgelisteten Web Services enthalten neben einer kurzen Beschreibung auch Kosteninformationen zu Nutzung der Web Services. Des Weiteren wird, falls vorhanden, die Möglichkeit angeboten, Resonanzen anderer Nutzer zu lesen oder selbst eine Bewertung abzugeben.

eSigma: <http://www.esigma.com>

Das Dienstverzeichnis eSigma beinhaltet eine Vielzahl an Web Services zu unterschiedlichen Themengebieten wie bspw. aus dem Marketingbereich oder Finanzbereich. Zu den bei eSigma gelisteten Web Services wird die Verfügbarkeit der letzten 30 Tage und die durchschnittliche Antwortzeit der vergangenen 14 Tage angegeben. Abbildung 5-1 zeigt, in welcher Form die nicht-funktionalen Zielgrößen Verfügbarkeit und Antwortzeit dargestellt werden. Für den Web Service FastWeather wurden eine Verfügbarkeit von 99,67% und eine durchschnittliche Ausführungszeit von 324,17 Millisekunden dokumentiert.



Bild 5-1 Dokumentation nicht-funktionaler Zielgrößen bei eSigma

WebServiceList: <http://www.webservicelist.com>

Im Dienstverzeichnis WebserviceList werden Web Services zu vielen unterschiedlichen Kategorien angeboten. Zu jedem Web Service ist eine kurze Beschreibung der Funktion angegeben. Weitere Informationen zu nicht-funktionalen Zielgrößen werden nicht genannt. Lediglich eine Beurteilung durch Vergabe von einem bis zehn Sternen durch Nutzer der Web Services und das Datum der letzten Aktualisierung des Web Services sind auf der Internetseite angegeben.

Soa Trader Ltd.: <http://www.soatrader.com>

Eine Vielzahl von Web Services in unterschiedlichen Kategorien sind auf der Internetseite von Soa Trader Ltd. gelistet. Zu den Web Services werden eine kurze Beschreibung, die durchschnittliche Verfügbarkeit und durchschnittliche Antwortzeit der vergangenen 360 Tage sowie die Kosten angegeben. Die exakte Anzahl der aufgelisteten Web Services konnte nicht herausgefunden werden. Es wurde jedoch durch das Überprüfen mehrerer Web Services herausgefunden, dass mit wenigen Ausnahmen alle Web Services eine durchschnittliche Antwortzeit von 0 Millisekunden und eine durchschnittliche Verfügbarkeit von 99,7% haben. Des Weiteren haben nahezu alle überprüften Web Services identische Ausführungskosten. Lediglich die neuesten Web Services, welche auf der Startseite der Internetseite gelistet sind, haben unterschiedliche Preise bzw. durchschnittliche Antwortzeiten.

Seekda GmbH: <http://webservices.seekda.com>

Eine große Palette an unterschiedlichen Web Services (22.272 Web Services, Stand 11.08.2011) von unterschiedlichen Web Service Anbietern listet die Firma seekda GmbH auf. Die aufgelisteten Web Services der verschiedenen Web Service Anbieter sind im Verzeichnis Web Services zu finden und werden dort zum Kauf bzw. zur Nutzung angeboten. Die dokumentierten nicht-funktionalen Zielgrößen sind die Verfügbarkeit und die Ausführungszeit sowie eine Nutzerbewertung. So ist bspw. für den Web Service StockQuote (<http://webservices.seekda.com/providers/webservicex.com/StockQuote>), der Aktienkurse anzeigt, Folgendes dokumentiert:

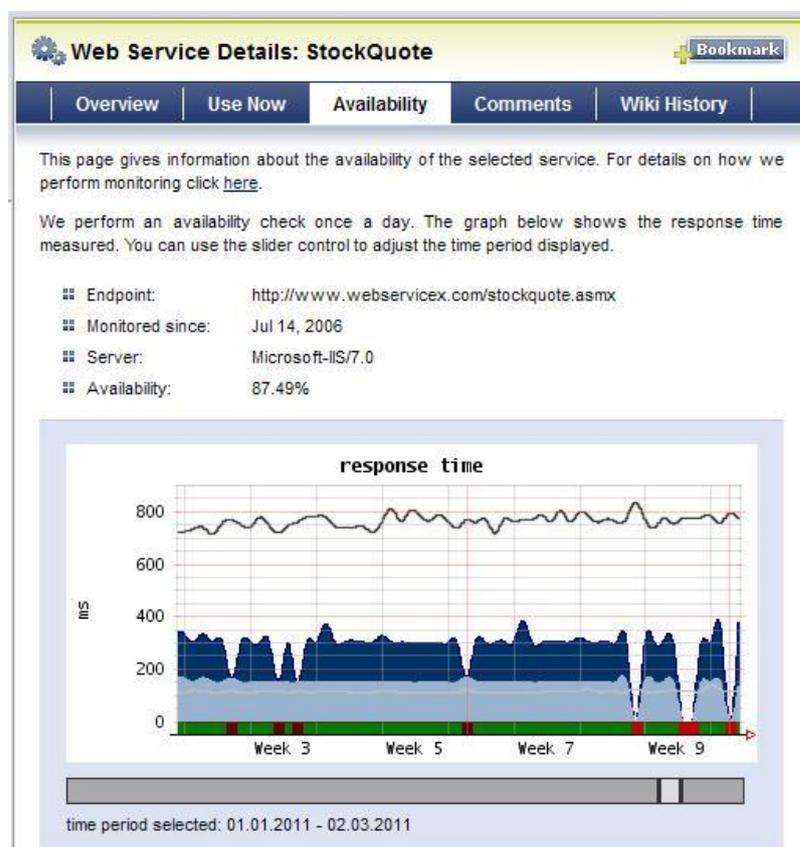


Bild 5-2 Dokumentation nicht-funktionaler Zielgrößen bei seekda

Aus der Abbildung 5-2 ist zu entnehmen, dass die nicht-funktionalen Zielgrößen seit dem 14. Juli 2006 dokumentiert werden. Die Verfügbarkeit liegt bei durchschnittlichen 87,49% und die Ausführungszeit für den Zeitraum vom 01.01.1011 bis zum 02.03.2011 durchschnittlich zwischen 700 und 800 Millisekunden (oberste Linie im Ausschnitt response time). Konkrete Angaben zu den Kosten werden nicht gemacht. Vereinzelt jedoch wird in der Beschreibung des Web Services eine kostenfreie Nutzung erwähnt, so z.B. beim Web Service „GlobalWeather“, welcher Wetterdaten für Großstädte auf der ganzen Welt liefert.

Programmableweb: <http://www.programmableweb.com>

Auf der Internetseite von Programmableweb.com sind 3647 Web Services aufgelistet (Stand 11.08.2011). Alle aufgelisteten Web Services werden kurz beschrieben. Des Weiteren werden die Nutzungsbedingungen der Web Services angegeben, wie bspw. die Kosten oder zeitliche Beschränkungen. Weitere nicht-funktionale Zielgrößen wie Verfügbarkeit oder Ausführungszeit werden nicht angegeben.

5.1.2 Web Service Anbieter

CDYNE: <http://www.cdyne.com>

Beim Dienstgüterverzeichnis CDYNE wird auf der Homepage eine per SLA garantierte Verfügbarkeit von 100% für alle angebotenen Web Services gewährleistet. Durch ein Fremdunternehmen (<http://status.watchmouse.com/16077>) wird dem Kunden für alle Web Services eine Historie über das Ausmaß der Verfügbarkeit bereitgestellt. Die zur Verfügung stehenden Daten umfassen einen Zeitraum von sieben Tagen. Des Weiteren werden die Ausführungskosten der Web Services bzw. die Nutzungskosten des Web Service Anbieters genannt. Die Kosten beinhalten eine monatliche Grundgebühr von \$9,99 plus die Kosten für die Web Service Ausführungen. Die Ausführungskosten sind für alle angebotenen CDYNE Web Services gleich und gestaffelt nach der Anzahl der Transaktionen.

Fraudlabs: <http://www.fraudlabs.com>

Die Firma Fraudlabs dokumentiert auf ihrer Internetseite für alle angebotenen Web Services ausschließlich die momentane Verfügbarkeit. Weitere nicht-funktionale Zielgrößen oder eine Historie der Verfügbarkeit werden nicht zur Verfügung gestellt. Zu jedem Web Service werden zudem Informationen zu den Kosten zur Verfügung gestellt. Diese teilen sich in einen festen monatlichen oder jährlichen Abonnementpreis und eine Gebühr für Credits auf, welche im Voraus bezahlt werden. Die Menge der Credits ist nach der gewünschten Anzahl der Serviceaufrufe pro Monat gestaffelt. Ein Kredit stellt dabei einen Serviceaufruf dar.

Xignite: <http://www.xignite.com>

Xignite ist ein Anbieter von Web Services von Daten über Finanzmärkte wie bspw. Web Services zu Daten über US-Aktienbörsen. Der Web Service Anbieter bietet im Moment 53 Web Services an. Als einzige nicht-funktionale Zielgröße wird die monatliche Verfügbarkeit seit Januar 2007 angegeben. Die zur angebotene Verfügbarkeit bezieht sich jedoch nur auf die des Systems, also auf die Infrastruktur der Firma, und betrifft somit alle angebotenen Web Services. Daten über einzelne Web Services oder weitere nicht-funktionale Zielgrößen stellt das Unternehmen nicht zur Verfügung. Die Daten über die Verfügbarkeit werden von einem Subunternehmen (<http://www.keynote.com>) erfasst. Die Kosten sind von Xignite nach Anzahl der gewünschten Serviceaufrufe gestaffelt und werden im Voraus verrechnet. Xignite bietet sechs

Stufen ausgehend von 600 Serviceaufrufen pro Monat bis zu einer unendlichen Anzahl von Serviceaufrufen an. Bei einer halbjährlichen, jährlichen oder zweijährlichen Bezahlung des Kontingents im Voraus werden Preisnachlässe gewährt.

Serviceobjects: <http://www.serviceobjects.com>

Serviceobjects stellt für den Zeitraum von März 2006 bis heute Performance-Berichte zur Verfügung, welche von einer unabhängigen Firma (AlertSite Inc) erstellt werden. Diese enthalten Daten über die nicht-funktionalen Zielgrößen Verfügbarkeit, Antwortzeit und Systemperformance aller Web Services zusammen. Angaben zu einzelnen Web Services werden nicht gemacht. Erwähnenswert ist, dass die Firma durch ein SLA eine Verfügbarkeit von 99,995% garantiert. Zudem wird in den SLA eine Internetlatenz von 100 Millisekunden im Backbone Netzwerk und ein Paketverlust von nicht mehr als 1% im Backbone Netzwerk garantiert. Der Paketverlust kann mit der nicht-funktionalen Zielgröße Fehlerrate verglichen werden. Die Kosten der Nutzung sind in Abonnements gestaffelt, welche eine unterschiedliche Anzahl an monatlichen Transaktionen also Serviceaufrufen beinhalten.

Amazon Web Services: <http://aws.amazon.com/de>

Die Firma Amazon Web Services bietet eine Vielzahl an unterschiedlichen Web Services an. Eine Übersicht der nicht-funktionalen Zielgrößen existiert nicht. Jedoch wird für einzelne Web Services eine Verfügbarkeitsgarantie gegeben, bspw. für den Web-Service Amazon Elastic Compute Cloud (Amazon EC2) eine garantierte Verfügbarkeit von 99,95%. Des Weiteren werden nur allgemeine Angaben gemacht, wie bspw., dass alle Web Services skalierbar, zuverlässig und sicher sind. Die Kosten sind je nach Web Service unterschiedlich gestaltet. So sind Preismodelle unter anderem nach Übertragung ausgehender Daten oder nach Dauer der Nutzung gestaltet.

Salesforce: <http://www.salesforce.com/de>

Eine detaillierte Dokumentation nicht-funktionaler Zielgrößen zu den angebotenen Web Services bietet auch die Firma Salesforce nicht an. Jedoch werden auf der Internetseite der Firma Angaben zu IT-Sicherheit, Datenschutz und Verfügbarkeit gemacht. So wurde in den vergangenen 8 Monaten ein Durchschnittswert von über 99,9% Verfügbarkeit und eine durchschnittliche Antwortzeit von unter 300ms bei allen angebotenen Web Services erreicht. Auf der Internetseite <https://trust.salesforce.com/trust/de/index.html> werden Maßnahmen genannt, welche zur Gewährleistung einer maximalen Sicherheit getroffen sind. Des Weiteren wird eine Übersicht zur Systemperformance der vergangenen 28 Tage zur Verfügung gestellt. Diese beinhaltet u. a. die durchschnittliche Ausführungszeit der Web Services und die tägliche Verfügbarkeit der einzelnen Web Services. Die Preisgestaltung in ist Abonnements gegliedert, welche unterschiedliche Leistungen beinhalten.

Google Code: <http://code.google.com>

Google bietet eine Reihe an Web Services an, wie etwa einen Web Service zum Erstellen einer Applikation (App) für die Software Android von Google. Auf der Seite werden keine Informationen zu nicht-funktionalen Zielgrößen gemacht. Auch zu den Kosten werden außer zum Web Service Google Checkout keine Infos gegeben. Der Web Service Google Checkout übernimmt die Zahlungsfunktion für einen Onlineshop. Dabei werden über die Internetseite des Onlinehändlers die Zahlungsinformationen an Google weitergegeben. Google führt unter anderem eine Risikoüberprüfung des Käufers durch und nachdem Google die Kreditkarte des Käufers belastet hat, wird das Geld abzüglich einer Gebühr an den Onlinehändler weitergeleitet. Der versprochene Vorteil des Web Services Google Checkout ist, dass Kunden, welche schon bei anderen Onlinehändlern eingekauft haben, die Google Checkout nutzen, mit dem Bezahlprozess vertraut sind und eine einfache und schnelle Bezahlprozedur vorzufinden wissen. Der Onlinehändler verringert sein Geschäftsrisiko durch Auslagerung des Bezahlprozesses und dessen Risiken und kann sich damit nahezu ausschließlich auf den Verkauf seiner Produkte oder Dienstleistungen konzentrieren. Die Gebühren des Web Services sind nach der Höhe des monatlichen Umsatzes und einer festen Transaktionsgebühr pro Einkauf gestaffelt. Bspw. sind bei einem monatlichen Gesamtverkaufsvolumen von unter \$3000 die von Google einbehaltenen Gebühren 2,9% des Umsatzes plus \$0,30 pro Bezahltransaktion. Bei einem Gesamtverkaufsvolumen von über monatlich über \$100.000 senken sich die Gebühren auf 1,9%. Die Gebühren von \$0,30 pro Bezahltransaktion bleiben gleich.

Yahoo! Developer Network: <http://developer.yahoo.com/>

Die Firma Yahoo! Inc. bietet eine große Bandbreite an Web Services an. Aktuell werden in 13 Kategorien insgesamt 101 Web Services angeboten. Die Nutzung der Web Services von Yahoo! ist kostenfrei, jedoch wird in den Nutzungsbedingungen darauf verwiesen, dass die Web Services auch kostenpflichtig werden können. Außerdem wird seitens Yahoo! die tägliche Anzahl der Serviceaufrufe auf 5000 limitiert. Bei einer höheren erwarteten Anzahl der Web Service Aufrufe muss die Firma Yahoo! Inc. kontaktiert werden. Zu nicht-funktionalen Zielgrößen werden von Yahoo! keine Angaben gemacht.

WebserviceX.NET: <http://www.webservicex.net>

Die Firma Generic Objects Technologies Ltd bietet auf der Internetseite webservicex.net eine Vielzahl an unterschiedlichen Web Services an. Laut eigener Aussage auf der Startseite werden täglich über 6 Millionen Web Service Transaktionen über webservicex.net bedient. Angaben zu nicht-funktionalen Zielgrößen werden nicht gemacht.

X Web Services: <http://www.xwebservices.com>

Die Firma XWebServices Inc. bietet u. a. Web Services für den E-Commerce Bereich, für das Content Management oder für Banken sogenannte Financial Services an. Jeder der elf ange-

botenen Web Services wird ausführlich beschrieben, Angaben zu den Funktionalitäten werden gemacht und eine Anleitung für eine einfache Implementierung wird bereitgestellt. Es werden zwar Angaben zu den Kosten gemacht, jedoch fehlen jegliche Angaben zu weiteren nicht-funktionalen Zielgrößen wie Verfügbarkeit oder Ausführungszeiten, welche im Bereich Banken oder E-Commerce durchaus wichtig sind.

LOKAD: <http://www.lokad.com>

Das Unternehmen stellt keine Angaben zu nicht-funktionalen Zielgrößen auf der Internetseite der Firma zur Verfügung. Die Kosten werden pro Servicenutzung erhoben.

Unified Software: <http://www.unifiedsoftware.co.uk>

Unified Software bietet u. a. Web Services für die Validierung von Daten aus dem Bankbereich und Web Services für die Verarbeitung von Zahlungen an. Seit der Markteinführung der sogenannten BankVal Web Services 2003 ist eine Verfügbarkeit von 100% gemessen worden. Die Sicherheit der Daten wird durch eine 128-Bit-Verschlüsselung gewährleistet. Weitere nicht-funktionale Zielgrößen werden auf der Internetseite des Unternehmens nicht veröffentlicht. Die Kosten der Web Service Nutzung sind pro Serviceaufruf und am Ende des Monats zu entrichten.

AdressDoctor: <http://www.addressdoctor.com>

Die Firma AdressDoctor stellt auf seiner Internetseite nur begrenzt Angaben zu nicht-funktionalen Zielgrößen bereit. So ist die durchschnittliche Ausführungszeit mit 100 Millisekunden angegeben. Für Serviceaufrufe aus Deutschland wird eine Ausführungszeit von 50 Millisekunden angegeben. Die Ausführungszeiten von Ländern außerhalb Europas weichen jedoch stark ab. So ist etwa für Brasilien eine Ausführungszeit von 420 Millisekunden genannt. Durch ein SLA wird dem Web Service Nutzer eine Verfügbarkeit von 99,5% oder 99,7% garantiert. In den vergangenen Jahren hat die Verfügbarkeit 99,9% übertroffen. Die Kosten gestalten sich auf zwei Arten. Zum einen ist es möglich ein Kontingent an Transaktionen zu erwerben, welches im Voraus bezahlt wird, zum anderen ist eine Bezahlung pro Transaktion möglich. Die Bezahlung pro Transaktion beinhaltet jedoch eine Grundgebühr, welche quartalsweise im Voraus zu entrichten ist.

StrikeIron: <http://www.strikeiron.com>

StrikeIron bietet eine Reihe an Web Services an, bspw. einen Web Service zur Validierung von Telefonnummern in Nordamerika und einen Web Service zur Umrechnung von Währungen in Fremdwährungen. Für keine dieser Web Services werden jedoch nicht-funktionale Zielgrößen bereitgestellt oder sind dokumentiert. Die Kosten der Web Services sind auf der Internetseite ebenfalls nicht verfügbar, jedoch kann zwischen einem jährlichen und monatli-

chen Kontingent gewählt werden. Für detailliertere Informationen muss die Firma direkt kontaktiert werden.

eCoComa: <http://www.ecocoma.com>

Das Web Service Angebot von eCoComa umfasst derzeit 52 konkrete Web Services, eingeteilt in acht Kategorien, z.B. werden in der Kategorie Versandservices Web Services für Preisinformationen von einigen großen Paketdienstunternehmen wie DHL oder FedEx angeboten. Jede Web Service Kategorie wird kurz beschrieben, außerdem wird die Funktionalität jedes einzelnen Web Service in einem Satz wiedergegeben. Leider werden zu keinem Web Service irgendwelche nicht-funktionalen Zielgrößen genannt. Auch zu den Kosten werden auf der Internetseite keine Informationen genannt.

6 Zusammenfassung, kritische Würdigung und Ausblick

Die Aufbauorganisation eines Unternehmens muss seinen Prozessen angepasst sein. Das Ziel eines Unternehmens ist deshalb die Effizienz seiner Prozesse und damit deren Beschleunigung, Kostenreduktion und Transparenz. (Lassmann et al. 2006, S. 298) Eine Möglichkeit, dieses Ziel in die Tat umzusetzen, ist die Verwendung von Web Services.

Web Services und serviceorientierte Architekturen sind Lösungen, die eine flexible Gestaltung von Geschäftsprozessen innerhalb eines Unternehmens und unternehmensübergreifend ermöglichen. Im Gegensatz zu traditionellen Workflow-Systemen kann damit praktisch sofort, ohne monatelange Wartezeiten, wesentlich einfacher auf geschäftliche Veränderungen reagiert werden. Einzelne Web Services werden je nach Bedarf aufgerufen und verwendet. Durch die Kombination mehrerer Web Services entsteht eine Web Service Komposition, welche Unternehmensprozesse funktional abbildet, mit dem entscheidenden Vorteil, diese flexibel verändern zu können. (Reynolds 2010, S. 72)

In den vergangenen Jahren ist die Zahl der auf dem Markt angebotenen Web Services sprunghaft angestiegen. (Legner 2009, S. 36-37; Arpinar et al. 2005; Motahari-Nezhad et al. 2010, S. 731) Folglich ist auch die Anzahl der funktional identischen Web Services angestiegen. Mittlerweile sind eine ganze Reihe an Web Service Anbietern auf dem Markt vertreten und es werden immer mehr. (Nüttgens und Dirik 2008, S. 37) Des Weiteren werden auf dem Web Service Markt zunehmend Web Services von Unternehmen zur Verfügung gestellt, welche ursprünglich für den internen Gebrauch gedacht und entwickelt wurden. (Nüttgens und Dirik 2008, S. 32; Legner 2009, S. 38-40) Mittelfristig ist durch den hohen Grad der Standardisierung der Web Services die Möglichkeit gegeben, sie flexibel einzusetzen und durch deren Interoperabilität eine steigende Nachfrage zu erwarten. (Legner 2009, S. 31) Gleichzeitig wird aber auch besonders durch die Möglichkeit, einen Web Service ohne viel Aufwand praktisch täglich zu wechseln, ein verschärfter Konkurrenzkampf auf dem Web Service Markt eintreten. Im Zuge des Konkurrenzkampfes offenbaren sich für Unternehmen hohe Potentiale zur Kostensenkung und zu einer flexibleren Gestaltung ihrer IT-Landschaft. Da die Web Service Technologie standortungebunden ist, können einzelne Geschäftsprozesse durch Web Service Kompositionen wesentlich einfacher mittels Outsourcing an IT-Dienstleister vergeben werden. (Nüttgens und Dirik 2008, S. 37) Ist der Gebrauch von Web Services einmal weit verbreitet, können sie in der Wertschöpfungskette eines Unternehmens weit über nur zwei Partner hinaus integriert werden. Des Weiteren bieten sie die Möglichkeit den Mangel an Flexibilität und Anpassungsfähigkeit der traditionellen IT zu beheben und bieten eine Datenverarbeitung in Echtzeit. (Reynolds 2010, S. 72-73)

Mit der Zunahme der verfügbaren Web Services pro Aktion in einer Aktionskette wird eine fundierte, zuverlässige und zeitlich akzeptable Selektion von Web Services anhand nicht-funktionaler Zielgrößen immer schwerer. Zugleich wird sie dadurch aber auch erst ermöglicht. Die angebotenen Web Services können bei funktionaler Gleichartigkeit durch nicht-funktionale Zielgrößen unterschieden werden. Nicht-funktionale Zielgrößen sind bspw. die

Kosten der Nutzung, die Ausführungszeit oder die Zuverlässigkeit eines Web Services. Sie stellen also den Grad der Güte dar, mit welcher die Funktion (funktionale Zielgröße) eines Web Services ausgeführt wird.

Für die Selektion der Web Services, d. h. für die Auswahl der Web Services, welche die Präferenzen der Web Service Nutzer am Besten erfüllen, gibt es in der Literatur bereits viele Selektionsverfahren. Diese Selektionsverfahren lassen sich durch ihre Selektionsmethode unterscheiden, d. h. auf welche Art und Weise die Selektion erfolgt. Die Web Service Selektionsverfahren bedienen sich dabei unterschiedlicher mathematischer Optimierungsverfahren, wie bspw. der linearen Integer Programmierung oder dem Branch and Bound Verfahren. Es existieren zum einen Selektionsverfahren, welche in der Lage sind, eine exakte Lösung des Optimierungsproblems zu finden, also die optimale Servicekette zu selektieren. Und zum anderen existieren Selektionsverfahren, welche eine nahezu optimale, auf jeden Fall aber ausreichende Lösung finden. Dies sind die sogenannten heuristischen Selektionsverfahren, sie selektieren die Servicekette wesentlich leistungsstärker bzgl. Selektionszeit und Größe des Optimierungsproblems. Seit der Entwicklung der ersten Selektionsverfahren etwa im Jahre 2002 sind die Selektionsverfahren bis heute in ihrer Leistungsfähigkeit weiterentwickelt worden. Aktuell stellen Selektionsverfahren basierend auf sogenannten genetischen Algorithmen den vielversprechendsten Trend dar und werden stetig weiterentwickelt. Jedoch können auch diese bisher nicht alle Herausforderungen der Web Service Selektion bewältigen. So ist bisher noch kein Selektionsverfahren in der Lage Abhängigkeiten der nicht-funktionalen Zielgrößen zu beachten.

Bis heute fehlte eine umfangreiche Übersicht der verfügbaren Web Service Selektionsverfahren. Mit dieser Arbeit wurde ein wichtiger Schritt in diese Richtung vollzogen. Bemängelt werden muss aber die uneinheitliche Evaluierung der Selektionsverfahren durch die Autoren. Der Vergleich der Web Service Selektionsverfahren würde wesentlich erleichtert werden, wenn eine einheitliche Evaluierung, also mit einheitlichen Größen der Optimierungsprobleme und Anzahl an nicht-funktionalen Zielgrößen, geschaffen würde. Bezüglich der veröffentlichten Arbeiten der Entwickler, in welchen sie Web Service Selektionsverfahren vorstellen, ist zu bemängeln, dass die von den Autoren selbst entwickelten Selektionsverfahren nur wenig kritisiert werden. D. h. deutliche Schwächen werden nur kurz bis überhaupt nicht in den Arbeiten erwähnt. Ebenso wird auf Verfahren anderer Autoren in vielen Arbeiten der Autoren sehr wenig eingegangen oder diese kritisiert.

Derzeit ist das Forschungsgebiet der Web Service Selektionsverfahren noch nicht ausgeschöpft. Aufgrund der noch nicht vollständig ausgereiften Selektionsverfahren und der steigenden Bedeutung von Web Services und der Notwendigkeit der Selektion sind zukünftig weitere Verbesserungen zu erwarten.

SOA und Web Services sind ein weiterer Schritt dahin, dass die IT durch das Business bestimmt wird und nicht andersherum. Hierzu muss jedoch auch beachtet werden, dass bei der Erstellung und besonders im Hinblick auf die Selektionsverfahren neben der technischen Seite auch der betriebswirtschaftliche Aspekt ausreichend berücksichtigt werden muss. Unternehmen, die zwar fachlichen Voraussetzungen erfüllen, aber Kundenanforderungen ungenügend

berücksichtigen, werden keinen Erfolg haben. (Nüttgens 2008, S. 31; Kontogiannis et al. 2007, S. 6)

Um eine qualitative Web Service Selektion durchzuführen, benötigt es neben den Web Service Selektionsverfahren eine fundierte Dokumentation der nicht-funktionalen Zielgrößen. D. h. es müssen zu jedem verfügbaren Web Service ausreichende und über einen längeren Zeitraum erfasste Informationen über dessen nicht-funktionalen Zielgrößen vorliegen. Idealerweise wären das bei einem kostenpflichtigen Web Service, welcher in einem Unternehmen für einen wichtigen Geschäftsprozess verwendet werden soll, die Kosten, die Ausführungszeit und die Verfügbarkeit. Sollten sensible Daten übertragen werden, ist zudem die Sicherheit ein wichtiges Kriterium. Weitere nicht-funktionale Zielgrößen können natürlich ebenfalls relevant sein und eingebunden werden.

Ein Blick in die Realität der Dokumentation der nicht-funktionalen Zielgrößen in Kapitel 5 zeigt jedoch ein schlechtes Bild. Die dort genannten Web Service Anbieter und Web Service Dienstverzeichnisse dokumentieren nicht-funktionale Zielgrößen mit wenigen Ausnahmen sehr mangelhaft oder überhaupt nicht. Einzig Informationen zu den Kosten werden bei fast allen Anbietern genannt. Zu den nicht-funktionalen Zielgrößen Verfügbarkeit oder Ausführungszeit werden besonders von Web Service Anbietern nur wenige Informationen auf deren Internetseiten preisgegeben. Hierbei muss natürlich auch in Betracht gezogen werden, dass unter Umständen diese Informationen erst bei einem ernsthaften Kaufinteresse zur Verfügung gestellt werden. Dies ist aber im Rahmen dieser Arbeit nicht untersucht worden. Des Weiteren muss auch bedacht werden dass bei kostenlosen Web Services wie bspw. einer Wetterauskunft eine qualitative Dokumentation der nicht-funktionalen Zielgrößen aufgrund des Aufwandes nicht sinnvoll ist.

Fakt ist, dass eine qualitative Dokumentation der nicht-funktionalen Zielgrößen gemacht werden muss. Denn dies ist besonders in einer sensiblen Umgebung sinnvoll, wie sie in Unternehmen vorzufinden ist. Sich allein auf die Versprechen der Web Anbieter zu stützen reicht nicht aus. Es müssen Leistungen und nicht-funktionale Zielgrößen in SLAs definiert und durch Dokumentation auch nachvollziehbar gemacht werden. Ein Einsatz von Web Services mit schlecht dokumentierten nicht-funktionalen Zielgrößen stellt ein unkalkulierbares Risiko für ein Unternehmen dar.

Im Angesicht der immer größer werdenden Anzahl an Web Services muss besonders für kostenpflichtige Web Services eine qualitative und einheitliche Dokumentation der nicht-funktionalen Zielgrößen geschaffen werden. Damit würde ein Vergleich von Web Services deutlich erleichtert werden. Auf Basis dieser einheitlichen Daten würden Selektionsverfahren in der Lage sein eine qualitative Web Service Selektion durchführen zu können. In weiterer Folge ist es auch wesentlich leichter möglich, die bis dato noch unzureichend erfassten Herausforderungen der Web Serviceselektion, wie etwa eine Berücksichtigung von Abhängigkeiten zwischen nicht-funktionalen Zielgrößen, in die Web Service Selektion mit einzubeziehen.

Realistisch gesehen ist eine einheitliche Dokumentation aber nur schwer durchzusetzen, da manche Unternehmen kein Interesse haben nicht-funktionale Zielgrößen preiszugeben oder diese erst auf Nachfrage oder bei ernsthaftem Interesse zur Verfügung stellen. Dennoch muss einem Web Service Anbieter bewusst sein, dass er sich durch eine ausführliche und qualitative Dokumentation von seiner Konkurrenz abheben kann und sich dadurch einen Vorsprung im Wettbewerb um Kunden erarbeiten kann. Durch eine langfristige Dokumentation, d. h. durch aussagekräftige nicht-funktionale Zielgrößen wird ein deutlicher Mehrwert kreiert, der nicht nur den Kunden zugute kommt.

Literatur

Aggarwal, Rohit; Verma, Kunal; Miller, John; Milnor, William (2004a): Constraint Driven Web Service Composition in METEOR-S. In: Proceedings of the IEEE International Conference on Services Computing (SCC 2004), Shanghai, China, S. 23-30.

Aggarwal, Rohit; Verma, Kunal; Miller, John; Milnor, William (2004b): Dynamic Web Service Composition in METEOR-S. Technischer Report, LSDIS Lab, Computer Science Department, University of Georgia, Athens.

Ai, Lifeng; Tang, Maolin (2008): A Penalty-based Genetic Algorithm for QoS-Aware Web Service Composition with Inter-Service-Dependencies and Conflicts. In: Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation (CIMCA'08), IEEE Computer Society, Wien, Austria, S. 738-743.

Alrifai, Mohammad; Risse, Thomass (2009): Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th international Conference on World Wide Web (WWW 2009), ACM, New York, NY, USA S. 881–890.

Alrifai, Mohammad; Skoutas, Dimitrios; Risse, Thomass (2010): Selecting skyline services for QoS-based web service composition. In: Proceedings of the 19th international Conference on World Wide Web WWW 2010), ACM, New York, NY, USA, S. 11–20.

Aiello, Marco; Khoury, Ellie el; Lozovik, Alexander; Ratelband, Patrick (2009): Optimal QoS-Aware Web Service Composition. In: Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC'09), IEEE Computer Society, Vienna, 2009, S. 491-494.

Ardagna, Danilo; Pernici, Barbara (2007): Adaptive Service Composition in Flexible Processes. In: IEEE Transaction on Software Engineering, 33 (6), S. 369-384.

Ardagna, Danilo; Mirandola, Raffaella (2010): Per-flow optimal service selection for Web services based processes. In: Journal of Systems and Software, 83 (8), S. 1512-1523.

Arpinar, Budak Ismailcem; Zhang, Ruoyan; Aleman-Meza, Boanerges; Maduko, Angela (2005): Ontology-Driven Web Services Composition Platform. In: Journal of Information Systems and e-Business Management, 3 (2), S. 175-199.

Bächle, Michale; Lehmann, Frank R. (2010): E-Business. Oldenbourg Wissenschaftsverlag, München.

Barros, Alistair P.; Dumas, Marlon (2006): The Rise of Web Service Ecosystems. In: IT Professional, 8 (5), S. 31-37.

Bartalos, Peter; Bieliková, Mariá (2010): QoS Aware Semantic Web Service Composition Approach Considering Pre/Postconditions. In: 2010 IEEE International Conference on Web Services (ICWS'10), S. 345-352.

Berbner, Rainer (2007): Dienstgüteunterstützung für Serviceorientierte Workflows. Dissertation. Universität Darmstadt. Books on Demand, Norderstedt.

Berbner, Rainer; Heckmann, Oliver; Mauthe, Andreas; Steinmetz, Ralf (2005): Eine Dienstgüte unterstützende Web-Service-Architektur für flexible Geschäftsprozesse. In: WIRTSCHAFTSINFORMATIK 47 2005 (4). S. 268-277.

Berbner, Rainer; Spahn, Michael; Repp, Nicolas; Heckmann, Oliver; Steinmetz, Ralf (2006): Heuristics for QoS-aware Web Service Composition. In: Proceedings of the IEEE International Conference on Web Services (ICWS'06), IEEE Computer Society, Chicago, USA, S. 72-82.

Bettag Urban (2001): Web-Services. In: Informatik Spektrum, 24 (5), S. 302–304.

Boehm, Bary W.; Abts, Chris; Brown, A. Winsor; Chulani, Sunita; Clark, Bradford K.; Horowitz, Ellis; Madachy, Ray; Reifer, Donald; Steece, Bert (2000): Software Cost Estimation with COCOMO II. Prentice Hall International, Upper Saddle River, New Jersey.

Braunwarth, Kathrin; Heinrich, Bernd (2008): IT-Service-Management - Ein Modell zur Bestimmung der Folgen von Interoperabilitätsstandards auf die Einbindung externer IT-Dienstleister. WIRTSCHAFTSINFORMATIK 50 2008 (2), S. 98-110.

Canfora, Gerado; Di Penta, Massimiliano; Esposito, Raffaele; Villani, Maria Luisa (2004): A lightweight approach for QoS-aware service composition. In: Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC'04), New York, USA, S. 36–47.

Canfora, Gerado; Di Penta, Massimiliano; Esposito, Raffaele; Villani, Maria Luisa (2005a): An approach for QoS-aware service composition based on genetic algorithms. In: Genetic and Evolutionary Computation Conference (GECCO 2005), Washington DC, USA, S. 1069-1075.

Canfora, Gerado; Di Penta, Massimiliano; Esposito, Raffaele; Villani, Maria Luisa (2005b): QoS-Aware Re-planning of Composite Web Services. In: Proceedings of the International Conference on Web Services (ICWS'05), Orlando, FL, Juli 2005, S. 121–129.

Canfora, Gerado; Di Penta, Massimiliano; Esposito, Raffaele; Perfetto, Francesco; Villani, Maria Luisa (2006): Service Composition (re)Binding Driven by Application-Specific QoS. In: 4th International Conference on Service-Oriented Computing (ICSOC 2006), Chicago, IL, USA, S. 141-152.

Canfora, Gerado; Di Penta, Massimiliano; Esposito, Raffaele; Villani, Maria Luisa (2008): A framework for QoS-aware binding and re-binding of composite web services. In: Journal of Systems and Software, Elsevier Science Inc., 81 (10), S. 1754-1769.

Cardellini, Valeria; Casalichio, Emiliano; Grassi, Vincenzo; Lo Presti, Francesco (2007): Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes. In: Proceedings of the IEEE International Conference on Web Services (ICWS'07), IEEE Computer Society, Salt Lake City, USA, S. 743-750.

Cardoso, Jorge; Miller, John; Sheth, Amit; Arnold, Jonathan; Kochut, Krys (2004): Quality of service for workflows and web service processes, In: Journal of Web Semantics: Science, Services and Agents on the World Wide Web Journal, 1 (3), S. 281-308.

Comuzzi, Marco; Pernici, Barbara (2005): An Architecture for Flexible Web Service QoS Negotiation. In: Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference (EDOC'05), Washington DC, USA, S. 70-82.

Dustdar, Schahram; Schreiner, Wolfgang (2005): A survey on web services composition. In: International Journal of Web and Grid Services, 1 (1), S.1–30.

Fan, Xiao-Qin; Fang, Xian-Wen; Jiang, Chang-Jun (2011): Research on Web service selection based on cooperative evolution. In: Expert Systems with Applications, 38 (8), S 9736-9743.

Faratin, Peyman; Sierra, Carles; Jennings, Nick R. (1998): Negotiation Decision Functions for Autonomous Agents. In: International Journal of Robotics and Autonomous Systems, 24 (3-4), S. 159-182.

Gao, Chunming; Cai, Meiling; Chen, Huowang (2007): QoS-aware Service Composition Based on Tree-Coded Genetic Algorithm. In: Proceedings of the 31st Annual International Computer Software and Application Conference Application Conference (COMPSAC'07), IEEE Computer Society, Peking, China, S. 361–367.

Gao, Yan; Zeng, Bin; Na, Jun; Yang, Lei; Dai, Yu; Gong, Quian (2005): Optimal Selection of Web Services for Composition Based on Interface-Matching and Weighted Multistage Graph. In: Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies (DPCAT'05), IEEE Computer Society, Dalian, China, S. 336-338.

Gao, Yan; Na, Jun; Zhang, Bin; Yang, Lei; Gong, Qiang (2006): Optimal Web Services Selection Using Dynamic Programming. In: Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06), Washington DC, USA, S. 365-370.

Gartner (2011): IT Definitions and Glossary. <http://www.gartner.com/technology/it-glossary/>, Abruf am 2011-07-31.

Georgetown University (2009): IEEE Conference on Commerce and Enterprise Computing CEC '09, Web Service Challenge '09, <http://ws-challenge.georgetown.edu/wsc09/results.html>, Abruf am 2011-07-13.

Gottschalk, K.; Graham, S.; Kreger, H.; Snell, J. (2002): Introduction to Web Services architecture. In: IBM Systems Journal, 41 (2), S. 170–177.

Haddad, Joyce El; Manouvrier, Maude; Ramirez, Guillermo; Rukoz, Marta (2008): QoS-Driven Selection of Web Services for Transactional Composition. In: Proceedings of the IEEE International Conference on Web Services, ICWS'08, S. 653–660.

Haddad, Joyce El; Manouvrier, Maude; Rukoz, Marta (2010): TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition. In: IEEE Transactions on Services Computing, 3 (1), 2010, S. 73–85.

Heinrich, Bernd; Huber, Andreas; Zimmermann, Steffen (2011a): MAKE-AND-SELL OR BUY OF WEB SERVICES A REAL OPTION APPROACH. In: Proceedings of the 19th European Conference on Information Systems (ECIS), Helsinki, Finnland 2011.

Heinrich, Bernd; Klier, Mathias; Zimmermann, Steffen. (2011b): Automatisierte Modellierung, Umsetzung und Ausführung von Prozessen – ein Web Service-basiertes Konzept. In: Bernstein, A., Schwabe, G. (Hrsg.), Proceedings der 10. Internationalen Tagung Wirtschaftsinformatik, 16.-18. Februar 2011, Band 1, Zürich, Schweiz, S. 88-98.

Huang, Angus F.M.; Lan, Ci-Wei; Yang, Stephen J.H. (2009a): An Optimal QoS-Based Web Service Selection Scheme. In: Information Sciences, 179 (19), S. 3309-3322.

Huang, Zhenqiu; Jiang, Wei; Hu, Songlin; Liu, Zhiyong (2009b): Effective Pruning Algorithm for QoS-Aware Service Composition. In: IEEE Conference on Commerce and Enterprise Computing CEC '09, IEEE Computer Society, Wien, S. 519-522.

Jiang, Honghong; Yang, Xiaohu; Yin, Keting; Zhang, Shuai; Cristoforo; Jerry A. (2011): Multi-path QoS-Aware Web Service Composition using Variable Length Chromosome Genetic Algorithm. In: Information Technology Journal, 10 (1), S. 113-119.

Jiang, Wei; Zhang, Charles; Huang, Zhenqiu; Chen, Mingwen; Hu, Songlin; Liu, Zhiyong (2010): Qsynth: A Tool for QoS-Aware Automatic Service Composition. In: Proceedings of the 2010 IEEE International Conference on Web Services ICWS'10, IEEE Computer Society, Miami, S. 42–49.

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

Ko, Andrew J.; Abraham, Robin; Beckwith, Laura; Blackwell, Alan; Burnett, Margaret; Erwig, Martin; Scaffidi, Chris; Lawrance, Joseph; Lieberman, Henry; Myers, Brad; Rosson, Mary Beth; Rothermel, Gregg; Shaw, Mary; Wiedenbeck, Susan (2011): The State of the Art in End-User Software Engineering. In: ACM Computing Surveys, 43 (3), Artikel 21, 44 Seiten.

Ko, Myoung J.; Ouk Kim, Chang; Kwon, Ick-Hyun (2008): Quality-of-service oriented web service composition algorithm and planning architecture. In: *Journal of Systems and Software*, 81 (11), S. 2079-2090.

Kontogiannis, Kostas; Lewis, Grace A.; Smith, Dennis B.; Litoiu, Marin; Müller, Hausi; Schuster, Stefan; Stroulia, Eleni (2007): The Landscape of Service-Oriented Systems: A Research Perspective. In: *Proceedings of the International Workshop on Systems Development in SOA Environments (SDSOA '07)*, S. 1-6.

Kopp, Oliver; Frenkler, Carsten; Lohmann, Niels (2006): Korrektheit und Zuverlässigkeit zusammengesetzter Web Services am Beispiel der Geschäftsprozess-Modellierungssprache BPEL. ftp://ftp.informatik.unistuttgart.de/pub/library/ncstrl.ustuttgart_fi/INPROC-2006-67/INPROC-2006-67.pdf, Abruf am 2011-08-14.

Korkmaz, Turgay; Krunz, Marwan (2001): Multi-Constrained Optimal Path Selection. In: *Proceedings of 20th Joint Conference of IEEE Computer and Communications Societies (INFOCOM'01)*, S. 834–843.

Kossmann, Donald; Leymann, Frank (2004): Web Services. In: *Informatik-Spektrum*, 26 (2), S. 117-128.

Krcmar Helmut (2010): *Informationsmanagement*. 5. Auflage, Springer Verlag, Berlin Heidelberg.

Lassmann, Wolfgang (Hrsg.); Schwarzer, Jens; Rogge, Rolf (2006): *Wirtschaftsinformatik*, Gabler Verlag, Wiesbaden.

Legner, Christine (2009): Is there a market for Web-Services?. In: *ICSOC 2007 Workshops* (Di Nitto, E. and Ripeanu, M. Eds.), Springer, Berlin, Heidelberg, S. 29–42.

Lotz, Volkmar (2007): SOA-Sicherheit für modern Unternehmen. In: *Datenschutz und Datensicherheit – DuD*, 31 (9), S. 644-647.

Ma, Yue; Zhang, Chengwen (2008): Quick convergence of genetic algorithm for QoS-driven web service selection, In: *Computer Networks*, 52 (5), S. 1093-1104.

Maamar, Zakaria; Hacid, Hakim; Huhns, Michael N. (2011): Why Web Services Need Social Networks. In: IEEE Internet Computing, 15 (2), S. 90-94.

Malan, Ruth; Bredemeyer, Dana (2001): Defining Non-Functional Requirements, Technical Report, Bredemeyer Consulting.

Mani, Anbazhagan; Nagarajan, Arun (2002): Understanding quality of service for Web Services, <http://www.ibm.com/developerworks/webservices/library/ws-quality/index.html>, Abruf am 2011-05-22.

Menascé, Daniel A. (2002): QoS Issues in Web Services. In: IEEE Internet Computing, 6 (6), S. 72-75.

Mertens, Peter; Back, Andrea; Becker, Jörg; König, Wolfgang; Krallmann Hermann; Rieger, Bodo; Scheer, August-Wilhelm; Seibt, Dietrich; Stahlknecht, Peter; Strunz, Horst; Thome, Rainer; Wedekind, Hartmut (2001): Lexikon der Wirtschaftsinformatik. Springer, Berlin, Heidelberg.

Motahari-Nezhad, Hamid-Reza; Xu, Guan Yuan; Benatallah, Boualem (2010): Protocol-aware matching of web service interfaces for adapter development. In: Proceedings of the 19th International Conference on World Wide Web (WWW '10), ACM, New York, NY, USA, S. 731–740.

Nüttgens, Markus; Dirik, Iskender (2008): Geschäftsmodelle für dienstebasierte Informationssysteme – Ein strategischer Ansatz zur Vermarktung von Webservices. In: WIRTSCHAFTSINFORMATIK, 50 (1), S. 31–38.

Patel, Chintan; Supekar, Kaustubh; Lee, Yugyung (2003): A QoS Oriented Framework for Adaptive Management of Web Service Based Workflows. In: LNCS (Hrsg.): Database and Expert Systems Applications, Volume 2736, Springer, Berlin, Heidelberg, S. 826-835.

Ran, Shuping (2003): A Model for Web Services Discovery with QoS. In: ACM SIGecom Exchanges, 4 (1), S. 1-10.

Reynolds, Jonathan (2010): E-Business: A Management Perspective. Oxford University Press, New York.

Richter, Jan-Peter; Haller, Harald; Schrey, Peter (2005): Serviceorientierte Architektur. In: Informatik Spektrum, 28 (5), S. 413-416.

Rojas, Ignacio; González, Jesús; Pomares, Héctor; Merelo, J. J.; Castillo, P. A.; Romero, G. (2002): Statistical Analysis of the main parameters involved in the design of a genetic algorithm. In: IEEE Transactions of Systems, Man and Cybernetics Part C: Applications and Reviews, 32 (1), S. 31-37.

Schöneburg, Eberhard; Heinzmann, Frank; Feddersen, Sven (1994): Genetische Algorithmen und Evolutionsstrategien. Addison-Wesley.

SCHUFA Holding AG (2011): <http://www.schufa.de/de/private/home/>, Abruf am 2011-08-25.

Srinivas, M.; Patnaik, Lalit M. (1994): Genetic algorithms - A survey. In: Computer, 27 (6), S. 17-26.

StrikeIron (2011): Frequently Asked Questions. <http://www.strikeiron.com/Support/FAQs.aspx>, Abruf am 2011-08-01.

Strunk, Anja (2010): QoS-Aware Service Composition: A Survey. In: Eighth IEEE European Conference on Web Services, 2010, S. 67-74.

Su, Sen; Zhang, Chengwen; Chen, Junliang (2007): An improved Genetic Algorithm for Web Services Selection. In: Distributed Applications and Interoperable Systems, Lecture Notes in Computer Science, Ausgabe 4531/2007, S. 284-295.

Sun Microsystems, Inc. (2004): Assessing Your SOA Readiness. A Technical White Paper, Juni 2004.

Tang, Maolin; Ai, Lifeng (2010): A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition. In: Tang, Maolin (Hrsg.) Proceeding of the 2010 World Congress on Computational Intelligence, IEEE, Centre de Convencions Internacional de Barcelona, 18.-23. Juli 2010, Barcelona, Spanien, S. 268-275.

Tian, M.; Gramm, A.; Ritter, H.; Schiller, J. (2004): Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework. In: Proceedings of the ACM/IEEE/WIC International Conference on Web Intelligence (WI'04), Peking, China, September 2004, S. 152-158.

University of Georgia Computer Science Department (2002): METEOR-S: Semantic Web Services and Processes, <http://swp.semanticweb.org>, Abruf am 2011-07-11.

W3C Working Group (2003): QoS for Web Services: Requirements and Possible Approaches. W3C Working Group Note, <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>, Abruf am 2011-05-22.

W3C Working Group (2004a): Web Service Architecture Requirements. W3C Working Group Note, <http://www.w3.org/TR/wsa-reqs/>, Abruf am 2011-07-31.

W3C Working Group (2004b): Web Services Glossary. W3C Working Group Note, <http://www.w3.org/TR/ws-gloss/>, Abruf am 2011-07-31.

Wada, Hiroshi; Champrasert, Paskorn; Suzuki, Junichi; Oba, Katsuya (2008): Multiobjective Optimization of SLA-aware Service Compositions. In: Proceedings of the IEEE Congress on Services (Services'08), IEEE Computer Society, Honolulu, Hawaii, USA, 6.-8. Juli, 2008, S. 368-375.

Wan, Changlin; Ullrich, Carsten; Chen, Limin; Huang, Rui; Luo, Jiewen; Shi, Zhongzhi (2008): On Solving QoS-Aware Service Selection Problem with Service Composition. In: Proceedings of the 7th International Conference on Grid and Cooperative Computing (GCC'08), IEEE Computer Society, Shenzhen, China, S. 467-474.

Workflow Management Coalition (WFMC) (1999): Workflow standard - Terminology & glossary. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf, Abruf am 2011-08-07.

Ye, Xinfeng; Mounla, Rami (2008): A Hybrid Approach to QoS-aware Service Composition. In: Proceedings of the IEEE International Conference on Web Services (ICWS'08), IEEE Computer Society, Peking, China, 23.-28. September 2008, S. 62-69.

Yu, Tao; Lin, Kwei-Jay (2005a): A Broker-Based Framework for QoS-Aware Web Service Composition. In: 2nd International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), Hong Kong, China, S. 22-29.

Yu, Tao; Lin, Kwei-Jay (2005b): Service Selection Algorithms for Composing complex Services with Multiple QoS Constraints. In: Proceedings 3rd International Conference on Service-oriented Computing (ICSOC'05), Springer, Amsterdam, Niederlande, 12.-15. Dezember, S. 130-143.

Yu, Tao; Lin, Kwei-Jay (2005c): Service Selection Algorithms for Web Services with End-to-end QoS Constraints. In: Journal of Information Systems and E-Business Management, 3 (2), S. 103-126.

Yu, Tao; Zhang, Yue; Lin, Kwei-Jay (2007): Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints. In: ACM Transactions on the Web, 1 (1), Artikel 6, S. 1-26.

Zeng, Liangzhao; Benatallah, Boualem; Dumas, Marlon; Kalagnanam, Jayant; Sheng, Quan Z. (2003): Quality Driven Web Services Composition. In: 12th International Conference on World Wide Web (WWW 2003), Budapest, Ungarn, S. 411-421.

Zeng, Liangzhao; Benatallah, Boualem; Ngu, Anne H.H.; Dumas, Marlon; Kalagnanam, Jayant; Chang, Henry (2004): QoS-aware Middleware for Web Service composition. In: IEEE Transactions on Software Engineering, 30 (5), S. 311-327.

Zhang, Chengwen; Su, Sen; Chen, Junliang (2006a): A Novel Genetic Algorithm for QoS-Aware Web Services Selection. In: Data Engineering Issues in E-Commerce and Services, Lecture Notes in Computer Science, Ausgabe 4055, Springer Berlin / Heidelberg, S. 224-235.

Zhang, Chengwen; Su, Sen; Chen, Junliang (2006b): Efficient Population Diversity Handling Genetic Algorithm for QoS-Aware Web Services Selection. In: Computational Science – ICCS 2006, Lecture Notes in Computer Science, Ausgabe 3994, Springer Berlin / Heidelberg, S. 104-111.

Zhang, Chengwen; Su, Sen; Chen, Junliang (2007): DiGA: Population diversity handling genetic algorithm for QoS-aware web services selection. In: Computer Communications, 30 (5), S. 1082-1090.