

Methodik und Referenzarchitektur zur
inkrementellen Verbesserung der
Metaqualität einer vertragsgebundenen,
heterogenen und verteilten
Dienstausführung

Dissertation

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

Dipl.-Inf. Josef Spillner
geboren am 08. Mai 1980 in Dresden

Gutachter: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill, TU Dresden
Prof. Dr.-Ing. Ralf Steinmetz, TU Darmstadt

Tag der Verteidigung: 07. September 2010

Dresden im Juli 2010

Danksagung

Die vorliegende Dissertation ist über das behandelte Themengebiet hinaus als komprimiertes Ergebnis einer mehrjährigen Tätigkeit mit einander verwandten, aber dennoch verstreuten wissenschaftlichen Fragestellungen entstanden. Während dieser Zeit konnte ich in zunehmendem Maße einen stetigen Erfahrungsgewinn dazu einsetzen, eine zentrale Problemstellung zu erkennen und eine Lösung als abgeschlossene strukturierte Arbeit zu verfassen.

Mein besonderer Dank gilt dafür meinem Betreuer Herrn Prof. Dr. rer. nat. Dr. h. c. Alexander Schill. Durch seine beständigen inhaltlichen Anregungen und Hinweise zur Darstellung ist die Fertigstellung des Manuskripts in der gegebenen Zeit realisierbar geworden. Überdies hat er einen abwechslungsreichen Einstieg in den akademischen Alltag ermöglicht und den Aufbau hilfreicher Kontakte zu anderen Forschern weltweit ermöglicht.

Auch bei Herrn Prof. Dr.-Ing. Ralf Steinmetz möchte ich mich herzlich bedanken. Die im Rahmen der Zweitbegutachtung und darüber hinaus während der gemeinsamen Projektarbeit generierten Ideen haben für die Entwicklung und Darstellung der Lösung signifikante Akzente setzen können.

Die kollaborative Ausgestaltung der Arbeit ist mit meinen Kollegen im besonderen Maße durchgeführt worden. Insbesondere Frau Dr.-Ing. Iris Braun möchte ich für ihre thematische Einführung in die Servicewelt seit meiner Studienzzeit danken. Meine permanenten Bürokollegen Mesfin Mulugeta, Sandro Reichert, Abrehet Mohammed Omer und Anne Kümpel, aber auch die zahlreichen an wissenschaftlichen und darüber hinaus gehenden Diskussionen interessierten temporären Bürobesucher, haben mich oft inspiriert und unterstützt. Dies gilt auch für die zumeist fernkommunizierend eingebundenen Kollegen aus dem Forschungsprojekt THESEUS/TEXO.

Mein Dank gilt nicht zuletzt denjenigen, die eine Realisierung der entwickelten Konzepte praktisch ermöglicht und so die abstrakten wissenschaftliche Resultate erfahrbar gestaltet haben. Die Beiträge meiner Hiwis Bastian, Jan, Ronny und Christian seien stellvertretend für alle weiteren studentischen Ergebnisse genannt. Neben der internationalen Free-Software-Community haben auch alle *Crowdserver* einen Beitrag geleistet, insbesondere mein DAAD-Student Anton und seine Kommilitonen vom KPI.

Bedanken möchte ich mich schließlich noch bei allen Korrekturlesern und spontanen Hinweisgebern im Verlauf der Erstellung des Dokuments.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Problemdefinition	10
1.2	Wissenschaftliche Beiträge der Dissertation	10
1.3	Aufbau der Arbeit	11
2	Grundlagen und Verwandte Arbeiten	13
2.1	Dienstbeschreibungen und Dienstnutzungsverträge	14
2.1.1	Beschreibung nichtfunktionaler Eigenschaften	15
2.1.2	Vertragliche Zusicherung von nichtfunktionalen Eigenschaften	21
2.1.3	Dienstmodelle und verteilbare Dienste	24
2.1.4	Zusammenfassung Dienstbeschreibungen und Dienstnutzungsverträge	25
2.2	Qualitätsbezogene Prozesse in Dienstumgebungen	26
2.2.1	Rückversicherungsstrategien: Ressourcenreservierung und Adaption	26
2.2.2	Operative Perspektive der Dienstbereitstellung und Dienstnutzung	29
2.2.3	Systemseitige und nutzerseitige Rückmeldungen	30
2.3	Klassifikation und Struktur von NFE	30
2.3.1	Grundlagen der NFE-Modellierung	30
2.3.2	Existierende Ansätze	31
2.3.3	Ableitung struktureller Merkmale	32
2.3.4	Zusammenfassung NFE-Strukturen	33
2.4	Dienst-, Datenqualität und Metaqualität	34
2.4.1	Dienstqualität	34
2.4.2	Erlebnisqualität	35
2.4.3	Datenqualität	35
2.4.4	Metaqualität	36
2.4.5	Zusammenfassung dienstbezogener Qualitätsaspekte	38
2.5	Dienstumgebungen	39
2.5.1	Systemnahe Dienstplattformen	39
2.5.2	Komponentenplattformen	40
2.5.3	Dienstorientierte Architekturen und SOA-Plattformen	40
2.5.4	Dienstplattformen für das Internet der Dienste	42
2.5.5	Zusammenfassung Dienstplattformen	43

3	Methodik zur Qualitätsverbesserung	47
3.1	Inkrementelle Qualitätsverbesserung durch Abgleichverfahren . . .	48
3.2	CPEM - Ein Modell für die Ausdrucksstärke von NFE	49
3.2.1	Expressivitätsstufen in PEM	50
3.2.2	Grafische Notation für CPEM	51
3.2.3	Beispiel für Ausdrucksstärken nach CPEM	53
3.3	Metaqualität von Diensten	54
3.3.1	Motivation	55
3.3.2	Definition	55
3.3.3	Berechnungsvorschrift	58
3.3.4	Beispiel für die Berechnung von SMQ	66
3.3.5	Angabe der Metaqualität in Beschreibungsdokumenten . .	67
3.3.6	Anwendungsfelder	70
3.3.7	Abgrenzung zu anderen Metriken	71
3.4	Verfahren zur inkrementellen Qualitätsverbesserung	72
3.4.1	Gewinnung von Aussagen über NFE aus Monitoringdaten	73
3.4.2	Vorhersage von NFE basierend auf Monitoringdaten . . .	74
3.4.3	Bestimmung der Metaqualität	77
3.4.4	Rückführung und Nutzung der Metaqualität	77
3.4.5	Zusammenfassung der Verfahren	82
3.5	Einbindung der Verfahren in IoS-Plattformen	82
3.5.1	TSM: Modell für handelbare Dienste mit dynamischen Dokumenten	82
3.5.2	Anforderungen an Dienstplattformen	85
3.5.3	Erweiterung von IoS-Plattformen um SMQ-Komponenten	85
4	Plattform für das Internet der Dienste	91
4.1	Konzeption der Architektur	91
4.1.1	Entwurfsrichtlinien	92
4.1.2	Anforderungen an Plattformdienste	93
4.1.3	Prozessabläufe	94
4.2	Umsetzung der Plattformdienste und Komponenten	96
4.2.1	Architektur der Plattformdienste	97
4.2.2	Einbindung der konzipierten Modelle TSM und CPEM . .	104
4.3	Integrierte Architektur und Derivate	105
4.3.1	Systemintegration	105
4.3.2	Plattforminstanzen als Marktplätze	106
4.3.3	Plattformderivate	106
4.4	Umsetzung und Integration der SMQ-Erweiterungen	109
4.4.1	Umsetzung der Dienstabstraktionsschicht SAL	109
4.4.2	Umsetzung des Metadatenkorrelationsdienstes MDCS . .	110
4.4.3	Gesamtsicht auf die erweiterte Plattform	112
5	Validierung von Methodik und Plattform	115
5.1	Anforderungen an die Validierung	116
5.1.1	Skalierungsbetrachtungen	116
5.1.2	Komplexitätsmodell	121
5.2	Experimentierumfeld für die Validierung	123
5.2.1	Dienstportal	124
5.2.2	Hybride Simulation und Simulationsagenten	126

5.2.3	Gesamte Simulationsumgebung	129
5.3	Experimentelle Abläufe und Ergebnisse	130
5.3.1	Ausbau der Grundkonfiguration	130
5.3.2	Durchführung der Evaluierungs-Sprints	131
5.3.3	Durchführung der Simulation	133
5.3.4	Bestimmung und Verbesserung der Metaqualität	134
5.4	Auswertung des Experiments	135
5.4.1	Aussagen zur Korrektheit	138
5.4.2	Aussagen zur Skalierbarkeit	140
6	Zusammenfassung	143
6.1	Verwertungspotenzial	143
6.2	Ausblick	144
A	Ausgewählte Publikationen	159
B	Experimentierumgebung	161

Kapitel 1

Einleitung

Im Internet ist eine ständig steigende Anzahl von Benutzern zu beobachten. Dabei erhöht sich speziell auch die Zahl der Personen mit Breitbandzugängen [63]. Parallel dazu steigt die Vielfalt der über das Internet angebotenen und genutzten Dienstleistungen an, wie aus veröffentlichten Verfügbarkeits- und Nutzungsmetriken von Dienstportalen und verwandten Berichten geschlossen werden kann [111]. Die Einsatzgebiete reichen von der Bürgerkommunikation mit Behörden über die Nutzung freier und kommerzieller Dienste für den privaten Gebrauch bis hin zu vertraglich abgesicherten langfristigen Dienstbeziehungen zwischen Unternehmen.

Klassische Dienstleistungen werden auf all diesen Gebieten zunehmend durch technische Verfahren ergänzt, gestützt oder gar abgelöst. Diese Konvergenz im entstehenden *Internet der Dienste* erfordert leistungsstarke Infrastrukturen zur Erbringung von Diensten mit erkennbaren, garantierbaren und für die Nutzer nachvollziehbaren Eigenschaften [92].

Trotz erkennbarer Verbesserungen der eingesetzten Server- und Netzwerktechnologien treten Fehler und Qualitätsverluste nach wie vor häufig auf, sowohl aufgrund nicht ausgereifter Technologien als auch aufgrund von Fehlern im Betrieb [74]. Als Konsequenz aus diesen Erfahrungen kann man davon ausgehen, dass auch in naher Zukunft eine Erbringung von Dienstleistungen im Internet nicht vollständig fehlerfrei möglich sein wird. Als Reaktion darauf müssen Infrastrukturen zur Dienstbereitstellung mit autonom agierenden Techniken zur Fehlererkennung und -behebung ausgestattet werden. Zudem muss eine Möglichkeit geschaffen werden, dem Nutzer die auftretenden qualitativen Unterschiede transparent zugänglich zu machen.

Mitunter kann ein Verlust an Zuverlässigkeit von Dienstimplementierungen absorbiert werden, indem gegen höhere Kosten eine frühzeitige Ressourcenreservierung und dynamische Anpassung der Ausführung an veränderte Bedingungen durch Adaptionsstrategien stattfindet. Dadurch kann in kritischen Situationen eine nach subjektiver Wahrnehmung gleichgute Dienstgüte erreicht werden. Solche Techniken müssen durch Infrastrukturen aufgenommen werden. Sie sind aber trotz aller Vorausplanungen zur Gewährleistung einer hohen Dienstgüte aufgrund plötzlich auftretender Fehlersituationen nicht genug, wenn etwa Hardware ausfällt oder Systemsoftware Anomalien aufweist. Schwierigkeiten treten auch durch den Einsatz heterogener Diensttechnologien auf, die eine Erkennung qualitativer Eigenschaften und erst recht eine darauf basierende Vorausplanung

nur bedingt ermöglichen. Die systematische plattformseitige Überprüfung der Dienstqualitätsangaben und ein darauf basierendes dynamisches Angebot an umfänglich beschriebenen Diensten für jede angefragte Funktionalität hilft bei der weiteren Reduktion von Risiken. An dieser Stelle setzt die Dissertation mit der Einführung der Metaqualität als Maß für die Güte von Dienstbeschreibungen und als Abstandsmaß zwischen beworbener und tatsächlicher Dienstqualität an.

Schließlich kann es Situationen geben, in denen eine technische Kompensierung nicht mehr möglich ist. In diesen Fällen lassen sich oftmals dennoch die Zufriedenheitswerte der Dienstnutzer halten, indem Rabatte oder andere nicht-technische Anreize in vorher definierten Dienstnutzungsvereinbarungen festgehalten werden. Aus diesem Grund sind an reale Gegebenheiten angepasste Vertragsverwaltungsfunktionen unter Berücksichtigung der Metaqualität zur präzisen Definition von Garantien auf nichtfunktionale Eigenschaften eine weitere notwendige Voraussetzung zur Erbringung von Dienstleistungen im Internet.

Das Ziel dieser Dissertation ist demnach die Entwicklung von Verfahren zur schrittweisen Qualitätsverbesserung in Dienstumgebungen durch Steigerung der Metaqualität und die Realisierung einer solchen Umgebung als modulare Erweiterung einer Basisplattform für das Internet der Dienste.

1.1 Problemdefinition

Aus der vorgestellten Problematik lassen sich zwei zusammenhängende Teilprobleme ableiten:

1. Eine flexible Nutzung elektronischer Dienstleistungen in komplexen Szenarien benötigt nicht nur eine Vielfalt an lose gekoppelten Einzeldiensten mit garantierbaren Eigenschaften, sondern darüber hinaus eine stets aktuell gehaltene, vollständige, abfragbare und unabhängige Bewertung der tatsächlich erbringbaren Dienstgüte. Bisher existiert keine durchgängige Methodik zur Berechnung, Vermittlung und Verbesserung dieser Metaqualitätsdaten.
2. Das Internet der Dienste benötigt leistungsfähige Infrastrukturen für das Anbieten, Suchen, Konfigurieren, Nutzen und Bewerten von Diensten. Eine Umsetzung der Methodik zur inkrementellen Qualitätsverbesserung durch Berücksichtigung der Metaqualität muss dabei ein integraler Teil der Infrastruktur sein. Zur Zeit existiert keine Dienstplattform, die diese Bedingung erfüllt oder durch eine Erweiterung um Metaqualitätsmechanismen erfüllen kann.

1.2 Wissenschaftliche Beiträge der Dissertation

Die vorliegende Arbeit leistet zwei zusammenhängende primäre wissenschaftliche Beiträge, die eine fundierte und praktikable Lösung der definierten Probleme darstellen:

1. Ein Konzept für inkrementelle Qualitätsverbesserungen von Diensten durch Nutzung von Laufzeitmechanismen zur dynamischen Beschreibung und

Bewertung nichtfunktionaler Eigenschaften. Dieses in den Lebenszyklus von Diensten integrierte Konzept sieht einerseits die weitestgehende Nutzung existierender formaler Beschreibungen vor, minimiert jedoch den Aufwand für deren Pflege und erhöht den Grad der Anpassung an reelle Gegebenheiten. Zur präzisen Erkennung vollständiger und korrekter Angaben wird die neue Metrik Metaqualität auf unterschiedlichen Granularitätsstufen eingeführt.

2. Eine Definition eines abstrakten Zusammenspiels von Architekturmustern und Prozessen zur vertraglich abgesicherten Gewährleistung von Dienstgüteeigenschaften in Dienstlandschaften in Form einer Referenzarchitektur. Hierbei werden bekannte Verfahren zur Spezifikation, Aushandlung und Anpassung von Dienstgütekriterien sowie deren objektive Bewertung durch Monitoring und subjektive Bewertung durch Benutzer eingebunden. Gegenüber existierenden Architekturansätzen werden dynamisierte beschreibende Dienstartefakte und somit die Propagierung der Metaqualität in allen Phasen der Dienstnutzung unterstützt.

Desweiteren sind zwei sekundäre Beiträge entstanden, die den bisherigen Stand verwandter Arbeiten signifikant erweitern und die primären wissenschaftlichen Beiträge belegen, unterstützen und nachnutzbar gestalten.

1. Eine modulare, mehrstufig integrierte Dienstplattform als Systemimplementierung zur Demonstration der Anwendbarkeit der erarbeiteten Konzepte. Dabei befinden sich auf der unteren Stufe eine Menge von für sich nutzbaren Komponenten und Plattformdiensten zur Verarbeitung von Dienstartefakten. Diese sind auf höherer Stufe zu einem Gesamtsystem als Basisplattform für das Internet der Dienste zusammengeschlossen. Die Umsetzung der Konzepte zur Metaqualität führt über eine modulare Erweiterung zu einer konkreten hochqualitativen Dienstplattform.
2. Ein prototypischer Dienstmarktplatz im Internet, der als experimentell evaluierter Anwendungsfall der Dienstplattform deren Skalierbarkeit und Wettbewerbsfähigkeit durch Belegung der Korrektheit der Metaqualitätskonzepte im realen Einsatz über einen bestimmten Zeitraum belegt.

1.3 Aufbau der Arbeit

Die Arbeit leistet einen Beitrag zur automatisierbaren, langfristigen qualitativen Verbesserung der Dienstaufführung und Dienstbeschreibung. Sie stellt zuerst Grundlagen und verwandte Arbeiten zur Dienst-, Daten- und Metaqualität vor und analysiert existierende Ansätze zur Beschreibung nichtfunktionaler Dienstmerkmale in Beschreibungen und Verträgen. Ein abstraktes annotierbares Beschreibungsmodell wird dabei basierend auf den Analyseergebnissen als Grundlage für die nachfolgenden Modellierungsansätze der Dissertation extrahiert. Der weitere Text zu den Grundlagen erläutert die Fähigkeiten bereits existierender Infrastrukturen für die Bereitstellung und die Inanspruchnahme von Diensten.

Im Konzeptteil wird einleitend ein generisches Modell für die Repräsentation unterschiedlich expressiver nichtfunktionaler Eigenschaften und deren Metaeigenschaften in allen Phasen des Dienstlebenszyklus erarbeitet. Dies führt zur

Definition der Metaqualität von Diensten. Die zur Erkennung, Propagierung und Verbesserung der Metaqualität notwendigen Verfahren und Berechnungsvorschriften werden in algorithmischer Tiefe ebenfalls im Konzept vorgestellt. Das Kapitel schließt mit einer abstrakten Betrachtung der Einbindung dieser Verfahren in Dienstumgebungen über Plattformschnittstellen und dynamisierte beschreibende Dienstartefakte sowie ein weiteres Modell zur Repräsentation von paketierte Diensten.

Darauf folgt ein Kapitel zur praktischen Umsetzung und Integration der entwickelten Verfahren in eine Dienstplattform. Es beginnt mit einer Diskussion von notwendigen Architekturentscheidungen zur Erbringung einer vertragsgebundenen adaptiven Dienstauführung. Darauf basierend werden die autarken Plattformbestandteile als Komponenten und eine integrierte Dienstplattform sowohl abstrakt als auch als konkret umgesetzte erweiterbare Basisplattform vorgestellt, die schließlich durch eine modulare Ergänzung die Konzepte zur Qualitätsverbesserung einfach und transparent integriert.

Schließlich wird für die Validierung ein Experiment vorgestellt, welches unter Ausnutzung der entwickelten Verfahren zeigt, dass sich durch die Qualitätsverbesserungen die Erlebnisqualität einer Dienstenutzung auch im praktischen Einsatz steigern lässt. Hierfür ist ein spezieller Marktplatz für handelbare Dienste auf Basis der Dienstplattform entwickelt und in einer kontrollierten Experimentierumgebung über längere Zeit beobachtet worden.

Das zusammenfassende Kapitel beinhaltet einen Ausblick auf das Verwertungspotenzial der Forschungsergebnisse und sich ergebende wissenschaftliche Anknüpfungspunkte.

Die nachfolgende Grafik repräsentiert den Aufbau der Dissertation und die möglichen und empfohlenen Pfade einer vollständigen oder interessenfokussierten Abarbeitung.

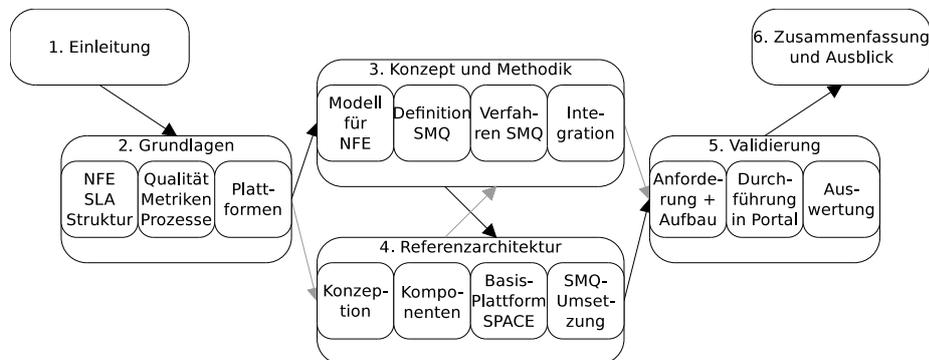


Abbildung 1.1: Kapitelstruktur der Dissertation

Kapitel 2

Grundlagen und Verwandte Arbeiten

Dieses Kapitel gibt einen umfassenden Überblick über verwandte Arbeiten in drei für das Verständnis der Arbeit vorausgesetzten Bereichen: Dienstbeschreibungen, Qualitätsbetrachtungen und Laufzeitplattformen.

Zuerst werden grundlegende Techniken zur Beschreibung von Diensten makroskopisch vorgestellt. Dabei wird auf die Einbindung nichtfunktionaler Eigenschaften (NFE) ebenso Wert gelegt wie auf deren Projektion auf Dienstnutzungsverträge. Darauf folgt eine erweiterte Betrachtung der operativen Verarbeitung von Dokumenten mit Angabe nichtfunktionaler Eigenschaften in allen Phasen der Dienstverarbeitung zur Laufzeit. Davon abgeleitet widmet sich der dritte Abschnitt einer vereinheitlichten Sicht auf Ansätze zum detaillierten strukturellen Aufbau nichtfunktionaler Eigenschaften aus mikroskopischer Perspektive.

Anschließend werden Vorarbeiten und laufende Projekte zur Sicherung der Dienstqualität beschrieben. Diese Betrachtungen werden ergänzt um eine Darstellung allgemeiner Qualitätskriterien auch im Bereich der Daten-, Erlebnis- und Metaqualität.

Schließlich wird durch einen systematischen Vergleich von Dienstplattformen untersucht, inwiefern die bereits bekannten Ergebnisse wie auch die in dieser Arbeit geplanten Erweiterungen bereits umgesetzt sind oder sich zumindest aufwandsarm umsetzen lassen.

Die resultierenden fünf Grundlagenabschnitte sind im Schema 2.1 grafisch zusammenhängend abgebildet.

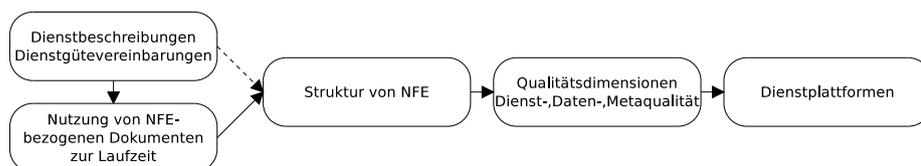


Abbildung 2.1: Unterkapitelstruktur zu Grundlagen und verwandte Arbeiten

2.1 Dienstbeschreibungen und Dienstnutzungsverträge

Der interdisziplinäre Begriff *Dienst* umfasst weit auslegbare Definitionen zur Erbringung von Leistungen ausgehend von einem Anbieter über mögliche Mittler hin zu einem Dienstnehmer. In der Informatik werden darunter elektronische Dienste verstanden. Eine scharfe Abgrenzung zum wirtschaftlich geprägten Dienstleistungsbegriff ist hingegen kaum möglich, da sich beide über hybride IT-gestützte Dienste zunehmend annähern [86, 101] und speziell wissensintensive Dienstleistungen vermehrt als elektronische Dienste angeboten werden [51]. Das Spektrum an elektronischen Diensten reicht von sitzungsorientierten Telekommunikationsdiensten [21] bis zu diskreten Diensten mit Anfrage-Antwort-Funktion. Diese wiederum treten aufgrund der chronologischen Entwicklung und spezialisierter Einsatzdomänen in unterschiedlichen Varianten auf. Beispiele sind entfernte Methodenaufrufe (*Remote Procedure Call*, RPC), lokale Systeme für Interprozesskommunikation und RPC-verbundene Systemdienste wie D-Bus, auf Internetprotokollen aufsetzende *Web Services* und an Web Services angelehnte Dienste zur Kapselung von Benutzerschnittstellen (*UI Services*) [31, 79]. Darüber hinaus existieren hybride Ansätze wie IPC-RPC-Gateways [38] oder stromverarbeitende Web Services [81] und erschweren somit eine Abgrenzung.

Ein schematischer Überblick über Dienste wird in Abbildung 2.2 gegeben. Der für diese Dissertation relevante Begriff verteilter und handelbarer Dienste umfasst alle deklarativ beschriebenen Dienste. Somit sind insbesondere Web Services und elektronisch auffindbare und verwaltbare konventionelle Dienstleistungen Gegenstand der Betrachtungen. Neben den Beschreibungen existieren als weitere Gemeinsamkeiten die Abhängigkeit von technischen, finanziellen und gegenständlichen Ressourcen sowie Abläufe für das Zusammenspiel gemeinsam genutzter Dienste.

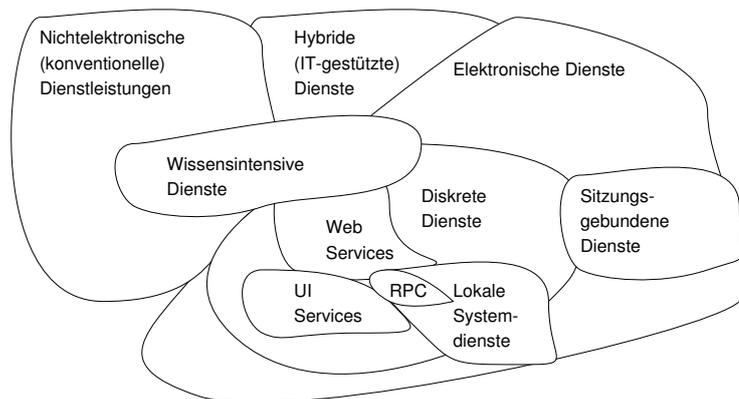


Abbildung 2.2: Dienstbegriffe und Diensttechnologien

Die deklarativen Dienstbeschreibungsdokumente werden primär dazu eingesetzt, die Spezifikation sowohl funktionaler als auch qualitätsbezogener und sonstiger nichtfunktionaler Eigenschaften von Diensten abzubilden. Diese sind zur eigentlichen Kommunikation und Nutzung der Dienstfunktionalität nicht zwingend erforderlich. Für eine flexible Dienstsuche, -nutzung und -komposition

sowie eine evolutionäre Weiterentwicklung, Integration und Verwaltung sind sie hingegen essentiell. Die in den Dienstbeschreibungen hinterlegten Angaben zu Diensteigenschaften werden auf Dienstplattformen in ähnlicher Form noch in weiteren Dokumenten wie etwa vertraglich abgesicherten Dienstgütevereinbarungen zur Kontrolle (Monitoring) und Anpassung (Adaption) der Dienstausführung genutzt.

In den folgenden Abschnitten werden zuerst die Grundlagen und der derzeitige Stand der Forschung zu deklarativen Angaben der Diensteigenschaften vorgestellt. Ein Hauptaugenmerk liegt dabei auf der Darstellbarkeit von Qualitätsmetriken. In der Folge werden weitere Grundlagen zu vertraglichen Zusicherungen nichtfunktionaler Eigenschaften und schließlich zu holistischen Dienstmodellen als Kapselung von Dienstbeschreibungen und Vertragsdokumenten vermittelt.

2.1.1 Beschreibung nichtfunktionaler Eigenschaften

Im Allgemeinen bedingt die Zusicherung einer Garantie, insbesondere einer Qualitätsgarantie, mehr als die bloße Zusicherung der Funktionalität eines Produktes oder einer Dienstleistung. Ein Produkt sollte beispielsweise eine hohe Verarbeitungsqualität, ein gutes Preis-Leistungs-Verhältnis und eine lange Garantiezeit aufweisen. Auch im Bereich der Dienstleistungen wird es ungern gesehen, wenn zwar ein Reparaturauftrag erledigt wird, dieser aber zeitintensiver und teurer wird als ursprünglich geplant. Gerade über diese Eigenschaften können sich Wettbewerber differenzieren, während die eigentliche Funktion oft den Charakter einer beliebigen Austauschbarkeit annimmt. Je stärker die angenommene Austauschbarkeit der Funktion ausgeprägt ist, desto wichtiger werden die nichtfunktionalen Aspekte für eine zielgerichtete Auswahl der besten Angebote aus der Vielfalt an Anbietern [77].

Dienstleistungen, die über das Internet angeboten werden, besitzen heute noch einen hohen Grad an Individualität. Dies betrifft nicht nur die Bandbreite der Angebote und Funktionen, sondern auch die Durchführung der Dienstleistung auf technischer Ebene. Die eingesetzten Übertragungsformate für Informationen, Protokolle und Ablaufsprachen zur Abwicklung unterstützender Prozesse wie etwa der Bezahlung sind hochgradig heterogen. Durch die Einführung standardisierter Schnittstellen und Formate durch Middleware-Ansätze und Web Services wächst jedoch bereits die Zahl der Dienste, die sich sowohl funktional vergleichen lässt als auch technische Kompatibilität aufweist. Dies ermöglicht damit zunehmend eine kurzfristige Auswahl des Diensteanbieters durch den Benutzer oder Kunden. In der Folge steigt damit auch in diesem Bereich die Wichtigkeit deklarativ spezifizierter nichtfunktionaler Eigenschaften (NFE) rapide an.

Dennoch ist nach heutigen Maßstäben für Benutzer von Diensten nur schwer erkennbar, welche NFE ein Dienst tatsächlich hat und inwiefern diese Schwankungen unterworfen sind. Die Zusicherung der NFE durch den Diensteanbieter ist aufgrund jener Schwankungen aus dessen Sicht ebenfalls kaum möglich, ohne häufige und schwer vorhersagbare Verletzungen der Zusicherungen zu riskieren. Das Anliegen dieses Unterkapitels ist daher die Schaffung eines Überblicks über existierende Notationen von NFE als Teil von Dienstbeschreibungen sowie über existierende Mechanismen zu deren Zusicherung. Auch werden bereits vorgeschlagene Erweiterungen und Lösungsansätze zu den vorhandenen Problemen vorgestellt und analysiert.

Beschreibung von Komponenteneigenschaften

Die Eigenschaften eines Dienstes leiten sich zwar aufgrund möglicher Einflüsse der Umgebung wie etwa Laufzeitcontainer nicht vollständig und unmittelbar, aber doch zu großen Teilen von der Implementierung der eigentlichen Funktionen ab. Somit bietet es sich an, das softwaretheoretisch bereits gut erforschte Feld der formalen Beschreibungssprachen für Eigenschaften von Softwarekomponenten zu betrachten.

Allgemein üblich für die abstrakte Darstellung von Komponenten ist die *Unified Modelling Language* (OMG UML, ISO/IEC 19501). Sie bietet eine grafische Notation für die Struktur einschließlich Komposition und hierarchischen oder anderen Beziehungen zwischen Komponenten. Die derzeitige Sprachversion 2.2 wird dabei ergänzt von der *Object Constraint Language* (OCL), mit der sich Eigenschaften von Parametern in Abhängigkeit von anderen Eigenschaften sowie Invarianten ausdrücken lassen.

Als Beispiel für die Modellierung in UML wird nun ein stereotypischer Dienst für die Beurteilung von Projektanträgen vorgestellt. Ein übermittelter Antrag wird zuerst automatisch auf formale Fehler wie fehlende Angaben oder überschrittene Abgabefristen geprüft. In einer zweiten Stufe wird ein eher langwieriger Prüf- und Annahmeentscheidungsprozess durchgeführt. Das Beispiel wird konsistent und vergleichend auch in den anderen Abschnitten verwendet werden. Es weist die interessante Eigenschaft auf, dass bestimmte Eigenschaften wie die Verarbeitungszeit der Eingabenachricht vom Inhalt abhängen und eine vorstellbare Nutzung des Dienstes zwar wiederholt, jedoch nicht ständig erfolgt.

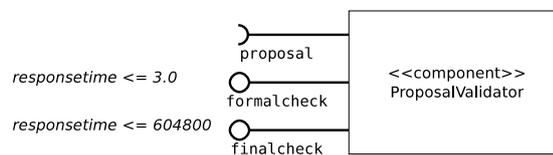


Abbildung 2.3: Beispiel einer Komponentenbeschreibung mit UML und OCL

Eine weitere Beschreibungssprache für Komponenteneigenschaften ist die *Component Quality Modelling Language* (CQML) [131]. Diese ist speziell für die Angabe von diskreten Qualitätsmerkmalen entworfen worden. In ihr sind einfache und komposite NFE typisiert und wertebeschränkt ausdrückbar. Sie stellt damit eine Alternative zu UML/OCL für nichtfunktionale Aspekte dar.

Eine Erweiterung von CQML um systemische Aspekte ist CQML+. Hierbei werden die Komponenten im Kontext eines Laufzeitcontainers betrachtet und die Qualitätsmerkmale an die Verfügbarkeit und Reservierbarkeit von Ressourcen gekoppelt [89]. Desweiteren sind fachliche Verbesserungen an der Sprache selbst wie eine zur Laufzeit maschinenauswertbare Repräsentation mit CQML+ hinzugekommen. Der nachfolgende Codeausschnitt 2.1 enthält ein CQML-Fragment zur rudimentären Spezifikation der Projektantragskomponente mit einer CQML+-spezifischen Ressourcenklausel. Ein Speicherbedarf der Komponente von mindestens 200 Kilobytes wird darüber zur Funktionserbringung erkennbar vorausgesetzt.

Listing 2.1: NFE-Beschreibung in CQML+

```

1  quality_characteristic responsetime {
2     domain: decreasing numeric real seconds;
3  }
4  quality quick(task: Task) {
5     responsetime(task) <= 3.0;
6  }
7  quality deferred(task: Task) {
8     responsetime(task) <= 604800.0;
9  }
10 resource memory {
11     quality_characteristic size (r: Resource) {
12         domain: numeric kilobytes;
13     }
14 }
15 quality memory_high(r: Resource) {
16     size(r).minimum > 200;
17 }
18 component proposalvalidator {
19     uses proposal incoming;
20     provides quick(formalcheck) outgoing;
21     provides deferred(finalcheck) outgoing;
22     resources memory_high(memory);
23 };

```

Erkennbares Verhalten und wechselseitige Einflüsse

Die vorgestellten Sprachen UML/OCL und CQML(+) sind weitestgehend auf die Angabe diskreter Eigenschaften zu jeder Funktion einer Komponente beschränkt. Eine direkte Ableitung von Metriken wie der Antwortzeit abhängig von der übermittelten Datenmenge ist mit ihnen nicht möglich. Gleichermaßen sind die Angaben auf isoliert ablaufende Vorgänge beschränkt, ein wechselseitiger Einfluss paralleler Anfragen oder Störungen durch Umgebungsparameter sind nicht vorgesehen.

Zwar existieren erste Ansätze zur Einbeziehung solcher Einflüsse in semantisch erweiterten Beschreibungssprachen. Diese sollen jedoch in dieser Arbeit nicht weiter betrachtet werden. Stattdessen wird die Annahme getroffen, dass jeder Dienst isoliert arbeitet und nichtfunktionale Eigenschaften unabhängig voneinander sind.

Beschreibung von Diensteigenschaften

Ein Dienst kann nicht nur aus mehreren Komponenten zusammengesetzt sein, sondern auch verteilt unter Nutzung verschiedener Komponententechnologien und weiterer nicht komponentenbasierter Bestandteile arbeiten. Insbesondere bei komplexen Diensten und Prozessen ist dies die Regel. Die Einbindung weiterer unterstützender Subsysteme wie Datenbanken, Replikationsmanager und Kommunikationsmodulen erfordert einen erweiterten Ansatz zur Modellierung, der über Komponenten hinausgeht und dienstspezifische heterogene Architek-

turen und deren Eigenschaften abbilden kann.

Als brauchbar für die architektonische Beschreibung der Dienstarchitektur hat sich dabei die grafische Notation *Fundamental Modelling Concepts* (FMC) herausgestellt, mit der man nicht nur den Aufbau eines Systems darstellen kann. In Anlehnung an Petrinetze lassen sich Abläufe sowie in Anlehnung an Entity-Relationship-Diagramme Wertebereiche formulieren. Insbesondere mehrschichtige Architekturen wie übliche 3-Schichten-Systeme mit einer Trennung zwischen der persistenten Datenspeicherung, der Datenverarbeitung und der Interaktion mit dem Benutzer lassen sich dadurch repräsentieren. Ebenso lässt sich eine Anwendung konzeptuell skalieren, indem ihre Bestandteile wie Datenbank oder Nachrichtenversand redundant ausgelegt und miteinander vernetzt werden. Eine solchermaßen ausgerichtete Architektur des Beispieldienstes zur Antragsprüfung ist im FMC-Blockdiagramm 2.4 dargestellt.

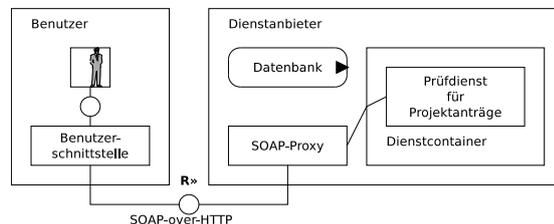


Abbildung 2.4: FMC-Blockdiagramm zur Beschreibung eines Dienstes

Als weiterer Vertreter sei die Systems Modelling Language (SysML) genannt, die derzeit in der Version 1.1 vorliegt und UML um Aspekte für komplexe Systeme erweitert. Mit dieser ebenfalls grafischen Notation ist der Einfluss von Komponenten vom Systemkontext darstellbar.

Weit verbreitet für die rein syntaktische Beschreibung der funktionalen Aspekte von Diensten sind XML-Beschreibungsdateien wie die *Web Service Description Language* (WSDL) für prozedurale Dienste und die *Web Application Description Language* (WADL) für ressourcenzentrische Dienste. Beide enthalten grundsätzlich keine Möglichkeit, nichtfunktionale Angaben vorzunehmen. Es existieren jedoch Erweiterungen wie etwa SAWSDL zur elementweisen Verknüpfung mit Konzepten in Ontologien, die wiederum NFE-Angaben enthalten können. Der nachfolgende Codeauszug 2.2 verdeutlicht den stark vereinfachten Aufbau einer WSDL-Datei. Gut erkennbar ist die notwendige Aufteilung der komplexen Dienstinteraktion auf zwei einzelne Operationen, deren Verknüpfung nicht erkennbar ist. Dies ließe sich über Erweiterungen wie WS-Conversation realisieren.

Als Vertreter der Dienstbeschreibungssprachen aus dem Bereich der semantischen Web Services bietet sich die *Web Service Modelling Language* (WMSL) an. Sie definiert weniger ein abgeschlossenes Beschreibungsmodell als vielmehr eine Syntax mit Basisnotationen zu Eigenschaften, Vor- und Nachbedingungen (IOPEs) sowie Verhaltensweisen von Diensten. Der Beispieldienst zur Überprüfung von Projektanträgen ist nachfolgend im Quelltext 2.3 dargestellt. Erkennbar ist die fehlende Funktions- und Nachrichtenbeschreibung, wofür IOPEs und Grounding-Angaben hinzugefügt werden müssten.

Die *Unified Service Description Language* (USDL) ist ein Ansatz, neben technischen auch operative, juristische und geschäftliche Dienstmerkmale aus-

Listing 2.2: NFE-Beschreibung in WSDL (Pseudocode-Notation)

```

1 service proposalvalidator {
2   port default {
3     operation validatequick {
4       message incoming proposal;
5       message outgoing formalcheck;
6     }
7     operation validatedeferred {
8       message outgoing finalcheck;
9     }
10  }
11 }

```

Listing 2.3: NFE-Beschreibung in WSMML

```

1 wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-
   flight"
2 namespace { _"http://smp/ontologies/Project/
   ProposalValidator.wsml#",
3   qos _"http://smp/ontologies/QoSBase.wsml#",
4   remoteqos _"http://smp/ontologies/RemoteQoSBase.wsml#",
5   wsml _"http://www.wsmo.org/wsml/wsml-syntax#" }
6
7 webService ProposalValidator
8   importsOntology _"http://smp/ontologies/RemoteQoSBase.
   wsml#"
9
10 capability ServiceCapability
11
12 postcondition
13   definedBy ?serviceType memberOf remoteqos#RemoteQoS.
14
15 interface Basic
16   importsOntology _"http://smp/ontologies/Project/
   ProposalValidator.wsml#Basic"
17
18 ontology Basic
19   importsOntology _"http://smp/ontologies/RemoteQoSBase.
   wsml#"
20
21 instance RT memberOf { remoteqos#ResponseTime, qos#
   ServiceSpec }
22   qos#value hasValue 10
23   qos#unit hasValue qos#Second

```

Listing 2.4: NFE-Beschreibung in USDL (Pseudocode-Notation)

```

1  service {
2    key "B00F165E-C993-60FA-DBB2-A7A340201C5E"
3    type "Atomic"
4    name "ProposalValidator"
5    serviceLevel {
6      invocationTime "3.0"
7    }
8    priceModels {...}
9    operational {
10     operations {
11      operation {
12        name "validatequick"
13        inputInterface {...}
14        outputInterface {...}
15      }
16    }
17  }
18  technical {...}
19 }

```

zudrücken [17]. Vor dem Hintergrund einer erhöhten Akzeptanz von Internetdiensten ist dies eine wichtige konzeptionelle Erweiterung. Im Gegensatz zu WSDL lassen sich in USDL Preismodelle, Qualitätsstufen, Dienstzugangsoptionen, Kontaktdaten und allgemeine Nutzungsbedingungen formulieren. Allerdings weist USDL Schwächen auf, indem Strukturmerkmale von WSDL dupliziert anstatt referenziert werden. Auch die strukturierte Angabe komplexer oder annotierter nichtfunktionaler Eigenschaften ist nicht vorgesehen. Das folgende Syntaxbeispiel 2.4 in USDL Version 1 wurde auf wenige ausgewählte Merkmale reduziert und als kompakte Variante dargestellt. Die Granularität der NFE-Angaben ist erkennbar nicht auf die Präsenz mehrerer Operationen ausgerichtet, sondern bezieht sich auf den gesamten Dienst. Das laufende Standardisierungsverfahren zu USDL Version 3 deutet derzeit nicht auf eine besondere Berücksichtigung flexibler NFE-Modellierung hin ¹.

Schlussfolgerungen

Die in der Tabelle 2.1 zusammengefasste Betrachtung der verschiedenen Ansätze zur Dienstbeschreibung hat gezeigt, dass es kein singuläres Format zur Angabe aller Diensteseigenschaften und -verhaltensweisen gibt. Insbesondere wird die Praxis nach wie vor durch rein syntaktische Formate wie WSDL dominiert, während sich weitergehende Beschreibungen unter Einbeziehung von Ontologien und über technische Aspekte hinausgehende Angaben zu geschäftlichen und operativen Eigenschaften noch nicht über Forschungsprototypen hinaus durchsetzen konnten. Dieser Situation werden die in dieser Arbeit entwickelten Konzepte gerecht werden müssen. Zur Maximierung der Anwendbarkeit auf existierende Dienste wird daher vorgeschlagen, die Diensteseigenschaften von ihrer konkreten

¹USDL-Spezifikation: <http://internet-of-services.com>

Tabelle 2.1: Merkmale von Dienstbeschreibungssprachen

Sprache und Notation	NFE-Angaben
UML/OCL (grafisch)	minimal definierbar, aber im Dienstumfeld nicht unterstützt
CQML(+) (kompakt, XML)	flexibel definierbar, aber im Dienstumfeld nicht unterstützt
FMC (grafisch)	keine
WSDL (XML)	nur über SAWSDL-Erweiterungen
WADL (XML)	keine
WSML (kompakt)	flexibel definierbar
USDL (XML)	wenige vordefinierte NFE

Darstellung in Dienstbeschreibungen über Abbildungsvorschriften zu abstrahieren.

2.1.2 Vertragliche Zusicherung von nichtfunktionalen Eigenschaften

Zwar werden deklarative Dienstbeschreibungen unter Entwicklern häufig als Kontrakt bezeichnet, der die zu nutzenden Schnittstellen festlegt und im Sinne einer modellgetriebenen Softwareentwicklung manifestiert und dokumentiert, aber im juristischen Sinne sind dies keine rechtsgültigen Dokumente.

Möchte sich ein Benutzer bestimmte Eigenschaften der Dienstauführung garantieren lassen, so ist dafür ein gesondertes Dokument notwendig, welches die Vereinbarungen zur Dienstgüte (*Service Level Agreement*, SLA) enthält. Üblicherweise beziehen sich die darin beschriebenen Garantien neben einer stets zu erbringenden Funktion auf verhandelbare NFE, deren Verhandlungsspielraum aus der Dienstbeschreibung oder einer speziellen SLA-Vorlage (*Service Level Agreement Template*, SLAT) ersichtlich ist. Eine SLA als technisches Kerndokument bildet ergänzt um relevante juristische Aspekte dann einen Vertrag. In diesem werden die zeitliche und räumliche Gültigkeit eingeschränkt, der Gerichtsstandort festgelegt sowie weitere wohlbekanntere Dienstnutzungsregeln (*Terms of Service*, ToS) festgelegt.

In diesem Unterkapitel werden relevante Sprachen für elektronische Verträge und Vertragsabschlüsse vorgestellt. Die Bezeichnungen Vertrag und technische Dienstgütevereinbarungen (SLA) werden, sofern nicht aus juristischen Gründen von gesondertem Interesse, synonym verwendet.

Aufbau von Dienstgütevereinbarungen

Die Struktur von Dienstgütevereinbarungen richtet sich primär nach den beteiligten Teilnehmern. Üblicherweise werden SLA zwischen einem Dienstanbieter

und einem Dienstanbieter betrachtet. Jedoch sind sie auch zwischen Dienstleistungsanbieter und Dienstentwickler gängig, um etwa dem Entwickler einen Teil des erzielten Einkommens zukommen zu lassen. Auf technischer Ebene sind weitere SLAs zwischen verteilten Systemkomponenten wie einer Dienstverwaltung und einem Ausführungsserver definierbar. Seltener sind Konstellationen, in denen SLA zwischen mehr als zwei Teilnehmern existieren.

Während viele SLA individuell gestaltet und somit zwar flexibel, aber nur bedingt maschinell auswertbar sind, sind mit dem Aufkommen von Gridsystemen und Web Services diverse Sprachen entwickelt worden, die eine einheitliche deklarative Notation von Vereinbarungen ermöglichen. Aufgrund ihrer hohen Verbreitung in der Literatur und konzeptionellen Spannweite seien hier die Sprachen WSLA [29], WS-Agreement [28] und SLAng [61] vorgestellt. Neben diesen Auszeichnungssprachen existieren vorlagenbasierte Ansätze, die zumindest hinsichtlich der Änderung von ausgehandelten Werten flexibel sind, jedoch strukturell keine Modifikationen zulassen.

Ein Vertrag gemäß der *Web Services Agreement Specification* (WS-Agreement) besteht aus einer Menge von allgemeinen Metadaten zu den Vertragspartnern und zum Vertrag selbst, sowie aus einer weiteren Menge von einzelnen Dienstgütezielen, auch *Service Level Objective* (SLO) oder *Key Performance Indicator* (KPI) genannt. Die konkrete Syntax der SLOs wird dabei durch WS-Agreement nicht vorgegeben, wodurch jede Implementierung eine Formatverfeinerung voraussetzen muss. Als brauchbar hat sich dabei das Format WS-Agreement+ValueTypes herausgestellt, in welchem jede SLO einen gerichteten und einheitennormierten Grenzwert oder als Vorlage einen Wertebereich besitzt. Eine alternative Spezialisierung für dynamisch aushandelbare SLO existiert für Verträge zwischen virtuellen Organisationen in Gridumgebungen [49].

Web Service Level Agreement (WSLA) ist ein älterer Ansatz, der sich aus einer Sprachspezifikation [29] und einer operativen Einbindung [50] zusammensetzt. Metriken können analog zu Typen in XML Schema definiert werden, wobei dafür proprietäre Sprachkonstrukte benötigt werden.

Ein ausführlicher Vergleich von WS-Agreement und WSLA findet sich in [83].

SLAng ist gegenüber WS-Agreement und WSLA stärker an eine geschichtete Systemarchitektur angelehnt [61]. Vertikal werden die SLO somit von der Netzwerkschicht über Container und Hostingumgebungen bis hin zur Anwendung betrachtet. Horizontal wird die Komposition von Diensten berücksichtigt. Nachteilig für die Verwendung von SLAng in Dienstlandschaften ist eine fehlende syntaktische Spezifikation.

Die Tabelle 2.2 fasst die Merkmale üblicher SLA-Sprachen anhand syntaktischer und operativer Kriterien zusammen.

Kein bisher spezifiziertes SLA-Format deckt alle Anforderungen ab. Beispielfähig seien hier nur der Aufbau und die Schwächen von WS-Agreement aufgeführt. Diese Spezifikation ist fest an einen bestimmten Aushandlungsalgorithmus gekoppelt und kennt keine Neuaushandlung. Die SLO-Angaben unterliegen keiner formalen Semantik. Ausführliche Analysen sind an anderer Stelle bereits durchgeführt worden ([68, 37]). Da im Rahmen dieser Arbeit keine eigene SLA-Sprache entworfen werden soll, sind die aufgezeigten Mängel im Entwurf der Lösung zu berücksichtigen.

Tabelle 2.2: Merkmale von SLA-Sprachen

Sprache und Notation	Typsystem	Präferenzen	Einschränkungen	Aushandlung	Monitoring
WS-Agreement (XML)	nicht definiert	ja	Abschnitt für Provider Constraints	fest gekoppelt an WS-Negotiation	nicht definiert
WSLA (XML)	angelehnt an XML Schema	nein	Abschnitt für Obligations	nicht definiert	definiert über WSLA-Framework
SLAng (abstrakt)	eigenes, MOF-basiert (Real, String, TimeUnit, Percentage etc.)	nein	in OCL formulierbar	nicht definiert	wohldefiniert

Aufbau von rechtsgültigen Verträgen

Die Akzeptanz qualitätswahrender Dienstanbieter im Internet der Dienste hängt wesentlich davon ab, inwiefern die gegebenen Garantien im Ernstfall einforderbar und gegebenenfalls sogar einklagbar sind. Diese Anforderung bedingt eine Einbindung der geschlossenen Vereinbarungen in rechtsgültige Verträge. Die rechtlichen Rahmenbedingungen dafür unterscheiden sich von Land zu Land und sind abhängig vom Kontext der Vereinbarung wie einer Geschäftsbeziehung oder einer Privatnutzung. Aus diesem Grund sind die hier dargelegten Ansätze auf die Bundesrepublik Deutschland beschränkt.

Aufgrund der teilweise diffusen rechtlichen Situation wurde speziell für die vorliegende Arbeit eine entwicklungsbegleitende Begutachtung durch juristische Projektpartner an der Universität Karlsruhe (TH) angefragt und durchgeführt [121]. Das grundlegende Verständnis eines Vertragsdokuments zwischen Dienstanbieter und Dienstnutzer ist dabei wie in Abbildung 2.5 dargestellt eine Komposition aus Allgemeinen Geschäftsbedingungen, einer Leistungsbeschreibung sowie aushandelbaren Parametern, die rechtlich eine Leistungseinschränkung bedeuten.

Überwachung der Einhaltung von Verträgen

Nach der Einführung in Dienstgütevereinbarungen und das Vertragswesen im Dienstleistungsbereich werden nun existierende Ansätze für die Überwachung von Verträgen dargestellt. Weitere vertragsgebundene Verwaltungsverfahren wie SLA-Aushandlungen werden für die vorliegende Arbeit nicht betrachtet. Ein Überblick wird in [68] gegeben. Spezielle Verfahren existieren für die Voraushandlung [11] und die eigentliche Aushandlung von Verträgen [126].

Ist ein Vertrag zustande gekommen, so sollte dessen Einhaltung möglichst

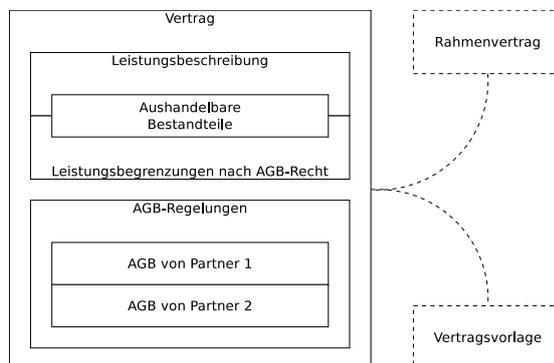


Abbildung 2.5: Aufbau eines rechtsgültigen Vertragsdokuments

vollständig autonom überwacht werden, um strittige Auffassungen von der Erfüllung der vertraglich festgelegten Leistungen zu minimieren. Auf technischer Ebene gibt es dazu eine Vielzahl von Verfahren, die ausgehend von einem Vertrag eine Überwachung mittels Monitoring ermöglichen. Dazu müssen NFE in Dienstbeschreibungen, Vertragsvorlagen und aktiven Verträgen auf systemseitig verfügbare Mess- und Aggregationsfunktionen abgebildet werden. Bislang existiert keine SLA-bezogene Sprache zur Realisierung dieser Anforderung. Neben der Interpretation der SLA zur Laufzeit existieren Verfahren zur Instrumentierung von Dienstimplementierungen oder Laufzeitumgebungen mit Monitoring-Code wie im CQML+-basierten Ansatz von COMQUAD [90] und im BPEL-basierten Prozessmonitoring [119]. Allen Ansätzen ist gemeinsam, dass bei Eintritt einer SLA-Verletzung korrigierende Aktionen wie eine Adaption der Ausführung vorgenommen werden.

Schlussfolgerungen

Die Absicherung der Dienstauführung durch Dienstgütevereinbarungen und Verträge ist einhergehend mit einer zunehmenden öffentlichen Bereitstellung von Diensten ein gut untersuchtes Gebiet. Die Aushandlung und Spezifikation der Vereinbarungen ist mit existierenden Ansätzen möglich. Allerdings sind trotz bisheriger speziell darauf ausgerichteter Forschungsprojekte verschiedene Aspekte der Vertragsgestaltung nur unzureichend beleuchtet worden. Dies betrifft insbesondere die Bereitstellung von Vertragsvorlagen, deren Angaben zu Diensteigenschaften im Kontext einer automatisierten Aushandlung unbedingt die tatsächlich erbringbare Leistung widerspiegeln müssen. Davon abgeleitet wird eine Betrachtung von Verträgen und Vertragsvorlagen als dynamische Dokumente mit einem wohldefinierten Prozess zur Anpassung benötigt.

2.1.3 Dienstmodelle und verteilbare Dienste

Ein bisher wenig untersuchter Bereich ist die gesamtheitliche Modellierung, Entwicklung und Verbreitung von Diensten.

Die Modellierung von abstrakten Geschäftsprozessen, deren Transformation auf ausführbare Prozesse und die Nutzung von Regelsprachen zur Reaktion auf Ereignisse während der Prozessausführung sind bisher weitestgehend entkoppel-

te Themenfelder. Eine kombinierte Betrachtung findet erst in jüngster Zeit statt [129]. Vereinzelt werden auch Aspekte wie der Übergang von der Idee zu einem Dienst zu seiner Realisierung [87] und die Verbindung eines Dienstes zu einem passenden Geschäfts- und Preismodell [40, 62] auf Basis von Ontologien in die Modellierung eingebunden.

Eine weitergehende Verknüpfung zwischen Modellen und Abbildungen auf verschiedenen Ebenen einer technischen und einer rollenbezogenen Dimension bietet die Methodologie zur Integrierten Dienstentwicklung (*Integrated Service Engineering*, ISE). Die ISE-Methodologie ist noch nicht abschließend fertig gestellt, eine Evaluierung wird derzeit anhand der dafür eigens auf Eclipse-Basis entwickelten Entwicklungsumgebung *ISE Workbench* [91] durchgeführt.

Um elektronische Dienstleistungen greifbar (*tangible*) und handelbar zu machen, müssen sie ähnlich wie Produkte von immateriellen zu materiellen Gütern umgestaltet werden. Technisch geschieht dies durch eine Paketierung des Dienstes für eine bestimmte Zielumgebung und anschließend durch den eigentlichen Handelsvorgang. Je nach Geschäftsmodell kann dafür eine Replikation des Dienstpaketes vonnöten sein. Derartig verteil- und handelbare Dienste repräsentieren nach weit verbreiteter Ansicht ein Kernelement für das Internet der Dienste.

Ein auf die ISE-Methodologie passendes Paketformat stellt das Service Archive (SAR) dar. In ihm werden die Dienstimplementierung als ausführbarer Prozess, die Beschreibungen und Vertragsvorlagen dieses Dienstes sowie die Beschreibungen von ihm genutzter Dienste abgelegt. Als Dienstbeschreibung kommt hierbei USDL zum Einsatz. Aufgrund der starren Spezifikation und starren Kopplung an USDL ist SAR für den Einsatz in heterogenen Umgebungen wie dem partizipativen Internet der Dienste nur bedingt geeignet.

Herkömmliche Paketformate, die durch die Auslieferung von Diensten als Teil von Softwaremodulen oder Webanwendungen um serviceorientierte Merkmale ergänzt worden sind, sind noch weniger geeignet, da der deterministische Zugriff auf Dienstbeschreibungen oder gar weitere Dokumente wie SLA-Vorlagen nicht möglich ist.

Aufgrund fehlender akzeptierter Ansätze zur Verbreitung von Diensten wird in dieser Arbeit die Einführung eines Modells für verteilbare Dienste bei größtmöglicher Beibehaltung existierender Paketformate als notwendiger Schritt gesehen. Die Modellierung und Entwicklung der Dienste wird nicht als Problematik betrachtet.

2.1.4 Zusammenfassung Dienstbeschreibungen und Dienstnutzungsverträge

Die Untersuchung der existierenden Ansätze zum Aufbau von Dienstbeschreibungen und Dienstnutzungsverträgen und zu deren Handhabung in Dienstplattformen hat etliche Mängel aufgezeigt, die an dieser Stelle noch einmal zusammengefasst werden sollen.

Es existieren nur wenige Beschreibungsformate, um Diensteigenschaften und darauf bezogene Garantierterme präzise und umfänglich formulieren zu können. Die darauf spezialisierten Sprachen wie CQML+ sind mit den in SOA-Umgebungen durchgängig unterstützten Formaten wie WSDL disjunkt. Soll diese Diskrepanz nicht in einem weiteren Format resultieren, so ist es notwendig,

die Verarbeitung von Teilaspekten der Beschreibungsdokumente wie die Auswertung nichtfunktionaler Eigenschaften zu abstrahieren.

Es fehlt zudem an Konzepten für dynamische Dokumente als Grundlage einer vollständig adaptiven Dienstumgebung. Sowohl für den Dienstentwickler als auch systemseitig muss es möglich sein, dienstbezogene Beschreibungen zu aktualisieren. Statisch abgelegte oder aus der Implementierung heraus generierte Dienstbeschreibungen reduzieren die Flexibilität von SOA-Umgebungen.

Schließlich existieren nur wenige Kontrollmechanismen, die über Rückkopplungsverfahren die spezifizierten Eigenschaften überprüfen und korrigieren können. Hierfür ist es notwendig, eine durchgängige Betrachtung der Diskrepanz zwischen angegebenen Werten und tatsächlich auftretenden Messgrößen durch die Metrik Metaqualität zu ermöglichen.

2.2 Qualitätsbezogene Prozesse in Dienstumgebungen

Nach der Darstellung der statischen Repräsentation von Diensteigenschaften und vertraglichen Zusicherungen soll die Betrachtung nichtfunktionaler Eigenschaften über den gesamten Lebenszyklus des Dienstes ausgeweitet werden. Es ergibt sich daraus die Anforderung der uniformen Repräsentation von NFE auch in Anfragesprachen, Dienstbewertungen und Monitoringergebnissen.

2.2.1 Rückversicherungsstrategien: Ressourcenreservierung und Adaption

Die Risikominimierung bei Anbietern von Versicherungen gegen Diebstahl oder Blitzschlag führte zur Bildung von Rückversicherern, die im Falle von Naturkatastrophen kurzfristig höhere Geldbeträge an die Versicherungen auszahlen und diesen damit überhaupt erst ermöglichen, ihre Beiträge und Leistungen über statistisch nicht vollständig erfassbare Modelle wie dem Wetter zu kalkulieren.

Analog dazu muss sich auch ein Dienstanbieter seinerseits rückversichern, damit nicht im Ausnahmefall zu hohe Kosten auf ihn zukommen. Eine rein monetäre Rückversicherung federt zwar die Kosten ab, liefert aber keinen Beitrag zur Kontinuität hochqualitativer Dienstbereitstellung, was sich sekundär in Form abwandernder Nutzer auswirken kann. Hierfür gibt es verschiedene Lösungsstrategien, die autonom agieren und Ausfälle oder drohende Qualitätsverluste im System ausgleichen können. Die bedeutendsten Ansätze sind die vorherige Reservierung einer adäquaten Menge an Ressourcen im System sowie die nachträgliche Anpassung von Ressourcen, Diensten oder Verträgen an veränderte Verhältnisse. Diese Differenzierung zwischen pessimistischer und optimistischer Voraussicht ist aus anderen Bereichen wie etwa der Sperrung von Datensätzen für den konkurrierenden Zugriff in Datenbanksystemen bekannt [46]. Im Rahmen der Bereitstellung von Diensten muss ein Kompromiss gefunden werden, der einerseits der kostspieligen Überprovisionierung vorbeugt, andererseits hingegen eine Grundsicherheit gegenüber Ausnahmesituationen bietet.

Ressourcenreservierung

Die von einem Dienst erforderlichen Ressourcen können sehr unterschiedlicher Natur sein. Dazu gehören die von einem Betriebssystem verwalteten Betriebsmittel wie Rechenzeit, Arbeitsspeicher oder persistenter Datenspeicher ebenso wie verteilt laufende Dienste.

Wird ein neuer Dienst in Betrieb genommen, so müssen Ressourcen entsprechend den Anforderungen des Dienstes reserviert werden. Dies bedingt die schichtenübergreifende Unterstützung solcher Mechanismen bis hin zum Betriebssystemkern.

Adaption des Systems

Ist keine ausreichende Ressourcenreservierung vor Beginn der Dienstinutzung vorgenommen worden oder ist ein veränderter Ressourcenbedarf erkennbar, so kann eine dynamische Anpassung der verfügbaren Ressourcen auf Systemebene zur adaptiven Dienstaufführung beitragen. Es existieren bereits Ansätze zur lastgesteuerten dynamischen Ressourcenreservierung in Cloud-Systemen [110]. Optimierte Verfahren unter Einbeziehung vertraglich zugesicherter Diensteseigenschaften sind jedoch bisher nicht bekannt.

Adaption von Diensten

Ist die Skalierung des Systems an die Grenzen geraten oder ist eine weitere Skalierung ökonomisch oder technisch nicht sinnvoll, so bietet es sich an, stattdessen den Ressourcenbedarf der Dienste zu senken. Hierbei muss zwischen der statischen Änderung der Dienstkonfiguration (*offline*) einerseits und der dynamischen Anpassung laufender Dienstinstanzen (*online*) andererseits unterschieden werden. Auch spielt der Grad der Anpassbarkeit von Diensten eine Rolle. In Dienstkompositionen können die genutzten Einzeldienste durch Neubindung an funktional äquivalente Alternativen (*Rebinding*) ausgetauscht werden. Sowohl in Kompositionen als auch in atomaren Diensten lassen sich mitunter über den Laufzeitcontainer Parameter rekonfigurieren.

Der Adaptionmechanismus Neubindung wird speziell auf Dienste angewendet, die ihrerseits wieder Web Services aufrufen. So können die an die Aufrufe gebundenen Dienste durch andere funktional kompatible, aber nichtfunktional eher den geänderten Anforderungen entsprechende Alternativen ersetzt werden. Die durch die darauf folgenden Aufrufe betroffenen NFE wie Verarbeitungszeit oder Größe von Nachrichten wirken sich dabei auf den aufrufenden Dienst aus und können das System entlasten.

Der Adaptionmechanismus Rekonfiguration betrachtet generell die Änderung der Konfiguration von Diensten und Dienstinstanzen durch dafür vorgesehene Schnittstellen wie Konfigurationsdateien, Umgebungsvariablen oder Aufrufparameter. Da die Dienstimplementierungen nur selten direkt auf die Ausnutzung von Ressourcen hin konfigurierbar sind, sondern die Ausnutzung sich eher indirekt als Folge von domänenspezifischen Konfigurationsparametern ergibt, versteht man speziell unter Rekonfiguration auch die Abbildung von Domänenparametern auf Systemparameter im Kontext jeder Anpassung der Konfiguration eines Dienstes.

Adaption von Verträgen

Bei Erreichen der Flexibilitätsgrenze während der Adaption des Systems und der Dienste wird im Falle einer erneut drohenden Vertragsverletzung die Wahrscheinlichkeit einer tatsächlichen Verletzung sehr hoch sein. In einem solchen Fall ist es unter Beachtung von zeitlichen und juristischen Einschränkungen möglich, die Verträge über die Dienstnutzung selbst unter Zustimmung der Benutzer zu modifizieren. Dies wird durch eine Neuaushandlung des betroffenen Vertrags erreicht. Im Fall einer fortwährenden Verletzung kann es hilfreich sein, stattdessen langfristig die Vertragsbedingungen konservativer zu gestalten. Dies wird durch eine Anpassung der für die Aushandlung genutzten Vertragsvorlagen erreicht.

Beiträge der Adaptionsverfahren zur Qualitätsverbesserung

Der Übergang von einer erkannten Garantieverletzung zu einer Behebung ist in den vorgestellten SLA-Sprachen wie WSLA oder WS-Agreement unterspezifiziert. Zur Steuerung der Adaption von Dienstkompositionen existiert daher als Ergänzung die Sprache WS-Re2Policy [85]. Die von den Bedingungen eines WS-Re2Policy-Dokuments akzeptierten Ereignisse lösen abhängig vom Grad der Abweichung korrigierende Reaktionen wie Neubindung, Neuplanung, Neuaushandlung oder Rekonfiguration aus. Ereignisse repräsentieren dabei QoS-Messwerte, die in einem eigenen Format spezifiziert werden. Als passive Reaktion kann ein Messwert aufgezeichnet werden. Eine Umsetzung existiert als Framework *AMAS.KOM*. Die Spezifikation und die Umsetzung sehen keine langfristige Qualitätsverbesserung vor, der Fokus liegt klar auf einmalig ausgelösten Adaptionen.

Schlussfolgerungen

Durch die Nutzung von Rückversicherungsstrategien erhält der Dienstanbieter einen Freiraum zur Optimierung der Gestaltung von Verträgen. Dieser Freiraum kann bei Ressourcenreservierung unter Umständen präzise berechnet werden, bei einer Einplanung von Adaptionsmechanismen kann er nur heuristisch bestimmt werden. Der Freiraum erstreckt sich dabei generell nur auf NFE, die unter der unmittelbaren Kontrolle des Anbieters stehen. Üblicherweise schließt dies die Qualität der Netzwerkverbindung zwischen Anbieter und Dienstanutzer aus. Um den Freiraum zu maximieren und wechselseitige Beeinflussungen und Neutralisierungen auszuschließen, ist eine koordinierte Adaption unter Einbeziehung aller Adaptionsstrategien, -mechanismen und -ziele erforderlich. Diese erfordert hochqualitative Beschreibungen der Qualitätseigenschaften der betroffenen Dienste.

Die Abbildung 2.6 zeigt die Relation zwischen dem Erwartungswert E einer NFE und dem garantierbaren Wert B . Dabei wird B aus betriebswirtschaftlichen Gründen meist niedriger als E angesetzt, um häufige Vertragsverletzungen zu vermeiden. Ist die Dienstauführung adaptiv gestaltet, so kann der Puffer durch Verschiebung von B auf M heuristisch vergrößert werden. Je präziser die NFE-Angaben gestaltet sind, desto sicherer kann eine Vergrößerung erfolgen. Dem Anbieter entstehen somit geringere Kosten. Bei einer durch Ressourcenreservierung erreichten vollständigen Qualitätsgarantie sind alle drei Größen M , B und E identisch.

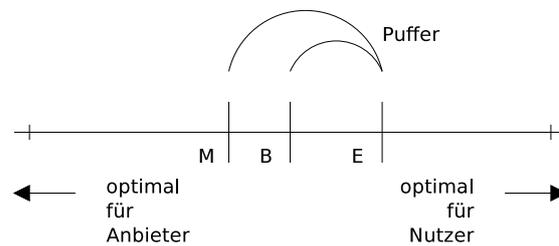


Abbildung 2.6: Heuristische Angabe von NFE in Dienstbeschreibungsdokumenten und Vertragsvorlagen

2.2.2 Operative Perspektive der Dienstbereitstellung und Dienstnutzung

In einem funktionierenden Ökosystem aus Diensteanbietern und Dienstnutzern ist es unerlässlich, dass standardisierte Vorgänge zum Auffinden, Konfigurieren und Nutzen von Diensten existieren. Die Steuerung der Vorgänge wie Registrierung, Dienstsuche, Vertragserzeugung und Nutzungsanalyse wird dabei zumeist zentral über einen Anbieter oder einen Marktplatz mit mehreren Anbietern durchgeführt. In diesem Unterkapitel werden die wichtigsten Vorgänge aus Sicht der Anbieter und Nutzer vorgestellt. Für jeden Vorgang werden existierende Arbeiten und Projekte genannt, die für die praktische Durchführung genutzt werden können.

Bereitstellung von Diensten

Abhängig vom Grad der Spezialisierung wird ein anzubietender Dienst vom Entwickler selbst betrieben oder aber an einen Diensteanbieter weitergegeben. Letzterer hat in der Regel die Voraussetzungen für einen permanenten, zuverlässigen und ökonomisch sinnvollen Betrieb in einem Rechenzentrum. Von beiden Seiten her können zusätzlich zur Implementierung Metainformationen wie die Dienstbeschreibung, Angabe von NFE, Vertragsvorlagen und Tarifinformationen hinzugefügt werden.

Die Bereitstellung kann durch einen dienstspezifischen Vertrag mit SLA oder durch eine Generalvereinbarung abgesichert werden. In jedem Fall wird der Dienst auf einem oder mehreren Servern in einem Repository vorgehalten oder in einer Laufzeitumgebung installiert. Verlaufen die Installation und abschließende Überprüfungen erfolgreich, so wird der Dienst in einem Verzeichnis (Dienstregistry) eingetragen. Von diesem Moment an kann er von einem potenziellen Benutzer gefunden werden.

Nutzung von Diensten

Der Prozess der Nutzung beginnt mit der Suche nach einem Dienstangebot in einer bestimmten Domäne. Neben dem Primärkriterium der Funktionalität können dabei je nach Unterstützung des Anbieters qualitative Anforderungen und Kontextbedingungen, aber auch zu erreichende Ziele semantisch definiert werden.

Ist ein passender Dienst gefunden worden, so kann dieser entweder sofort oder erst nach einer Konfiguration und vertraglichen Bindung genutzt werden.

Der Vorgang der Vertragsbindung beinhaltet im Regelfall die Auswahl der zu garantierenden Eigenschaften durch den Nutzer, die Erstellung eines Vertragsangebotes aus Vorlagen und die Entscheidung über eine Annahme des Angebotes durch den Betreiber. Stellvertretend für den Betreiber können dabei Software-Agenten zum Einsatz kommen.

Wurde ein Vertrag abgeschlossen, so hat der Benutzer von da an die Möglichkeit, den Dienst zu den abgeschlossenen vertraglichen und tariflichen Bedingungen zu benutzen. Kommt es zu einer tatsächlichen oder drohenden Verletzung der Bedingungen seitens des Betreibers, so können autonome Anpassungen der Dienste, Dienstumgebungen oder vertraglicher Bestimmungen im Rahmen vorher definierter Toleranzbereiche vorgenommen werden.

2.2.3 Systemseitige und nutzerseitige Rückmeldungen

Adaptive und selbstadaptive Systeme benötigen eine ständige Rückkopplung von beobachtetem Verhalten zu gewünschtem Verhalten. In Dienstplattformen sind zwei Rückkopplungskanäle von besonderer Bedeutung: Eine objektive Beurteilung der Dienstgüte durch Monitoring, und eine subjektive Beurteilung durch Benutzerbewertungen.

2.3 Klassifikation und Struktur von NFE

Jeder Dienst besitzt funktionale Eigenschaften, die angeben, was ein Dienst leistet. Darüber hinaus besitzt jeder Dienst jedoch auch nichtfunktionale Eigenschaften (NFE), die angeben, wie ein Dienst seine Leistung erbringt. Für funktionale Eigenschaften gibt es, wie im vorangegangenen Abschnitt aufgeführt, vielfältige syntaktische und semantische Beschreibungsmöglichkeiten, welche teilweise bereits standardisiert sind. Die Spezifikation der NFE wird hingegen zumeist informell vorgenommen. Es fehlen neben einer standardisierten Ausdrucksmöglichkeit insbesondere auch gemeinsame Modelle, die einen Vergleich ermöglichen. Die vorliegende Arbeit benötigt allerdings ein NFE-Modell für eine feingranulare Angabe von Metaqualitätsmetriken. Ausgehend von einer Analyse existierender üblicher NFE-Angaben wird daher ein erweiterbares Modell der Innenstruktur von NFE abgeleitet.

2.3.1 Grundlagen der NFE-Modellierung

Die Modellierung von NFE lässt sich auf zwei zusammenhängende Teilmodelle aufteilen. Ein makroskopisches Modell stellt die Klassifikation der NFE dar, welche Informationen über die Domänen der Eigenschaften enthält. Eine derartige Klassifikation teilt die NFE in QoS-Eigenschaften, Metadaten zur Herkunft eines Dienstes, Preisangaben und Kontextinformationen ein. Auch Abhängigkeiten zwischen NFE lassen sich auf der Makroebene modellieren. Die Klassifikation lässt sich baumstrukturiert als Hierarchie oder graphstrukturiert als Ontologie repräsentieren. Ein zweites, mikroskopisches Modell betrachtet den generischen inneren Aufbau der NFE. Es enthält Informationen über die Expressivität der Eigenschaftsangaben, die Möglichkeit zu deren Annotation, Wertbeschränkungen, Prioritäten und Messungen.

2.3.2 Existierende Ansätze

Die makroskopische Sicht wird in dieser Arbeit nicht vertiefend betrachtet. Klassifikationen von NFE sind in ausreichender Zahl in der Literatur zu finden [10, 8, 75]. Eine dominierende Klassifikation ist nicht erkennbar und wird demzufolge auch nicht vorausgesetzt.

Hingegen ist eine präzise und ausdrucksstarke Notation der einzelnen NFE von besonderer Bedeutung. Als besondere Anforderung wird die Annotationsfähigkeit von NFE aufgestellt. Dieser Aspekt ist in der Literatur bisher nur in Ansätzen betrachtet worden und muss für diese Arbeit ausgebaut werden.

Die meisten Ansätze behandeln NFE als *atomare* Eigenschaften. Sie können je nach Ausdrucksstärke *simpel* oder *komplex* sein, von einfachen Schlüssel-Wert-Paaren mit ganzzahligen Werten bis hin zu kontextabhängigen, berechnungsintensiven Wertzuweisungen. Sie sind jedoch selten als *komposite* Eigenschaften repräsentiert. Analog zur zunehmenden externalisierten Strukturierung von Diensten durch Prozesssprachen mit deklarativen Anteilen anstelle von reinen imperativen und objektorientierten Programmiersprachen ist auch die Betrachtung der Innenstruktur von NFE sinnvoll. Ein paar ausgewählte Ansätze zur Spezifikation von NFE werden daraufhin untersucht.

Die Web Service Modelling Language (WSML) ermöglicht die Angabe von NFE mit optionalen Annotationen [22]. Diese nachträgliche Ergänzung der Sprache ist ein generischer Mechanismus ohne Vorgabe bestimmter Annotationen. WSML selbst beinhaltet dabei keine Möglichkeit der Angabe komplexer Werte oder sonstiger Strukturen in NFE. Eine Formalisierung der NFE-Spezifikation als WSML-Erweiterung wird durch das Policy Centered Metamodel (PCM) [77] vorgeschlagen. Dabei werden die NFE-Angaben mit Hilfe von Ontologien klassifiziert sowie Wertebereiche und abhängig von NFE berechenbare aggregierte NFE eingeführt.

NFE nach dem Modell von O’Sullivan [75] sind teilweise sehr umfangreich strukturiert. Im Gegensatz zu WSML sind auch temporale Aspekte ausdrückbar. NFE in den Domänen Preis, Bezahlmodalitäten, Rechte und Pflichten eines Dienstanwenders sowie Sicherheit sind mit ausdrucksstarken Operatoren, Wertebereichen und Einheiten versehen. Allerdings ist das Modell stark domänenabhängig, es stellt keine generische Struktur für NFE zur Verfügung. Dabei wäre eine Trennung zwischen NFE-Aufbau und NFE-Domäne sinnvoll.

Der Aufbau von NFE im WSQoSX-Framework [8] erfolgt als Quadrupel bestehend aus Name, Einheit, Operator für gerichtete Werte sowie einem Aggregationsparameter. Im Fall der Nutzung der NFE in SLAs wird eine Untermenge von WSLA genutzt. Nichtfunktionale Anforderungen können für die Aushandlung der Verträge numerisch und als Wahrheitswerte angegeben werden. Weiterführende strukturelle Merkmale von NFE, insbesondere die Annotierbarkeit sowie verstärkte Expressivität, sind in WSQoSX nicht vorgesehen.

Die Contract Quality Modelling Language (CQML bzw. CQML+) definiert ebenfalls den Aufbau von NFE, allerdings auf Komponentenebene, so dass dienstspezifische Anwendungen wie SLA-Vorlagen nicht untersucht worden sind [131]. In CQML+ sind sowohl numerisch getypte Wertangaben möglich als auch Mengen und Enumerationen. Einheiten sind als informelle Kennzeichnungen möglich. Einschränkungen etwa in Verträgen lassen sich durch komplexe gerichtete Grenzwerte angeben, wobei die Grenzen hart oder weich sein können. Komplexe Wertverteilungen lassen sich über statistische Methoden und Abhängigkeiten

von NFE untereinander ausdrücken. Eine Annotierbarkeit von NFE ist hingegen nicht explizit vorgesehen.

Die Object Management Group (OMG) betreibt Standardisierungsbestrebungen im SOA-Umfeld. Die derzeitigen Arbeiten zur standardisierten Angabe von NFE sehen eine hohe Qualität und Ausdruckskraft vor, enthalten aber keine expliziten Angaben zu Annotationen oder sonstigen strukturellen Merkmalen [73].

Die Integration der NFE-Verarbeitung in einen Dienstlebenszyklus ist ein wenig untersuchtes Gebiet. Wie bei allen Metadaten ist auch die manuelle Spezifikation von NFE kostenintensiv und wird daher oftmals nicht als Teil des Entwicklungsprozesses von Diensten gesehen. Automatische und semiautomatische Verfahren existieren, sind aber durch einen Bruch zwischen experimentellen Retrieval-Techniken und Anwendungsunterstützung sehr eingeschränkt nutzbar [47, 6]. Sowohl generische NFE wie Dublin-Core-Metadaten als auch überwachbare dienstspezifische Angaben benötigen somit eine bessere Integration in den Lebenszyklus und damit verbunden in Entwicklungsumgebungen und Dienstplattformen. Anforderungen an NFE speziell für den Dienstauswahlprozess als einen Schritt des Lebenszyklus sind bereits untersucht worden [127]. Sie beinhalten ein leistungsfähiges Modell für nichtfunktionale Eigenschaften sowie eine flexible Auswertung der Eigenschaften zur Laufzeit unter Berücksichtigung der Domänen der Eigenschaften. In der Untersuchung sind jedoch nur zwei Ansätze zur Evaluierung der Eigenschaften zur Laufzeit gefunden worden, von denen keine Annotationen auf NFE unterstützt. In diesen Ansätzen können Aspekte wie die Herkunft oder Gültigkeit von NFE somit nicht ausgedrückt werden.

2.3.3 Ableitung struktureller Merkmale

Basierend auf der Analyse der NFE-Ansätze lassen sich gemeinsame strukturelle Merkmale erkennen, die als Anforderungen für ein in dieser Arbeit vorausgesetztes NFE-Modell in Frage kommen. Dies betrifft insbesondere die Notwendigkeit der Bildung eines annotierbaren, ausdrucksstarken Modells sowie die Integration in den Lebenszyklus.

NFE sind demnach durch einen eindeutigen Schlüssel identifiziert, der je nach Klassifikationsschema ein Name oder ein ontologisches Konzept sein kann. Sie haben dazu einen Wert unterschiedlicher Komplexität und Expressivität. Zusätzlich zu dieser Schlüssel-Wert-Relation kann es eine Anzahl von Annotationen geben, die weitere Aussagen über die NFE zulassen. Beispiele dafür sind:

1. zur Laufzeit generierte Daten wie dynamisch gemessene Werte und Qualitätsprüfzahlen
2. von Benutzern gesammelte Rückmeldungen wie Bewertungen und Reputationseigenschaften
3. kontextabhängige Anbietereinschränkungen zur Gestaltung differenzierter Qualitätsprofile
4. Zugriffsrechte zur Einschränkung der Sichtbarkeit von Eigenschaften

Weitere Annotationen sind auf der Metamodell- bzw. Schemaebene möglich, etwa die Dokumentation, Herkunftsangabe und Standardisierungsinformationen

zu jeder Eigenschaft. Ein abgeleitetes Modell verbindet mehrere Ausprägungen dieser Eigenschaften anhand einer Klassifikation miteinander.

Die Abbildung 2.7 fasst grafisch die Strukturen zusammen, die den meisten Arbeiten zu NFE gemeinsam zugrundeliegen. Erkennbar ist der hohe Wiederverwendungsgrad struktureller Merkmale der NFE-Angaben in Dienstbeschreibungen. Neben reinen Ergänzungen ist allerdings auch eine Reinterpretation eines Merkmals, der Wertangabe, notwendig. Diese ist stark von der Ausdrucksstärke und der Einbindung in den Lebenszyklus abhängig.

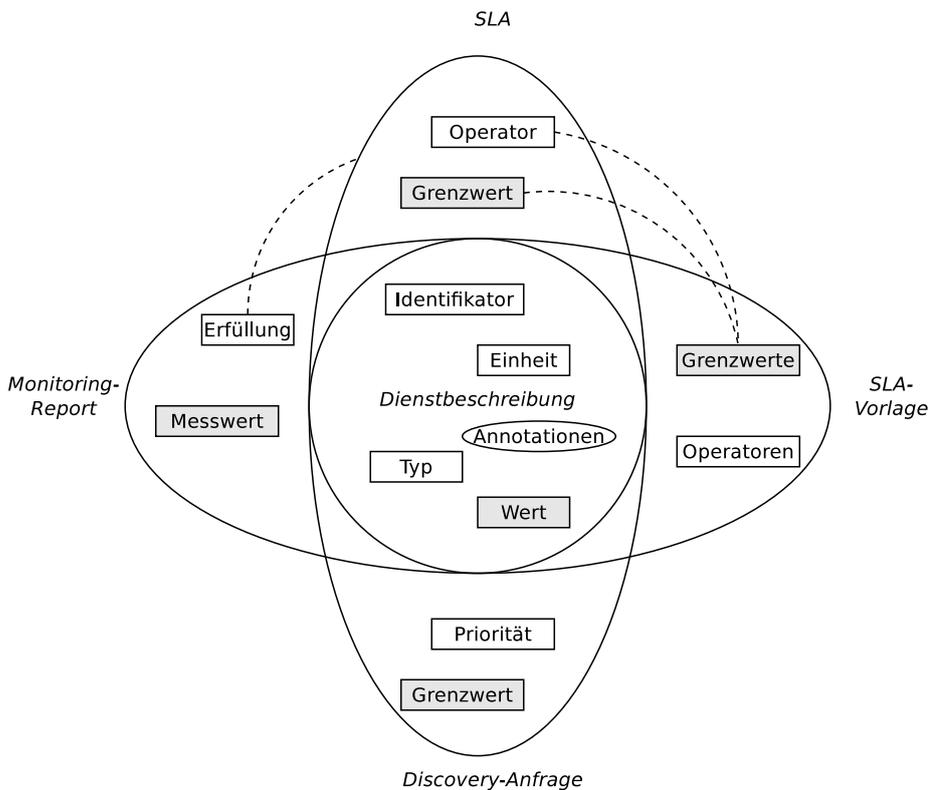


Abbildung 2.7: Modelleigenschaften: Annotierbare interne Strukturen je nach Auftreten der NFE

2.3.4 Zusammenfassung NFE-Strukturen

Bisherige Arbeiten zu NFE gehen meist auf die Nutzung und stellenweise auf die Klassifikation der Eigenschaften ein, jedoch nur in seltenen Fällen auf den strukturellen Aufbau. Die Bildung eines annotierbaren NFE-Modells ermöglicht demgegenüber eine feingranulare Annotation jeder Eigenschaft beispielsweise mit Qualitätsangaben.

2.4 Dienst-, Datenqualität und Metaqualität

Die Berücksichtigung qualitativer Aspekte einer Dienstauführung ist unter dem Term Dienstgüte oder *Quality of Service* (QoS) bekannt. Die Aspekte werden zunehmend direkt in deklarativen Dienstbeschreibungen und semiformalen elektronischen Vertragsdokumenten als Annotationen im Sinne nichtfunktionaler Eigenschaften festgehalten. Eine umfassende Betrachtung des Themas Dienstqualität muss somit auch die Qualität der Beschreibungen berücksichtigen, wodurch zusätzlich zur QoS auch die Datenqualität (DQ) oder Quality of Data (QoD) in den Fokus rückt. Qualitätsmodelle und -maße existieren in großer Zahl je nach Anwendungsdomäne. Man unterscheidet beispielsweise nach Qualität von Software, Diensten, Daten, Prozessen und Erzeugnissen (*Quality of Process/Product*, beide QoP). Im Sinne einer Ende-zu-Ende-Dienstleistungskette ist zusätzlich die Erlebnisqualität (*Quality of Experience*, QoE bzw. QoX) relevant. Die folgenden Abschnitte führen in die Grundlagen zu Qualitätsbetrachtungen ein und führen aktuelle, für die Themenstellung relevante Ansätze und Umsetzungen auf.

2.4.1 Dienstqualität

Der direkte Zusammenhang der Güte einer erbrachten Dienstleistung mit der Bereitschaft zur Gegenleistung hat im Dienstleistungssektor den Begriff Dienstgüte oder QoS hervorgebracht. Vertragliche oder informelle Vereinbarungen darüber werden als *Service Level Agreement* (SLA) bezeichnet. Überlegungen zu einer formal definierbaren und überwachbar einhaltbaren Dienstgüte haben ihren Ursprung in der Telekommunikationsindustrie [70]. Über die Konvergenz der Netze hat sich das Thema dann zunehmend auf moderne paketorientierte und mobile Netzwerkarchitekturen ausgebreitet [25, 64]. Über Betriebssysteme und komponentenbasierte Middlewaresysteme ist die unbedingte Einhaltung der garantierten Dienstgüte auch zu einem Softwareproblem geworden [35, 69].

Je nach Systemebene und den damit zusammenhängenden Ressourcen kommen so unterschiedliche Verfahren zur vorausschauenden und reaktiven Wahrung der Dienstgüte zum Einsatz, die zumeist auf einem Modell konkurrierender Dienstzugriffe basieren, teilweise aber auch unerwünschte Fehlerquellen wie Hardwareausfälle berücksichtigen.

Netzwerk Reservierung und Priorisierung von Bandbreite über Verfahren wie DiffServ oder IntServ sowie Queuing-Disziplinen und dynamische Beschränkung der Bandbreite (Throttling).

Betriebssystem Reservierung von Ressourcen wie CPU, Arbeitsspeicher und nichtflüchtigem Speicher, Echtzeitscheduling, Isolation von Prozessen und Virtualisierung.

Dienstcontainer Priorisierung der Dienstauführung anhand von Konfigurationseinträgen und Verträgen, dynamische Auswahl der Inhaltsrepräsentation (Content Negotiation).

Dienst Spezifische Adaptionsmechanismen wie etwa gezieltes Auslassen von Elementen in einem Datenstrom.

Die Einhaltung der Dienstgüte kann in allen Domänen durch Dienstnutzungsvereinbarungen vertraglich garantiert und durch adaptive Systembestandteile begünstigt werden, wenn die vertraglich festgelegten Grenzwerte und Kosten einer Nichteinhaltung bekannt sind. Erkennbar ist ebenfalls eine domänenübergreifende Kombination aus pessimistischen Verfahren mit vorausplanender Reservierung von Ressourcen mit optimistischen Verfahren, die sich ausschließlich auf eine Adaption im Fehlerfall stützen. Es kommen dabei häufig Verfahren zum Einsatz, die die Dienstauführung bezüglich der zugesicherten Eigenschaften automatisch durch von den Garantietermen abgeleitete Messvorgänge überwachen [35].

2.4.2 Erlebnisqualität

Die Erlebnisqualität gibt an, inwieweit die Qualität eines Dienstes über sein Transportmedium und seine Interaktionsschnittstellen bei dessen Benutzern ankommt oder inwiefern Darstellungs-, Daten- oder Transportmedienmängel eine beeinträchtigende Wirkung entfalten können [118].

Der Begriff Wahrnehmungs- oder Erlebnisqualität bezieht sich somit auf eine unterschiedliche subjektive Wahrnehmung von Qualität durch Menschen im Gegensatz zu objektiven, messbaren Dienstqualitätsparametern. Dieses Prinzip wird bereits in vielen Bereichen der Informatik erkannt. So ist zwar in der Kryptographie die Kollisionswahrscheinlichkeit eines Hash-Algorithmus berechenbar, die besten Algorithmen sind jedoch nicht gleichzeitig die intuitiv nachvollziehbarsten. Insbesondere audiovisuelle Inhalte können vom Betrachter trotz informationstheoretisch hoher Unterschiede als identisch klassifiziert werden, was als Grundprinzip für Wahrnehmungshashfunktionen [24] genutzt wird. Nach ähnlichem Muster ist die Wahrnehmung einer Dienstauführung subjektiv zwar zu großen Teilen von den qualitativen Eigenschaften beeinflusst, es spielen aber noch weitere Faktoren eine Rolle, insbesondere die Benutzbarkeit sowie domänenspezifische Größen [116]. Die Erlebnisqualität als Ausdruck einer subjektiv wahrgenommenen Ende-zu-Ende-Dienstgüte kann über objektive Monitoringdaten hinaus als subjektive Benutzerrückmeldung registriert werden.

2.4.3 Datenqualität

Die Qualität von Daten (DQ), und somit auch von dienstbezogenen Dokumenten, ist eine mehrdimensionale Größe. Für viele Probleme in der Praxis reichen die beiden DQ-Dimensionen Vollständigkeit und Genauigkeit aus. Eine weit verbreitete feinere Einteilung der Datenqualität sieht vier Qualitätsarten vor: intrinsische, kontextuelle, repräsentative und Zugriffsqualität [113]. In die ersten beiden Kategorien können objektive DQ-Metriken oder DQ-Dimensionen wie Genauigkeit, Konfidenz, Objektivität, Reputation, Vollständigkeit, Datenmenge und Aktualität eingeordnet werden. Daraus lassen sich subjektive DQ-Metriken wie Relevanz oder Mehrwehrt für eine bestimmte Anwendung in einem bestimmten Kontext ableiten. Die repräsentative DQ-Kategorie umfasst Metriken wie die Interpretierbarkeit, Verständlichkeit, Konsistenz und Prägnanz von Daten. Schließlich drücken die Zugänglichkeit und die Sicherheit von Daten ihre Zugriffsqualität aus. Diese feingliedrige Unterteilung der Datenqualität ist in vielen Anwendungsbereichen wie der Datenstromverarbeitung nützlich und dient unter anderem als Grundlage für das Datenqualitätsmodell DQMx [54]. Darüber

hinaus sind weitere DQ-Metriken wie Aktualisierungsfrequenz und Granularität bekannt [67], so dass man von einer stets unvollständigen und anwendungsabhängigen Klassifikation ausgehen muss.

2.4.4 Metaqualität

Der Begriff Metaqualität taucht bereits vereinzelt in verschiedenen Wissenschaften auf, ohne dass es jedoch eine genaue oder gar allgemeingültige Definition dafür gibt. Erkennbar ist ein Bezug zu ausgewiesenen Qualitätsmerkmalen von Daten und Prozessen als Anwendung von DQ-Metriken auf qualitätsbezogene Beschreibungen. Für einen systematischen Vergleich von Ansätzen zur Bestimmung und Verwendung von Metaqualitäts-Metriken sind vor allem die unterstützten DQ-Dimensionen sowie die Rückführbarkeit als Annotation in die qualitätsbezogenen Beschreibungen von Interesse.

Begriffsklärung

In [122] bezeichnet Metaqualität bezogen auf Abläufe in Bibliotheken die *Qualität von Qualitätsprozessen in einer Organisation*. Im Bereich der Informationssysteme für geografische Anwendungen (GIS) wird Metaqualität nach [109] bezogen auf Qualitätsdaten bezeichnet als *Qualität der Qualitätsinformation zur Charakterisierung der Qualität der Qualitätsbeschreibung*. Dabei werden die Konfidenz als notwendiges, hingegen die Zuverlässigkeit, die Methodologie und das Abstraktionsniveau als hilfreiches Kriterium im Sinne einer DQ-Dimension angegeben. Zur Angabe von Metaqualitätsinformationen werden zunehmend Ontologien eingesetzt. Die Trennung der Ontologien von den eigentlichen Daten sowie die Angabe einer für Produzenten und Konsumenten gleichermaßen objektiv brauchbaren Tauglichkeitsmetrik ist dabei Gegenstand aktueller Forschung [36].

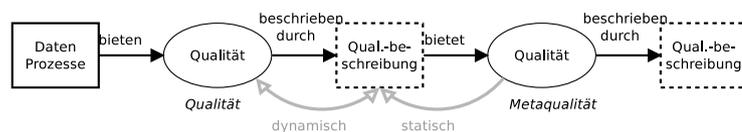


Abbildung 2.8: Allgemeine Definition der Metaqualität

Eine allgemeine Definition des Begriffs Metaqualität lässt sich somit wie in Abbildung 2.8 dargestellt als Qualitätsmaß für Qualitätsangaben ableiten. In der Informatik, speziell im Bereich der Entwicklung und des Einsatzes von Komponenten und Diensten, ist der Begriff Metaqualität bisher nicht geläufig. Dennoch gibt es verschiedene Ansätze zur Beurteilung der Qualität von Dienstbeschreibungen, wenn auch nicht speziell der darin enthaltenen Qualitätsangaben und weiteren NFE. Diese Ansätze und ihre statistischen Maße sollen in der Arbeit betrachtet werden, um eine spezialisierte und wohldefinierte Metaqualitätsdefinition für Dienste zu bilden.

Ansätze zur Qualitätsbewertung

Ein Problem nahezu aller QoS-Spezifikationen ist ihre Qualität bezogen auf die Präsenz, Aktualität und Präzision der Angaben. Sind QoS-Angaben nicht

präsent, können sie bedingt über Lernverfahren in Bayesschen Netzen anhand von vertrauenswürdigen subjektiven Rückmeldungen probabilistisch ermittelt werden [120]. Der Erfolg solcher Verfahren wird durch Mehrdimensionalität, Kontextabhängigkeit und Nichtdeterminismus der oftmals domänenspezifischen Eigenschaften eingeschränkt.

Ein Ansatz unter Nutzung des *Web of Trust* beschäftigt sich konkret mit den Qualitätsmaßen **Vertrauenswürdigkeit** und **Zuverlässigkeit** der Dienstbeschreibungsmetrik Bewertung [16]. Über einen Kalman-Filter wird der Einfluss der Bewertungen verteilenden Personen bestimmt. Eine Generalisierung des Ansatzes ist von den Autoren nicht vorgesehen.

Eine weitere Arbeit betrachtet die Qualität von Ontologien [42] im biomedizinischen Bereich. Da Ontologien zur Beschreibung funktionaler und nichtfunktionaler Diensteseigenschaften eingesetzt werden können soll dieser Ansatz mit betrachtet werden. In ihm werden die drei Qualitätsdimensionen Vollständigkeit, Korrektheit und Effektivität klassifiziert. Die Vollständigkeit als intrinsisches Maß wird in einem offenen System nie zu 100% erreicht, da das vorhandene Wissen stets wachsen kann. Die Abdeckung dieser Dimension ist jedoch zumindest bestimmbar. Die Korrektheit als weiteres intrinsisches Maß definiert die Wohldefiniertheit und Wohlstrukturiertheit sowie die Resistenz gegenüber evolutionären Veränderungen einer Ontologie. Als extrinsisches Maß ist die Effektivität von einem Anwendungsfall abhängig und definiert dafür das Verhältnis von Aufwand und Nutzen einer Ontologieerstellung, beispielsweise unter Einbeziehung automatischer Inferenzmechanismen oder manueller Annotationen. QoS-bezogene Prüfungen wie automatisiertes Monitoring werden nicht behandelt.

Im experimentellen SOA-Umfeld existiert mit dem Projekt QWS [3] ein aufbereiteter Datensatz zu dynamischen QoS-Werten und statischen Metriken wie der Vollständigkeit der Dokumentation einer WSDL-Dienstbeschreibung und der Konformität der Beschreibung mit der WSDL-Spezifikation und den WS-Interoperabilitätsrichtlinien. Diese Werte wurden unter Verwendung eines eigenen Crawlers namens WSCE für Dienstbeschreibungen anhand von UDDI-Einträgen generiert. Als aggregierte Qualitätsmetriken werden eine Relevanzfunktion WsRF sowie eine vierteilige Klassifikation der Dienstgüte von Bronze bis Platin generiert.

Eine ähnliche Analyse wurde im Rahmen des Forschungsprojekts ServiceFinder durchgeführt [111]. Hierbei stand allerdings die Problematik der Duplikaterkennung und automatischen funktionalen Klassifizierung von Diensten anhand ihrer Beschreibungen im Zentrum der Arbeiten. Erkennbar ist aus den Ergebnissen eine deutliche Dominanz von WSDL 1.1 als Dienstbeschreibungssprache. Damit ist es bezüglich internetweit verfügbarer Dienstbeschreibungen nicht möglich, Aussagen zur statischen Qualität nichtfunktionaler Dienstangaben zu treffen oder gar deren Übereinstimmung mit tatsächlich erzielbaren Werten festzustellen.

Eine statische Analyse der Qualitätsmetrik **Kopplungsgrad** von BPEL-Prozessen ist ebenfalls bekannt [125]. Diese als *Average Service Invocation Coupling* (ASIC) bezeichnete Metrik gibt an, wie stark ein Dienst von einem anderen genutzt wird. Die Metrik unterteilt sich in eine Zustandsabhängigkeitsgrad (DSD), einen Persistenzabhängigkeitsgrad (DPD) und eine durchschnittliche Dienstabhängigkeit (ARSD). Eine quantitative Angabe der Übereinstimmung von Leistungsdaten der analysierten Dienste mit ihren Dienstbeschreibungen

ist in dem Ansatz nicht vorgesehen.

Die dynamische Analyse der Unterschiede zwischen beworbener und tatsächlich erzielter Dienstgüte wurde sowohl systemseitig [76] als auch nutzerseitig [78] bereits vorgeschlagen. Besonders der zweite Ansatz beinhaltet die subjektive Qualitätseinschätzung im Sinne einer QoE. Die Metrik der Abweichung wird als *user rating* bezeichnet. Es existiert allerdings bisher noch keine Umsetzung, da die Untersuchungen rein simulativ durchgeführt worden sind. Im alternativen Ansatz mit dem Namen *Quality of Web Services* (QoWS) werden zum Selektionszeitpunkt Benutzerbewertungen mit Messungen gewichtet kombiniert, um die Übereinstimmung der angeforderten mit den geleisteten Qualitätsdaten zu bewerten.

Ein Ansatz für die kontextabhängige Dienstausswahl berücksichtigt bereits explizit die Qualität der Kontextangaben in der Dienstbeschreibung [53]. Dabei wird die Sammlung der Kontextinformationen durch spezielle Geräte und der Einfluss inexakter Übereinstimmungen durch variierende Kontextqualität untersucht. Der Ansatz nimmt allerdings die a-priori-Spezifikation des Grades der *Unsicherheit* durch die Hersteller der Kontextsensoren an. Die Theorie hinter Kontextunsicherheit ist dabei bereits in [20] erarbeitet worden. Der Ansatz liefert einen Beitrag zur Entwicklung eines Frameworks für die Metaqualität von Diensten. Er ist allerdings weder vollständig, da er sich auf Kontext-NFE beschränkt, noch ausreichend, da er eine bereits bekannte und ohne weitere Verifikation als gültig angenommene Spezifikation der Kontexteigenschaften annimmt.

Die Schlussfolgerung aus den bisherigen Ansätzen zur Bestimmung von Qualitätsmetriken in Diensten ist ein deutliches Defizit an uniform spezifizierten und für die Dienstausswahl auf unterschiedlichen Granularitätsebenen nutzbaren Metaqualitätsinformationen. Die Tabelle 2.3 fasst die vorhandenen Ansätze zur Qualitätsbewertung von Dienstbeschreibungen noch einmal hinsichtlich zweier DQ-Dimensionen und zweier die Informationen nutzender Verfahren zusammen. Während die Bestimmung einer der Metaqualität vergleichbaren Größe zumeist nur auf eine DQ-Dimension bezogen erfolgt und noch seltener ein Abruf der Metriken als Beschreibungsdokument konzeptionell vorgesehen ist, sind Behebungen erkannter Qualitätsdefizite in keinem Ansatz enthalten.

2.4.5 Zusammenfassung dienstbezogener Qualitätsaspekte

Von den untersuchten qualitätsbezogenen Themenfeldern ist bezogen auf Dienstplattformen bislang ausschließlich die eigentliche Dienstqualität (QoS) hinreichend untersucht worden. Ansätze zur dynamischen Auswahl oder Ersetzung von Diensten verlassen sich auf vorhandene Qualitätsangaben und stellen die Qualität der Angaben selten in Frage. Eine solche Herangehensweise ist den bislang zumeist in geschlossenen Systemen betriebenen Diensten geschuldet, wobei die Beschränkung auf solche und die Notwendigkeit für erweiterte Konzepte für einen Betrieb in offenen, heterogenen Systemen bereits bekannt und akzeptiert ist. Diese Arbeit sollte demnach einen Beitrag leisten, die Qualitätsangaben zu überprüfen und anhand von deren Metaqualität den Prozess der Auswahl an hochqualitativen Diensten mit belastbaren Eigenschaftsangaben zu verbessern.

Tabelle 2.3: Ansätze zur Qualitätsbewertung von Dienstbeschreibungen

Name	Vollständigkeit	Übereinstimmung	Annotation der Metaqualität	Korrektur der NFE
Bayessche Netze	nein	subjektiv durch Benutzer	nein	nein
Web of Trust	nein	teilweise: Zuverlässigkeit	nein	nein
Qualität von Ontologien	ja	teilweise: nicht QoS-bezogen	nein	nein
QWS	teilweise: WSDL-Metriken	nein	ja: vierstufig	nein
Service-Finder	nein	nein: nur Monitoring	nein	nein
ASIC	teilweise: BPEL-Metriken	nein	nein	nein
User Ratings	nein	subjektiv durch Benutzer	ja (abstrakt)	nein
QoWS	nein	ja	nein	nein

2.5 Dienstumgebungen

Die folgenden Abschnitte geben einen Querschnittsüberblick über existierende Ansätze zur Verwaltung und zum Betrieb von Diensten. Dabei werden etablierte Dienstplattformen auf System- und Middlewaredbene den neueren Ansätzen in serviceorientierten Architekturen gegenübergestellt. Die wichtigsten Vertreter jeder Gruppe werden vorgestellt und auf ihre Brauchbarkeit bezüglich des Vorhabens der Dissertation untersucht.

2.5.1 Systemnahe Dienstplattformen

Der Entwurf und die Umsetzung von Softwarearchitekturen für die Bereitstellung von digitalen Diensten ist bereits mit dem ersten Aufkommen von entfernt aufrufbarer Funktionalität als eigenständiges Thema identifiziert worden. Diese Entwicklung vollzog sich beginnend mit einer Separierung der Adressierung von Diensten auf Betriebssystemebene durch Einführung von logischen Netzwerkanschlüssen, den heutigen TCP-Ports [27], hin zu dienstunterstützender Middleware wie dem Distributed Computing Environment [55] und den damit zusammenhängenden Forschungsfragen wie Dienstregistrierung und zentraler Dienstverwaltung in dezentralen Systemen. Bereits DCE kannte das Konzept von Plattformdiensten *Core Services* als plattformintegrale modulare Bestandteile mit standardisierter Funktionalität und einheitlich auf RPC aufsetzenden Kommunikationsschnittstellen. Es sind jedoch nur wenige, zumeist systemnahe, Core Services definiert und umgesetzt worden. Da diese nicht im Sinne heutiger

Dienste mit einer einheitlichen auffindbaren Beschreibung und einer Lebenszyklusverwaltung versehen sind, ist DCE kein geeigneter Kandidat für qualitätssichernde Verfahren.

2.5.2 Komponentenplattformen

Als Vertreter der Komponentenplattformen stellt die vormalige *Open Services Gateway Initiative* ein Java-basiertes Modulsystem zur Verfügung. Alle Module, auch als *Bundles* bezeichnet, sind dynamisch installierbar und laufen innerhalb eines Adressraums. Die Kommunikation erfolgt über lokale Dienstschnittstellen. Sämtliche Dienste sind zur Laufzeit abfragbar. Zusätzliche Bundles können von einem globalen Archiv wie OBR (*OSGi Bundle Repository*) bezogen werden. Über die Erweiterung R-OSGi kann die Plattform auch im verteilten Betrieb arbeiten [84].

Im Gegensatz zu DCE wird eine wesentlich höhere Dynamik und Modularität erzielt. Gleichzeitig fehlt auch bei OSGi eine benutzerzentrische Funktionalität zum Anbieten, Suchen und Nutzen von Diensten. Eine vertragliche Zusicherung von Diensteseigenschaften wird nicht unterstützt. Zudem sind alle Dienste an eine Java-Implementierung gebunden, was gegen das Prinzip der Implementierungstransparenz verstößt.

2.5.3 Dienstorientierte Architekturen und SOA-Plattformen

Die steigende Popularität des Internet, speziell des World Wide Web, führte zu einer Zunahme an global verfügbaren Onlinediensten, deren Standardisierung maßgeblich durch Webstandards beeinflusst wurde und somit auf eine höhere Netzwerkschicht als bei den bis dahin gängigen Dienstprotokollen abgebildet wurde. Aus dieser Entwicklung sind die mittlerweile üblich gewordenen Web Services konzeptionell hervorgegangen. Der Einfluss des Web lässt sich insbesondere bei der Verwendung von deklarativen Auszeichnungssprachen wie XML oder JSON feststellen. Das XML-Format wird in Protokollen wie HTTP und teilweise darauf basierender Protokolle wie XML-RPC und SOAP eingesetzt.

Die Weiterentwicklung auf Protokollebene und die Globalisierung von Dienstangeboten bedingte eine parallel stattfindende Verbesserung von Dienstarchitekturen. Neben eher dezentral und dynamisch wachsenden ressourcenorientierten HTTP-Dienstangeboten, den RESTful Web Services, haben sich dedizierte Web-Service-Management-Plattformen herausgebildet, welche meist auf prozedurale oder dokumentenorientierte SOAP-Dienste beschränkt sind. Die Gründe für die unterschiedlichen Sichtweisen auf Webdienste liegen in den unterschiedlichen Herangehensweisen zur Dienstbeschreibung und zur Komposition auf Basis von Choreographie oder Orchestrierung sowie in der asymmetrischen industriellen Standardisierung und Werkzeugunterstützung [130].

Abgrenzend zu den bisher betrachteten eingeschränkten Komponenten- und Dienstumgebungen soll der Begriff Dienstplattform somit Verwaltungs- und Ausführungsfunktionen bezeichnen, die den Lebenszyklus eines Web Services von der Installation über die Nutzung bis hin zur langfristigen Verwaltung unterstützen.

Die momentan verfügbaren Dienstplattformen unterscheiden sich primär hinsichtlich der ausführbaren und verwaltbaren Diensttechnologien. Dies betrifft

die Paketierung der Dienste, deren Beschreibung sowie die zur Entwicklung eingesetzten Modelle und Frameworks. Sekundär sind Verwaltungsfunktionen wie Benutzer- und Sicherheitsverwaltung, QoS-Monitoring, adaptive Skalierbarkeit und Integration mit weiteren Plattformen von Bedeutung. Der weitaus größte Anteil wird durch Plattformen zur Ausführung von Diensten basierend auf Java-Technologien gestellt.

Das aus den Forschungsarbeiten zu semantischen Diensten unter Nutzung der WSMO-Methodologie hervorgegangene *Web Service Execution Environment* (WSMX) [41] stellt neben einigen Entwicklungswerkzeugen wie Transformationswerkzeugen und Editoren für die Abbildung funktionaler Konzepte vor allem eine Laufzeitplattform für die Dienstaufführung bereit. Zentrale Bestandteile sind Ontologie-Reasoner sowie Komponenten wie Dienstregistrierung und -suche, zielorientierte automatisierte Komposition, Mediation und Ausführung. Über das Dienstbeschreibungsformat WSML lassen sich nichtfunktionale Eigenschaften über ein dafür vorgesehenes Syntaxkonstrukt (nfp) direkt ausdrücken. Die weitere Verarbeitung der NFE ist in WSMO im Rahmen der Komposition und Mediation stark ausgebaut. Es sind jedoch keine dynamischen NFEs oder SLAs konzeptionell vorgesehen und sind demzufolge in WSMX auch nicht umgesetzt. WSMX wird kontinuierlich öffentlich entwickelt, eine laufende Instanz im Internet existiert nicht.

Einen ähnlichen Ansatz wie WSMX verfolgte die ASG-Plattform als Umsetzung eines *Adaptive Services Grid*. Es erweiterte die WSMO-Ansätze um Adaptivität speziell in kompositen Diensten durch Strategien wie Planung von Prozessen, Einfassung von Diensten in die Prozesse und dynamisches Neuplanen und Neubinden abhängig von durch Monitoring überwachten Qualitätskriterien [57]. Die Entwicklung beinhaltete schließlich jedoch keine Laufzeitplattform für die verwalteten Dienste. Der ASG-Prototyp ist nachnutzbar, eine permanente Instanz existiert hingegen nicht.

Die Plattform WSQoSX [8] stellt die qualitätsbewusste Dienstaufführung in den Mittelpunkt. Dazu gehört die Beschreibung von Qualitätseigenschaften, die heuristisch optimierte Suche nach passenden Diensten unter Berücksichtigung nichtfunktionaler Anforderungen, die Bildung von Verträgen und die Adaption von (komponierten) Diensten in Hinblick auf die Einhaltung der vertraglich garantierten Dienstgüte über ein Neuplanungsverfahren. Der entwickelte Prototyp ist nicht öffentlich verfügbar oder als Instanz im Einsatz.

In europäischen Ländern lässt sich eine verstärkte Förderung von gesamtgesellschaftlichen Dienstplattformen erkennen. Zwischen einer zugrundeliegenden Erweiterung von Netzwerk- und Kommunikationstechniken hin zu Netzwerken der nächsten Generation mit Ende-zu-Ende-Garantien und einer anwendungsseitigen Ausrichtung auf multimediale, mehrdimensionale und greifbare Inhalte werden unter dem Schlagwort Internet der Zukunft dazwischenliegende Plattformen als dienstorientierte Middlewaresysteme mit der Sammelbezeichnung Services and Software Architectures, Infrastructures and Engineering (SSAI) propagiert. Diese teilen sich wiederum auf in virtualisierte Basisinfrastrukturprojekte, Dienstplattformen und Benutzerschnittstellen für Dienste. Nebengeordnet zu dieser Laufzeitsicht werden neuartige Entwicklungsumgebungen für Dienste geschaffen.

Auf Architekturebene sind diese Plattformen entweder als monolithische Middlewaresysteme realisiert, deren Funktionen nur teilweise als Dienstschnittstelle zur Verfügung gestellt wird, oder als modulare dienstbasierte Middleware, deren

Kernbestandteile auf lose gekoppelten und austauschbaren Plattformdiensten beruht, analog zu den Core Services von DCE.

Neben den akademischen Ansätzen, von denen bisher keine permanente geschäftlich genutzte Instanz im Internet existiert, gibt es auch kommerzielle Anbieter mit teils ähnlicher Ausrichtung, die sich für einen Vergleich eignen.

Salesforce [33] betreibt mit `force.com` eine SaaS-Portal zur kostenpflichtigen Bereitstellung von Dienstleistungen über das Internet. Durch den Anbieter werden jedoch einige der Grundprinzipien von SOA nicht berücksichtigt. Insbesondere sind Dienste nicht technologieunabhängig, da zwangsweise die proprietäre Programmiersprache Apex eingesetzt werden muss. Da es keine weiteren Hostinganbieter für Programme in dieser Sprache gibt, ist zudem eine Anbieterabhängigkeit entgegen des Prinzips der losen Kopplung gegeben.

Die selektiv vorgestellten Vertreter von Dienstplattformen weisen bereits etliche Merkmale auf, die sie für einen Einsatz als Dienstmarktplätze qualifizieren würden. Es werden jedoch stets nur wenige der Anforderungen gleichzeitig erfüllt. Insbesondere existiert unter diesen existierenden Ansätzen keine Plattform, die eine echte Handelbarkeit von Diensten ermöglicht. Aus dem Grund werden die Betrachtungen nun auf zwar noch nicht im Internet nutzbare, aber zumindest in Entwicklung befindliche Prototypen und Forschungsansätze ausgeweitet.

2.5.4 Dienstplattformen für das Internet der Dienste

Für einen Betrieb im Internet der Dienste sind konventionelle SOA-Plattformen nur bedingt geeignet. Zum einen ist der Architekturstil der Plattformdienste nicht ausgereift genug, über den Austausch und die Bereitstellung von Plattformfunktionalität in komponierten Mehrwertdiensten komplexe handelbare Dienstleistungen technisch realisieren und anbieten zu können. Weiterhin sind die Plattformen zwar auf den Betrieb und die Verwaltung von Diensten ausgelegt, jedoch kaum auf eine Mediation zwischen Anbietern und Nutzern von Diensten. Die wenigen Ansätze zu einer nutzerzentrischen Mediation sind wiederum meist nur auf die Auswahl von Diensten, nicht jedoch auf die weitergehenden Aspekte wie langfristige vertragliche Bindung oder Ausführung ausgelegt. Als dritter Grund sei die unzureichende Einstellung auf das extrem heterogene Diensttechnologiespektrum genannt, welches sich bei einer Zunahme von Dienstanbietern eher noch verbreitern wird.

Aus diesem Grund entstehen zur Zeit eine Reihe von Infrastrukturansätzen, die die Forschung an inkrementellen Qualitätssteigerungen im Internet der Dienste überhaupt erst ermöglichen. Sie erweitern SOA-Plattformen um Aspekte des globalen Handels von Diensten auf speziellen Marktplätzen. Insbesondere wird der erprobte Betrieb unter realen Bedingungen im Internet als weiteres Kriterium für den Vergleich vorausgesetzt.

Durch die Abschaltung der UDDI-basierten *Univeral Business Registry* (UBR) ist ein neuerlicher Bedarf nach globalen Dienstverzeichnissen entstanden. Dienstverzeichnisse unterschiedlichen Umfangs füllen diese Lücke mittlerweile aus. Als Vertreter sind etwa Service-Finder unter `demo.service-finder.eu`, Service-Repository unter `service-repository.com` und Opossum unter `fusion.cs.uni-jena.de` zu nennen. Einige der Verzeichnisse enthalten bereits textuelle und grafische Metadaten zur Verfügbarkeit und zur Popularität der Dienste. Ein in die Dienstbeschreibung integriertes Format respektive dynamisch dadurch angeereicherte Beschreibungsdokumente sind jedoch derzeit nicht im Einsatz. Auch

beschränken sich die Metadaten auf das Monitoring entfernter Dienste, ein eigenes Hosting erfolgt nicht. Auch der Abschluss von Verträgen wird unabhängig davon in keinem der Ansätze unterstützt.

Die im Rahmen europäischer Förderprojekte getragene NESSI-Initiative für vernetzte Forschung an Software und Diensten führt einige der SSAI-Dienstplattformvorhaben in der Referenzarchitektur NEXOF-RA zusammen [23]. Diese noch in Entwicklung befindliche Architektur unterliegt einem für derartige Projektverhältnisse relativ offen zugänglichen Entwurfsverfahren. Die Architektur ist sehr umfangreich und ambitioniert ausgelegt, wie aus Abbildung 2.9 ersichtlich. Es existiert jedoch noch kein lauffähiger Prototyp. Im Fall einer generellen Verfügbarkeit dieser Entwicklungen ist für zukünftige Forschungsvorhaben zu Diensten und Dienstplattformen eine Konsolidierung auf eine Architektur denkbar. Die relevanten Architekturformen sollen am Beispiel eines SSAI-Projekts aufgezeigt werden.



Abbildung 2.9: Gesamtansatz von NEXOF-RA, vereinfachte Darstellung

Das Projekt SLA@SOI propagiert eine SLA-bezogene Dienstauführungs- und verwaltungsumgebung. Die Architektur reicht dabei von der Infrastrukturebene über die Dienste bis hin zu komplexen Geschäftsprozessen. Der Ansatz sieht zur Unterstützung der in den SLAs vereinbarten Qualitätskriterien einen Entwicklungsprozess vor, der die Modellierung von NFE zur Vorhersagbarkeit von SLA-Grenzwerten bereits beinhaltet. Durch Gewinnung von Monitoringinformationen zur Laufzeit können diese Informationen weiter ergänzt werden [30]. Die Ziele von SLA@SOI decken sich zu großen Teilen mit dem Thema dieser Arbeit. Eine Umsetzung der angestrebten Architektur des Projekts steht allerdings noch nicht zur Verfügung, wenn auch eine Veröffentlichung angestrebt wird.

2.5.5 Zusammenfassung Dienstplattformen

Zur besseren Übersichtlichkeit werden die existierenden SOA-Plattformen in der Tabelle 2.4 noch einmal bezüglich ihrer Funktionalität gegenübergestellt. Die ergänzende Tabelle 2.5 fasst zudem die nichtfunktionalen Aspekte zusammen.

Im Bereich der SOA-Umgebungen existieren nur wenige tatsächlich eingesetzte Systeme, die den Handel mit verteilbaren Diensten unterstützen. Auch die aktuellen Forschungsprojekte weisen kaum die notwendige holistische Herangehensweise auf, um einen hinreichend integrierten Dienstmarktplatz betreiben zu können. Zudem bietet kein Ansatz bisher eine durchgängige Überprüfung der

Tabelle 2.4: Funktionaler Vergleich existierender SOA-Plattformen

Ansatz	Registry für Dienstbeschreibungen	Discovery	Hosting von Diensten	Vertragsbindung	QoS-Garantien
DCE	RPC-Bindungen	RPC-Bindungen	RPC-Dienste	nein	nein
OSGi	Java-Objekte	Filterung nach Eigenschaften gemäß LDAP/RFC 1960	OSGi-Bundles	nein	nein
WSMX	WSML-Spezifikationen	funktionaler und nicht-funktionale Matchmarker	nein	nein	nein
ASG	WSML-Spezifikationen	funktionaler und nicht-funktionaler Matchmarker	BPEL-Kompositionen	teilweise, eigenes SLA-Modell Unified Monitoring Information	teilweise
WSQoSX	WSDL-Beschreibungen und SLA	Matchmaking während Dienstauf-ruf im Proxy	nein	Parameterdefinitions-datei	für bestimmte Metriken
SLA@SOI	nein	nein	nein	eigenes Modell auf Basis von WS-Agreement	Monitor-Plugins
NEXOF-RA	abstrakt	abstrakt	abstrakt	abstrakt	abstrakt
Salesforce	webbasiert	nein	Apex-Dienste	nicht-maschinenlesbar	für bestimmte Metriken

Tabelle 2.5: Nichtfunktionaler Vergleich existierender SOA-Plattformen

Ansatz	Technologie- neutralität	Anbieter- unabhängigkeit	Weiterentwicklung
DCE	+	+	-
OSGi	-	+	+
WSMX	+	+	+
ASG	+	+	-
WSQoSX	+	-	-
SLA@SOI	+	+	+
NEXOF- RA	+	+	+
Salesforce	-	-	+

Dienstgüte und davon abgeleitet eine programmatisch auswertbare Bereitstellung dynamisch angepasster Dienstbeschreibungsdokumente. Nur wenige Registrierdienste überwachen objektiv und durch Benutzerbewertungen subjektiv die Dienstauführung. Insbesondere ist in diesen Ansätzen kein Bezug auf individuell ausgehandelte Dienstgütevereinbarungen erkennbar.

Abbildung 2.10 zeigt abschließend, auf welchen Ebenen und auf welchem Abstraktionsniveau sich die in dieser Arbeit zu erstellenden konzeptionellen und architektonischen Erweiterungen von SOA-Technologien einordnen lassen. Die vorhandenen praktischen Ansätze zur Beschreibung nichtfunktionaler Eigenschaften und zur Auslieferung von Dienstpaketen müssen um eine modelltheoretische Sicht ergänzt werden, um darauf basierend eine Methodik zur Verbesserung der Qualität der Dienstauführung definieren zu können. Die Umsetzung einer der Methodik entsprechenden Dienstplattform soll hingegen vorrangig als konkrete Implementierung durchgeführt werden, um anschließend eine Evaluierung der Konzepte zu erzielen.

	abstrakt	konkret	
Dienst- plattform	--	WSQoSX NEXOF-RA WSMX ...	Atomarität der NFE-Angaben ↓
Dienst- pakete	TSM	WAR Rusco SAR ...	
Dokumente	--	WSDL WADL USDL WSML WS-Agreement ...	
Nichtfunktionale Eigenschaften	PCEM	WSAG-VT CQML+ WSML+PCM	

Abbildung 2.10: Schwerpunkte der Arbeit auf Modell- und Implementierungsebene

Kapitel 3

Methodik zur Qualitätsverbesserung

Dieses Kapitel beinhaltet die Darstellung des Konzepts zur inkrementellen Verbesserung der Metaqualität und damit der Gesamtqualität angebotener Dienstleistungen in Dienstumgebungen. Dafür wird zuerst eine Übersicht über den konzeptionellen Gesamtansatz gegeben, gefolgt von einer Vorstellung der zentralen Modelle, mathematischen Definitionen und algorithmischen Verfahren. Das Kapitel schließt mit einer Betrachtung der Nutzung der Verfahren in einer Dienstplattform.

Der Gesamtansatz beschreibt dabei die Gewinnung, Nutzung und inkrementelle Verbesserung der Metaqualität von Diensten (*Service Meta Quality, SMQ*). Aufbauend auf den im vorherigen Kapitel erarbeiteten Grundlagen zur Modellierung von NFE wird dazu ein spezielles Expressivitätsmodell für nichtfunktionale Eigenschaften präsentiert, welches als Berechnungsgrundlage für die Verfahren dient. Unter Nutzung dieses Modells wird eine eindeutige Definition für die Metaqualität von Diensten als hierarchisch strukturierte Metrik sowie als Ausprägung in Form von Annotationen gegeben. Die Gewinnung, Verwendung und Verbesserung der Metaqualitätsdaten wird anschließend als Menge von Prozessen dargestellt. Anhand eines Beispiels werden dabei die Vorteile beim Einsatz der Verfahren deutlich. Darauf folgt abschließend eine abstrakte Betrachtung zur Integration der Verfahren in Dienstplattformen.

Der Zusammenhang aller Teilkonzepte wird durch die Abbildung 3.1 veranschaulicht.

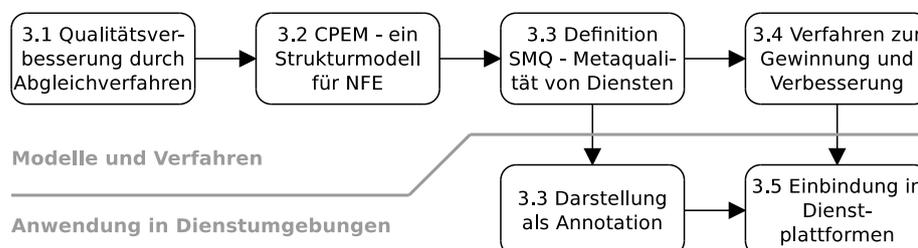


Abbildung 3.1: Erarbeitung der Teilkonzepte in diesem Kapitel

3.1 Inkrementelle Qualitätsverbesserung durch Abgleichverfahren

Dienste im Sinne dieser Arbeit sind durch deklarative Spezifikationen funktionaler und nichtfunktionaler Eigenschaften gekennzeichnet. Die Abgleichsverfahren haben zum Ziel, die Qualität der Spezifikation aller nichtfunktionalen Eigenschaften zu erfassen (*Gewinnung, Bestimmung*), abfragbar auszudrücken (*Rückführung, Propagierung*) und langfristig zu verbessern (*Verbesserung*).

Durch einen Abgleich von spezifizierten Qualitätsmerkmalen eines Dienstes mit einer Erwartungsvorlage einerseits und seinen tatsächlich erzielten Qualitätswerten andererseits wird die Qualität der Beschreibungen eines Dienstes bestimmt. Die Betrachtung dieser Metaqualität punktuell oder über die Zeit lässt Rückschlüsse auf zufällige Spezifikationsfehler oder inhärente Abweichungen durch unpassende Spezifikationsansätze zu. Eine Plattform zum Betrieb des Internets der Dienste mit integrierten Abgleichsverfahren kann systematisch die Metaqualität von Diensten bestimmen und somit langfristig die Markierung oder den Ausschluss der problematischen Dienste sicherstellen. Allein dadurch wird nicht nur die Qualität jedes einzelnen Dienstes in die Betrachtungen eingezogen, sondern im Hinblick auf verteilte komposite Dienstleistungen die Gesamtqualität jeder Plattforminstanz spürbar und messbar verbessert. Eine weitere Steigerung der Gesamtqualität ist möglich, wenn die Metaqualität nicht nur erkannt und propagiert, sondern selbst auch noch gesteigert wird. Die drei Verfahren zur Gewinnung, Rückführung und Verbesserung der Metaqualitätswerte werden durch die Grafik 3.2 anhand ihrer Anknüpfung an die Struktur von NFE veranschaulicht.

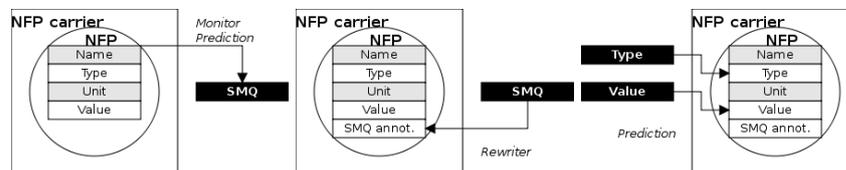


Abbildung 3.2: Prozessschritte der Metaqualitätsgewinnung, -rückführung und -verbesserung

Das Konzept zur inkrementellen Metaqualitätsverbesserung umfasst eine größtmögliche Einbeziehung von statischen und dynamischen Informationsquellen zur Laufzeit. Es werden dabei nichtfunktionale Diensteigenschaften aus Dienstbeschreibungen, Vertragsvorlagen, Monitoringinformationen zu aktiven Verträgen und Benutzerbewertungen berücksichtigt. Zusätzlich wird die Vollständigkeit und Präzision der Metaqualitätsangaben durch Einbeziehung von existierenden Datenabhängigkeits- und Vorhersagemodellen erhöht.

Die datenintensive Kombination von Angaben über Diensteigenschaften ist bisher in keiner Dienstplattform zu finden. Sie ist jedoch notwendig, um vor dem Hintergrund der zukünftig zu erwartenden Menge an angebotenen Diensten klare Entscheidungen für oder gegen die Nutzung bestimmter Dienste treffen zu können. Aus diesem Grund wird das Konzept erst detailliert dargelegt, anschließend dessen Einbindung in Dienstplattformen für das Internet der Dienste erläutert und schließlich anhand einer konkreten qualitätsfokussierten Dienstplattform experimentell validiert.

3.2. CPEM - EIN MODELL FÜR DIE AUSDRUCKSSTÄRKE VON NFE 49

Ein beispielhafter Prozess ist in der Abbildung 3.3 dargestellt. Es sind hierbei zwei NFE angedeutet, die zu unterschiedlichen Zeitpunkten in ein anfangs leeres Dienstbeschreibungsdokument eingetragen werden. Dabei weist insbesondere der zweite Wert eine hohe anfängliche Schwankung auf, verbleibt jedoch letztlich in geringer Abweichung seines spezifizierten Wertes. Für die beiden betrachteten Qualitätskriterien führt diese Stabilisierung zu anhaltend hohen Werten.

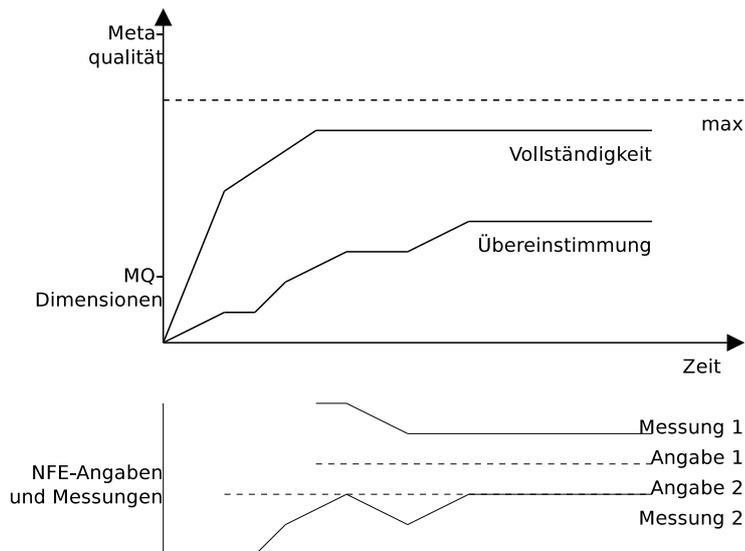


Abbildung 3.3: Iterative Verbesserung der Metaqualität als imaginäres Beispiel

3.2 CPEM - Ein Modell für die Ausdrucksstärke von NFE

Das *Property Expressivity Model* (PEM) definiert Expressivitätsniveaus zum Vergleich der Ausdrucksstärke von Spezifikationen nichtfunktionaler Eigenschaften. Es dient als Grundlage für die präzise Berechnung der Metaqualität.

Das erweiterte *Concentric Property Expressivity Modell* (CPEM) dient als grafische Notation für sowohl die Expressivität von NFE als auch deren Einordnung in den Lebenszyklus von Diensten. Es soll somit basierend auf den Vorarbeiten zu NFE-Strukturen und unter Einbeziehung der PEM-Stufen (siehe Abbildung 3.4) der Definition der Metaqualität vorangestellt werden.

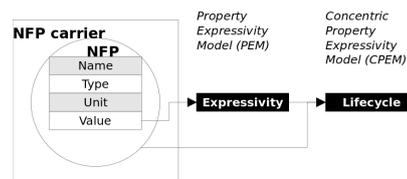


Abbildung 3.4: Ableitung expliziter Expressivitätsstufen von NFE

Die Modellierung von PEM und CPEM leitet sich aus dem Aufbau und der Verwendung handelbarer Dienste ab. Die mehrstufige Einordnung von NFE in solchen Diensten wird an dieser Stelle grob skizziert, um das Verständnis der nachfolgenden Erläuterungen zur Notation zu erleichtern. In Abbildung 3.5 wird gezeigt, wie sich ein Dienst aus verschiedenen abstrakten Artefakten a_n zusammensetzt. Als Artefakt zählen dabei neben der Implementierung auch beschreibende Dokumente, welche, sofern sie nichtfunktionale Diensteeigenschaften beschreiben, auch als NFE-Trägerdokumente bezeichnet werden. Dazu zählen insbesondere Dienstbeschreibungen und SLA-Vorlagen, die als Teil eines Dienstes bereitgestellt werden. Zusätzliche NFE-Trägerdokumente sind nicht notwendigerweise an ein physisches Artefakt des Dienstes gekoppelt. Dies betrifft vor allem nutzerseitige Trägerdokumente wie Zielanfragen, SLA-Vorschläge und Bewertungen sowie systemseitige Trägerdokumente wie Monitoringberichte und ausgehandelte SLAs. Als imaginäres NFE-Trägerdokument wird der konzentrische Trägerdokumentverbund a_X eingeführt, welcher in den einzelnen Phasen p_n der Dienstverwendung unterschiedliche Ausprägungen aufweist, die aus logisch zusammenhängenden Trägerdokumenten bestehen. Beispielsweise gibt es einen im Sinne der Dienstauführung logischen Zusammenhang der NFE-Angaben in einer Dienstbeschreibung, einer davon abgeleiteten SLA-Vorlage und einem daraus generierten SLA, nicht jedoch zwischen diesem und weiteren SLAs. Desweiteren können in jeder Phase die NFE-Angaben im Trägerdokument unterschiedlich ausdrucksstark mit der Notation e_n sein. Die entstehende grafische Notation soll nachfolgend definiert werden.

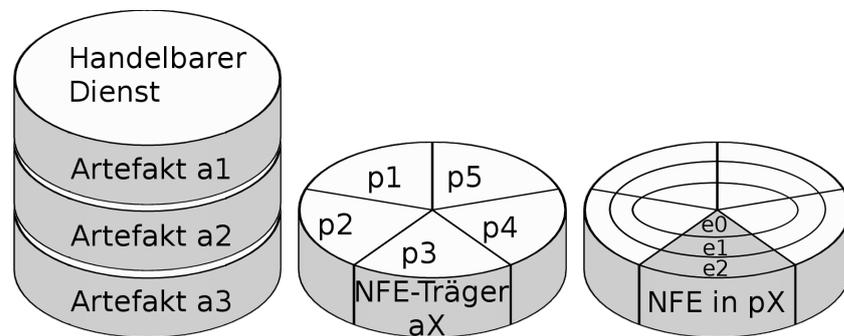


Abbildung 3.5: NFE-Modell CPEM: Zuordnung zu Artefakten eines handelbaren Dienstes, dessen Phasen und deren Ausdrucksstärken

3.2.1 Expressivitätsstufen in PEM

Die Expressivität oder Ausdrucksstärke einer NFE e_n gibt an, wieviele Informationen über den Wert der NFE und dessen Zusammensetzung direkt analytisch erkennbar sind. Ausgehend von Einzelwerten mit oder ohne statistischen Kennzahlen (e_1 bzw. e_{1+}) über gewichtete Wertebereiche (e_2) bis hin zu präzise erfassten, kontext-, eingabe-, ressourcen- und zeitabhängigen Merkmalen von Eigenschaften mit möglichen Interdependenzen zu anderen NFE (e_3) ist eine Analyse auf diese Weise vergleichbar möglich. PEM definiert häufig anzutreffende Expressivitätsstufen ohne Anspruch auf Vollständigkeit und ihre Überführbarkeit in jeweils angrenzende Stufen durch *Lifting* und *Lowering*.

Tabelle 3.1: Ausdrucksstärken von NFE im PEM

Expressivität	Erläuterung
e_0	fehlende Angabe
e_1	Einzelwerte
e_{1+}	Einzelwerte mit statistischen Kennzahlen, insbesondere Lage- und Streuungsmaße
e_2	gewichtete Wertebereiche
e_3	Abhängigkeit von Eingabe- und Kontextparametern, Ressourcenbindung, temporale Abhängigkeiten, Interdependenzen - extrinsische Eigenschaften

Tabelle 3.2: Kompatibilität unterschiedlicher Ausdrucksstärken

Expressivität	Lifting zu e_{n+1}	Lowering zu e_{n-1}
e_0	nicht möglich aufgrund fehlender Information	–
e_1	nicht möglich aufgrund fehlender Information	trivial durch Löschung
e_{1+}	Aufteilung auf mehrere Wertebereiche möglich, abhängig von statistischen Kennzahlen	Beschränkung auf Durchschnittswert, Verlust der Semantik dieses Wertes
e_2	nicht möglich, Problem der Dekomponierbarkeit von Wertreihen	fuzzy-Ableitung der Statistik aus Wertebereichen
e_3	–	Berechnung der Wertebereiche

Abzuwägende Faktoren für die Wahl einer Expressivität sind neben der aufwändigeren Verarbeitung höherer Stufen auch die damit einhergehenden performanzreduzierenden umfangreicheren Spezifikationen in NFE-Trägerdokumenten.

Von PEM werden in dieser Arbeit neben vollständig fehlenden Angaben nur die beiden unteren Expressivitätsstufen e_1 und e_2 betrachtet, da diese für die meisten Anwendungsfälle bereits hinreichend aussagekräftig sind. Diese sollen nachfolgend als *Einzelwerte* und *Wertebereiche* bezeichnet werden. Die Unabhängigkeit der NFE voneinander und von äußeren Einflüssen wird angenommen. Tabelle 3.1 fasst die definierten Ausdrucksstärken zusammen.

Die Berücksichtigung der Expressivitätsniveaus ist für sämtliche Vergleichsverfahren von hoher Bedeutung. Nicht jede Ausdrucksstärke lässt sich auf die jeweils höhere oder niedrigere Stufe abbilden, so dass gerade in heterogenen Dienstumgebungen Kontrollverfahren eingesetzt werden müssen, um die fehlende Vergleichbarkeit zu signalisieren. Die Kompatibilität der Niveaus hinsichtlich der Vergleichbarkeit zeigt Tabelle 3.2.

3.2.2 Grafische Notation für CPEM

CPEM erweitert PEM um kausale Zusammenhänge anhand des Lebenszyklus handelbarer Dienste. Die Expressivität oder Ausdrucksstärke einer NFE e_n wird in der grafischen Notation durch den symbolischen Radius n eines Rings bestimmt, wobei der Kreismittelpunkt eine völlig fehlende Angabe e_0 repräsentiert.

tiert. Je größer der dargestellte Radius ist, desto detaillierter ist die Angabe der NFE.

Die Einordnung in den Lebenszyklus erfolgt durch Kreissektoren, deren Anzahl sich aus der Zahl der NFE-Verarbeitungsschritte p_n auf einer Dienstplattform ergibt. Durch die Analyse der Vorarbeiten wurden fünf allgemeine Schritte identifiziert, so dass sich eine Einteilung in Quintupel ergibt. Das erste Quintupel beschreibt die Angabe von NFE in Dienstbeschreibungen (*Service Descriptions, SD*), die üblicherweise von den Anbietern eines Dienstes bereitgestellt werden. Das zweite Quintupel beschreibt die NFE-Angaben innerhalb von SLA-Vorlagen (*SLA Templates, SLAT*), die oftmals von den Dienstbeschreibungen abgeleitet sind. Im dritten Quintupel werden die Anforderungen und Zielstellungen eines potenziellen Dienstbenutzers über ihre NFE dargestellt (*Goal*). Das vierte Quintupel repräsentiert die NFE-Angaben in den Dienstgütevereinbarungen eines Vertrages, der durch eine Aushandlung entstanden ist (*SLA*). Der Aushandlungsprozess kann dabei die NFE aus der Vertragsvorlage und den Anforderungen nutzen und resultiert zuerst in einem *SLA Offer*. Dieses ist jedoch bezüglich der NFE-Angaben semantisch äquivalent zum SLA und muss somit nicht als gesonderte Phase betrachtet werden. Das fünfte Quintupel enthält schließlich die gemessenen NFE basierend auf den vertraglichen Zusicherungen (*Monitor, Mon*). Ausgehend von den Messungen können über eine Rückkopplung nun wiederum die NFE-Angaben in den Dienstbeschreibungen und SLA-Vorlagen aktualisiert oder annotiert werden.

Das Modell CPEM ist in der Abbildung 3.6 dargestellt. Die weiter außen befindlichen Ringe e_{3+} mit höherer Expressivität sind in der Abbildung nicht enthalten. Die grafische Notation erlaubt eine Darstellung der in der jeweiligen Nutzungsphase der NFE-Trägerdokumente voraussetzbaren Expressivität und eine intuitive Erkennung von Übergängen.

Um die NFE in ihren Phasen unterscheiden zu können, wird eine Notation mit Indizes eingeführt. Somit stellen nfe_{sd} die NFE in Dienstbeschreibungsdokumenten dar, nfe_{slat} solche in Vertragsvorlagen, nfe_{goal} solche in Zielstellungen des Konsumenten, nfe_{sla} solche in Dienstgütevereinbarungen und nfe_{mon} die durch ein Monitoringframework gemessenen Werte.

Jede einzelne NFE wird dabei als zusammengesetzte, annotierbare Struktur angenommen (vgl. Abschnitt 2.3.3 und Abbildung 2.7). Die Annotierbarkeit wird für Benutzerbewertungen auf nfe_{sla} und Metaqualitätsangaben zu nfe_{sd} genutzt. Beide Angaben können aufgrund abweichender Basistypisierung und ebenfalls unterschiedlicher Einheiten nicht als eigene Phase in CPEM repräsentiert werden. Sie sind demnach nur als Ergänzung in der grafischen Notation dargestellt.

Die Eigenschaften nfe_{slat} und nfe_{sla} werden oftmals auch als Service Level Objectives (SLO) oder Service Level Specifications (SLS) bezeichnet. Die Angaben nfe_{sd} und nfe_{goal} entsprechen dabei den Angaben *offers/provides* und *requires/uses* in Ansätzen zur NFE-Repräsentation in Komponentensystemen. Das Modell CPEM geht somit über diese bisherige Darstellung hinaus und ermöglicht durch eine generalisierte Darstellung die Bildung von vergleichbaren NFE in Dienstlandschaften.

CPEM definierte keine syntaktische Darstellung der NFE. Diese ist abhängig von den heterogenen Dokumentenformaten und deren Varianten wie im Fall der SLA-Sprache WS-Agreement, welche gerade die präzise Angabe der garantierten Eigenschaften nicht spezifiziert. In dieser Arbeit wird daher stets mit

3.2. CPEM - EIN MODELL FÜR DIE AUSDRUCKSSTÄRKE VON NFE 53

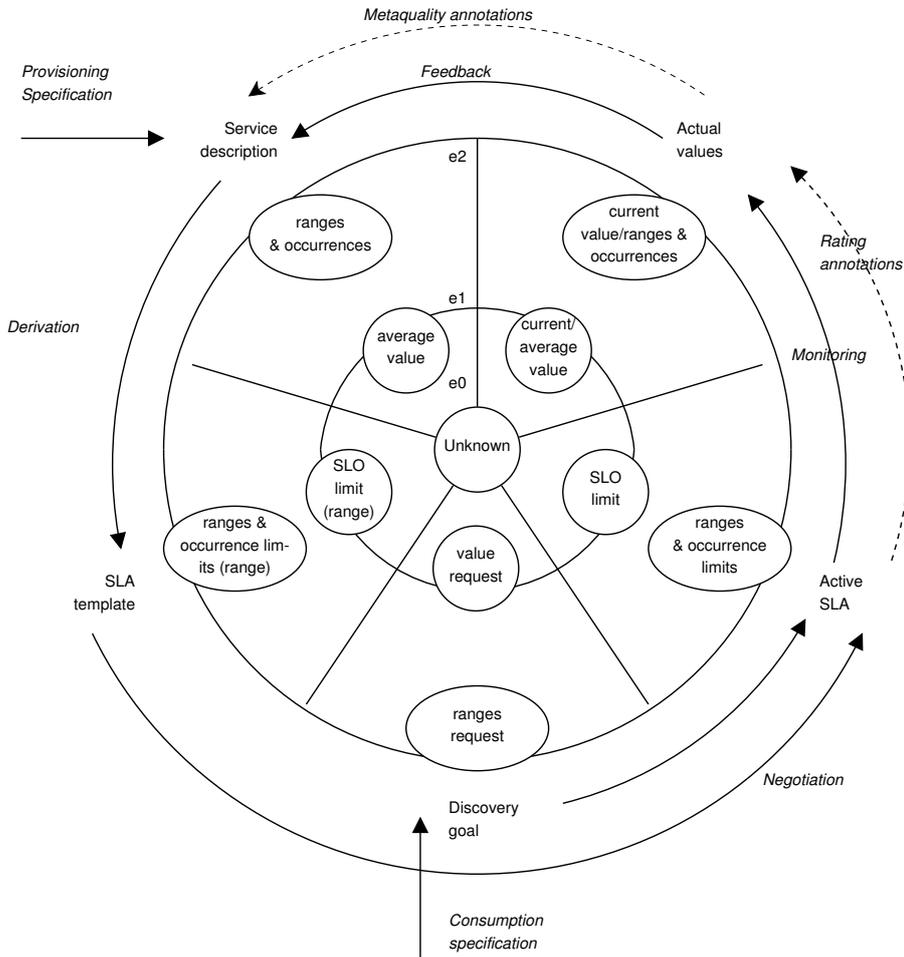


Abbildung 3.6: NFE-Modell CPEM: Expressivität und Lebenszykluseinordnung für nichtfunktionale Eigenschaften

einer adäquaten Pseudo-Notation gearbeitet, sofern keine konkrete Syntax zur Verfügung steht.

3.2.3 Beispiel für Ausdrucksstärken nach CPEM

Die unterschiedliche Expressivität der NFE lässt sich mit einem Beispieldienst demonstrieren. Dieser sei bezüglich der zwei Eigenschaften Antwortzeit und Aufruf Erfolg beschrieben. Die Antwortzeit (*rt* für *response time*) ist dabei ein mit einer Einheit versehener numerischer Wert, der angibt, wieviele Sekunden der Aufruf eines Dienstes benötigt. Der Aufruf Erfolg (*success*) ist ein einheitenloser Wahrheitswert für den erfolgreichen Aufruf ohne Dienst- und Übertragungsfehler.

Im Fall einfacher Wertangaben über Einzelwerte mit teilweiser Nutzung statistischer Merkmale wie dem arithmetischen Mittel μ (*average*) und der Standardabweichung σ (*sigma*) sind die folgenden NFE-Spezifikationen in e_1 und e_{1+} möglich:

- nfe_{sd} : success=true, rt=6.7s
- nfe_{slat} : success=[true..false], rt<[5..10]s
- nfe_{goal} : success=true, rt<4s
- nfe_{sla} : success=true, rt<5s
- nfe_{mon} : success:median=true, rt:average=6.7s, rt:sigma=2.9s

Auffällig sind dabei die typabhängig unterschiedlich sinnvoll einsetzbaren Verfahren zur Bildung von Durchschnittswerten. Vor allem Wahrheitswerte lassen sich nur bedingt nützlich als aggregierte Werte darstellen, da dieser Angabe unter Beachtung der Typsicherheit wiederum nur ein Wertebereich der Kardinalität zwei zur Verfügung stünde. Bei Verwendung von gewichteten Wertbereichsangaben unter Angabe der relativen Auftrittshäufigkeiten sind wesentlich genauere Aussagen über die Diensteseigenschaften in e_2 erzielbar:

- nfe_{sd} : success=true(98%), rt=[5;9]s(95%)/[9;∞]s(5%)
- nfe_{slat} : success=true([>=<][70..100]%), rt=[5;9]s([>=<][80..100]%)
- nfe_{goal} : success=true(>80%), rt=[5;9]s(>90%)
- nfe_{sla} : success=true(>=94%), rt=[5;8]s(>85%)
- nfe_{mon} : success=true(98%), rt=[5;8]s(87%)

Üblicherweise stehen für die Vertragsaushandlung beliebige Vergleichsoperatoren ($<$, \leq , $=$, \geq , $>$) zur Verfügung. Die Angabe des Operators für nfe_{goal} und nfe_{sla} ergibt sich hier aus der aus Sicht des Anfragers sinnvollen Sortierung.

Die in SLAs oft vorkommende SLO-Formulierung, dass eine NFE einen Grenzwert höchstens zu einem bestimmten Prozentsatz unter- bzw. überschreiten dürfe, ist ein binär vereinfachter Spezialfall der Wertbereichsangaben.

3.3 Metaqualität von Diensten

Ausgehend vom allgemeinen Begriff Metaqualität (vgl. 2.4.4) lässt sich ein dienstspezifischer Begriff ableiten. Diese Dienstmetaqualität (*Service Meta Quality*, SMQ) ist strukturell auf die Angabe von Qualitätseigenschaften in diensttarfaktgebundenen NFE-Trägerdokumenten wie Dienstbeschreibungen und Vorlagen für Dienstgütevereinbarungen gemäß der CPEM-Notation angepasst. In Abbildung 3.7 ist diese spezialisierte Form der Metaqualität unter Berücksichtigung mehrerer Granularitätsstufen schematisch abgebildet.

In den weiteren Unterabschnitten wird die Metaqualität für Dienste definiert, ihre mathematische und algorithmische Berechnungsvorschrift angegeben und ihre Manifestierung in NFE-Trägerdokumenten exemplarisch gezeigt.

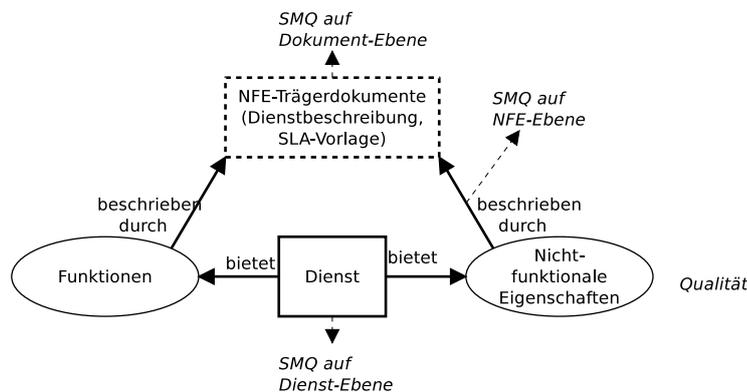


Abbildung 3.7: Einordnung der Metaqualität von Diensten

3.3.1 Motivation

Die Angabe von NFE in Dienstbeschreibungen, SLA-Vorlagen und SLAs kann über unterschiedliche Verfahren realisiert werden. Sie reicht von der willkürlichen Festlegung basierend auf Abschätzungen über projizierte Simulationsergebnisse bis hin zu formal bestimmten Parametergrenzen. Typischerweise sind der Ursprung und die Qualität der NFE-Angaben jedoch unbekannt und werden nicht kritisch untersucht. Die Metaqualität für Dienste, *Service Meta Quality* (SMQ), soll die damit verbundenen Nachteile ausschließen.

Je mehr dienstorientierte Anwendungslandschaften von bestimmten NFE abhängen, desto wichtiger wird es zu wissen, was genau mit deren Spezifikation ausgedrückt wird. Bisherige Forschungsergebnisse verlangen bereits eine hohe Präzision und Expressivität von NFE sowie deren Komponierbarkeit und effiziente automatische Berechnung und Validierung [128]. Auch sollen neben technischen Eigenschaften nutzerzentrische Aussagen über Dienste ausgedrückt werden können. Im Kontext der vertragsgebundenen Dienstaufführung ist es essentiell, dass sich der Dienstanutzer auf Zusicherungen bezüglich der Dienstgüte verlassen kann. Diese Zusicherungen verlangen wiederum eine präzise und akkurate Spezifikation der nichtfunktionalen Eigenschaften in Dienstbeschreibungen, Vertragsvorlagen und anderen NFE-Trägerdokumenten.

Wie im Abschnitt zu verwandten Arbeiten zur Dienst- und Metaqualität dargelegt sind bisher keine Ansätze bekannt, die eine Unschärfe, Unsicherheit oder sonstige ungenügende Qualität von NFE für die Vertragsbildung, die Dienstaufwahl und das Hosting von Diensten berücksichtigen. Als Lösung soll dafür die Metaqualität von Diensten definiert werden.

3.3.2 Definition

Die Metaqualität von Diensten, oder SMQ, ist eine komposite Qualitätsmetrik, die sich auf die Angabe von QoS-Eigenschaften und anderen NFE in den in Dienstlandschaften genutzten dienststartefaktgebundenen NFE-Trägerdokumenten wie Dienstbeschreibungen und SLA-Vorlagen bezieht. SMQ definiert eine hierarchisch gebildete skalare Metrik, die auf jeder Stufe aus teils mehrdimensionalen Teilmetriken gebildet wird. Die Metriken sind jeweils auf den Prozentbe-

reich 0% bis 100% und somit auf den Bereich $[0..1]$ normiert. Für jedes NFE-Trägerdokument wird SMQ als Matrix definiert, die für jede NFE-Angabe deren Qualitätseigenschaften als Zeilenvektor beschreibt. Hierbei steht jede Komponente des Vektors für eine Datenqualitätsdimension (vgl. Aufzählung in Abschnitt 2.4.3), also beispielsweise die Aktualität, Vollständigkeit, Brauchbarkeit und Effizienz der Angabe der nichtfunktionalen Eigenschaft.

Die umfassende Betrachtung sämtlicher Vektorkomponenten übersteigt die Anforderungen dieser Arbeit. Als Einschränkung soll mit SMQ nunmehr jeweils der Teilvektor bezeichnet werden, welcher für eine hochqualitative Dienstauführung im Internet der Dienste den größten Beitrag liefert. Das Schema 3.8 fasst dazu häufig auftretende Qualitätsprobleme als Abstände zwischen jeweils zwei NFE-Angaben zusammen. Neben (1) fehlenden oder anderweitig unterspezifizierten NFE-Angaben sind dies bei Voraussetzung einer hinreichend nutzbaren Spezifikation (2) Abweichungen von den tatsächlich erbrachten NFE und (3) interne Abweichungen innerhalb der möglicherweise aus mehreren Dokumenten zusammengesetzten Spezifikation für eine NFE eines Dienstes.

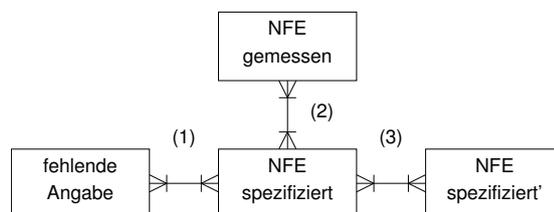


Abbildung 3.8: Semantik der ausgewählten Qualitätsdimensionen als Abstandsmaße zwischen NFE-Spezifikationen

SMQ wird demnach als Tupel definiert, welches sich aus einer extern beobachtbaren und statisch analysierbaren Dimension (SMQ-Coverage) und einer dynamisch messbaren Dimension bei entsprechender Plattformunterstützung (SMQ-Validity) zusammensetzt. Damit werden bezüglich der qualitativen Beurteilung von Qualitätsangaben zwei Qualitätsdimensionen, nämlich Vollständigkeit und Korrektheit, abgedeckt. Die Präzision als weitere Dimension wird nicht bewertet, sondern durch die Nutzung von CPEM implizit berücksichtigt. Die Konsistenz wird ebenfalls nicht bewertet, Inkonsistenzen lassen sich jedoch von auftretenden Unterschieden zwischen SMQ-Bewertungen eines Dienstes ableiten. Andere Dimensionen wie Aktualität oder Objektivität werden nicht berücksichtigt, da die dafür notwendigen Zusatzinformationen zumeist nicht im System verfügbar sind. Die durch die Dissertation geleistete Abdeckung der Qualitätsdimensionen bezüglich der Angabe nichtfunktionaler Eigenschaften wird durch Abbildung 3.9 zusammengefasst.

Die Metrik SMQ-Coverage gibt an, wieviel Informationen über die NFE eines Dienstes abrufbar sind, beispielsweise durch Dienstbeschreibungen oder Vertragsvorlagen. Diese Informationen werden dabei mit einem Anforderungskatalog AK verglichen. Die Abdeckung wird entweder binär verlangt (ja/nein) oder reicht von fehlenden Informationen über einzelne Durchschnittswerte bis hin zu komplexen stochastischen Verteilungen mit Berücksichtigung temporaler und eingabeabhängiger Einflüsse gemäß der Expressivität in CPEM. Dabei enthält AK keine Zielspezifikation zu den konkreten Werten der Eigenschaften wie etwa nfp_{goal} , sondern dient als Vorstufe zur Gewährleistung einer möglichst

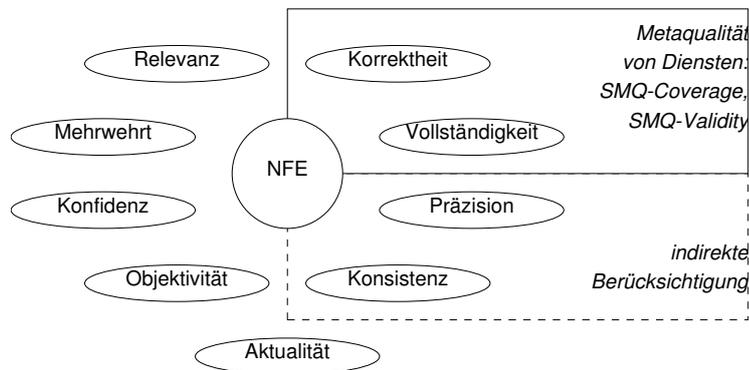


Abbildung 3.9: Durch SMQ berücksichtigte Qualitätsdimensionen von NFE

breiten Anwendbarkeit einer Zielanfrage auf alle Dienste.

Die Metrik SMQ-Validity basiert auf Messungen und Berechnungen zum Abgleich der angegebenen NFE mit den tatsächlichen Eigenschaften eines Dienstes. Es baut daher direkt auf den Festsetzungen von SMQ-Coverage auf und erzeugt einen Eindruck von den vorliegenden Unzulänglichkeiten in der Spezifikation von NFE. Dabei können neben den Ergebnissen des Monitorings zur Laufzeit auch Vorhersagetechniken und Benutzerbewertungen (Ratings) einbezogen werden. SMQ-Validity ist aus diesem Grund nur auf messbare Eigenschaften anwendbar. Darin enthalten sind die meisten QoS-Eigenschaften, nicht jedoch statische Metadatenangaben wie die Autorenschaft nach Dublin Core.

Als Bildungsvorschrift der Gesamtmetriek SMQ wird vorgeschlagen, das arithmetische Mittel aus SMQ-Coverage und SMQ-Validity zu bilden. Abhängig vom Einsatzzweck kann aber auch eine gewichtete Bildung oder eine Erweiterung um die in dieser Definition ausgelassenen Qualitätsdimensionen erfolgen. Die formale generische Bildungsvorschrift wird im folgenden Unterabschnitt angegeben.

Die beiden Qualitätsdimensionen von SMQ sowie die erreichte Expressivität können durch eine grafische Darstellung in einem zweidimensionalen Koordinatensystem veranschaulicht werden. Derartige Illustrationen bieten einen schnellen visuellen Überblick über die Metaqualität von Diensten und zugehörigen nichtfunktionalen Eigenschaften. Beispiele werden in der Abbildung 3.10 gezeigt.

Die Metaqualität kann auf unterschiedlichen Granularitätsstufen angegeben werden. Jede in einem Dienstbeschreibungsdokument erfasste oder ausgelassene NFE kann auf der untersten Ebene bewertet werden. Das Beschreibungsdokument kann anschließend eine aggregierte Bewertung erhalten. Sämtliche Beschreibungsdokumente eines Dienstes zusammen ergeben anschließend die auf höchster Ebene aggregierte SMQ-Angabe für den Dienst. Die Gesamtheit aller Dienste benötigt schließlich eine Dienstplattform, welche mit Mechanismen zur Verarbeitung von Metaqualitätsdaten ausgestattet ist. Die konkrete Berechnung der SMQ auf der untersten Ebene und die Aggregation zu höherwertigen Metriken wird im nachfolgenden Abschnitt detailliert erläutert.

Die Anwendbarkeit der SMQ-Teilmetriken auf NFE in ihren Trägerdokumenten wird durch Grafik 3.11 herausgestellt. Neben Dokumentformaten mit rein funktional-syntaktischen Angaben wie WSDL existieren auch solche, die nur

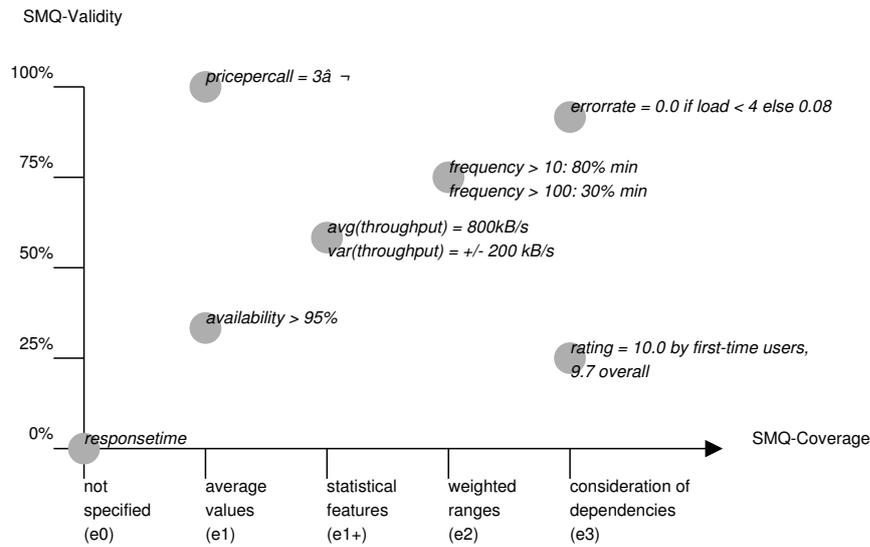


Abbildung 3.10: SMQ-Diagramm mit beispielhaften SMQ-Resultaten

nichtfunktionale oder gemischte Angaben enthalten. Dazu gehören SAWSDL, WSMML und USDL. Die nichtfunktionalen Eigenschaften lassen sich in quantitative und nichtquantitative Metadaten unterteilen, und die quantitativen wiederum in automatisch überprüfbare und nicht oder nur mit unverhältnismäßig hohem Aufwand überprüfbare Eigenschaften. Zur Vereinfachung werden die überprüfbaren Eigenschaften mit QoS-Eigenschaften gleichgesetzt, auch wenn es in Grenzfällen definitorische Unterschiede gibt, etwa bezüglich der Nichtüberwachbarkeit von Ende-zu-Ende-Eigenschaften. Die nichtüberwachten Eigenschaften betreffen somit numerische, boolesche oder sonstige quantitative Angaben zum Kontext, zum Preis und zur Reputation eines Dienstes, die zusammengefasst als CPR-Angaben bezeichnet werden sollen. Schließlich fallen Autorangaben oder Versionsangaben unter die nichtquantitativen Metadaten. Diese Einordnung erweist sich als günstig für die Abgrenzung der Anwendbarkeit der Metaqualitäts-Verfahren. Dafür werden hiermit die Bezeichnungen **QoS-NFE**, **CPR-NFE** und **Metadata-NFE** eingeführt und an den betreffenden Stellen referenziert.

3.3.3 Berechnungsvorschrift

In diesem Abschnitt erfolgt die mathematisch-algorithmische Definition der Berechnungsvorschrift für die Metaqualität SMQ. Zuerst wird nach einführenden mathematischen Vorbetrachtungen zu expressivitätsabhängigen Abstandsmaßen eine generische Vorschrift für die Berechnung der SMQ einer einzelnen nichtfunktionalen Eigenschaft über alle Qualitätsdimensionen gegeben, wobei die Algorithmen für SMQ-Coverage und SMQ-Validity erläutert werden. Anschließend erfolgt die Betrachtung der matrizengestützten Aggregation der SMQ-Werte auf Dokumentenebene sowie die weitere hierarchische Aggregation bis hin zur SMQ von Dienstportalen.

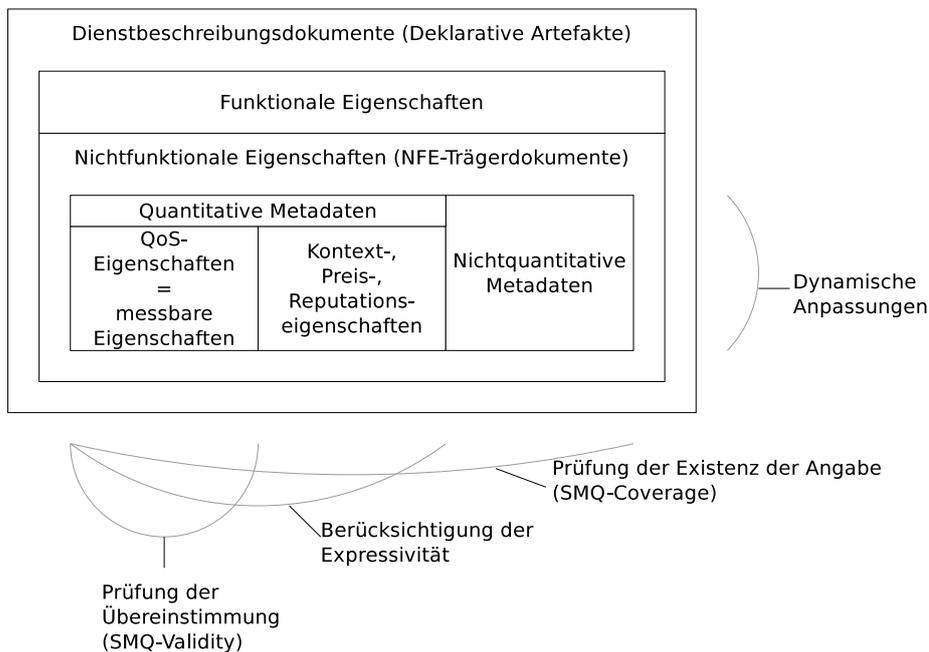


Abbildung 3.11: Anwendbarkeit der Metaqualität auf Beschreibungsdokumente

Abstandsmaße nach Expressivitätsniveaus

Die Metaqualität für Dienste wird auf der Ebene der NFE je nach deren Expressivität wie folgt berechnet.

1. Nicht vorhandene NFE, Expressivität e_0 : Es kann nur eine statische Analyse mit dem Ergebnis nicht vorhandener Angaben von **QoS-NFE**, **CPR-NFE** und **Metadata-NFE** durchgeführt werden. Ist die NFE-Angabe allerdings erforderlich, wird die SMQ-Coverage für die NFE auf 0 gesetzt, jene für den Dienst wird folglich zumindest reduziert.
2. Einzelwerte, Expressivität e_1 : Die Qualität der Angaben im Sinne der SMQ-Validity wird durch das Abstandsmaß zwischen gemessenen und angegebenen Werten für alle **QoS-NFE** bestimmt. Hierbei wird entweder der Durchschnittswert für alle Messungen oder die mittlere quadratische Abweichung als Berechnungsgrundlage herangezogen. Die SMQ-Coverage wird bei einer entsprechenden Erfordernis für die NFE auf 1 gesetzt, jene für den Dienst dadurch folglich erhöht.
3. Wertebereiche, Expressivität e_2 : Ein Abstandsmaß für die SMQ-Validity von **QoS-NFE** wird analog zu den Einzelwerten, jedoch für jeden Wertebereich richtungsabhängig kumulativ ermittelt. Dabei bestimmt die Richtung eine Bevorteilung des Nutzers bzw. des Anbieters, falls dies etwa durch Grenzwerte in Dienstbeschreibungen und Verträgen vorgesehen ist. Für eine objektive Bestimmung entfällt die Kumulation. Die Bestimmung der SMQ-Coverage verläuft unabhängig von der Expressivität und somit analog zum Vorhandensein von Einzelwerten.

Beispielhaft sei an dieser Stelle die Bestimmung der SMQ-Validity für eine über Wertebereiche garantierte NFE Antwortzeit in Abbildung 3.12 dargestellt.

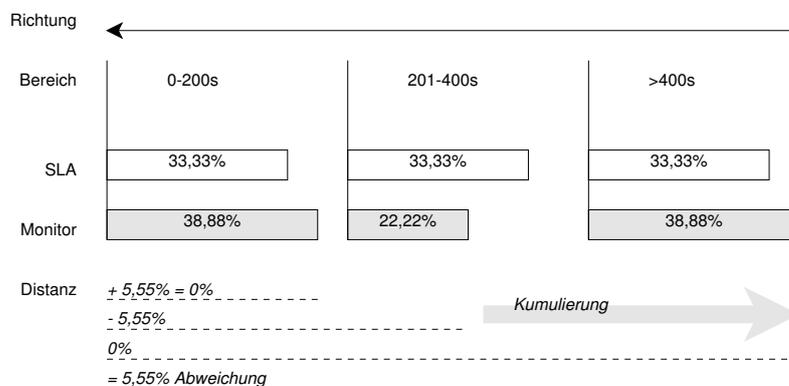


Abbildung 3.12: Berechnungsbeispiel für Wertebereiche

Restriktionen der Berechnung der Abstandsmaße

Die Berechnungsformel für Abstandsmaße muss dabei mehrere Besonderheiten berücksichtigen, um unabhängig von den Berechnungsumständen ihre Konsistenz zu behalten. Die beiden wichtigsten Besonderheiten werden an dieser Stelle als *Berücksichtigung der Übererfüllung* und *Berücksichtigung der Mittelwertbildung* bezeichnet.

Übererfüllung Für NFE-Angaben in SLAs sowie sonstige gerichtete Angaben gilt, dass die garantierten Grenzwerte in die Berechnung so einbezogen werden müssen, dass eine Übererfüllung der Garantien sich nicht negativ auf die Metaqualität auswirkt. Für Einzelwerte lässt sich dies über gerichtete Schranken zusichern. Beträgt die Spezifikation einer NFE in e_1 beispielsweise 0,5 mit der implizierten Richtung \leq und dem beobachteten Wert 0,4, so beträgt der Abstand gerichtet $-0,1$ und wird auf 0 beschränkt, wodurch die SMQ maximiert wird. Beträgt der beobachtete Wert hingegen 0,6, so wird das Abstandsmaß ausgehend vom absoluten Abstand 0,1 berechnet. Für Wertebereiche empfiehlt sich hingegen eine kumulative Aufrechnung, so dass übererfüllte Werte unabhängig von ihrer Verteilung nur in der Summe den spezifizierten Bereichswahrscheinlichkeiten genügen müssen. Lautet die Spezifikation beispielsweise, dass 50% aller Werte zwischen 0,0s und 0,4s liegen, weitere 40% zwischen 0,4s und 0,8s und nur 10% schlechter als 0,8s, so würde ohne Berücksichtigung der Übererfüllung bei Beobachtung von 60% Auftrittshäufigkeit im unteren und 35% im mittleren Intervall der Abstand im mittleren Intervall ungleich Null und auch nicht übererfüllt sein. Durch iterative Kumulation beträgt die Häufigkeit im ersten und zweiten Intervall jedoch 95% und übererfüllt damit den spezifizierten Wert von 90%. Die unteren Intervalle dienen dabei als Entlastung für das jeweils nachfolgende, womit das letzte Intervall stets minimiert werden muss, da kein weiteres folgt. Der Fall der Existenz zweier Bereiche, etwa für spezifizierte Auftrittswahrscheinlichkeiten boole-

scher Variablen, ist demnach trivial und von dieser Berücksichtigung nicht betroffen.

Mittelwertbildung Das Abstandsmaß der SMQ-Validity bezieht sich stets auf die Spezifikation der NFE im Vergleich zu einem beobachteten Referenzwert, der sich aus Monitoring-, Vorhersage- und Bewertungsinformationen zusammensetzt. Wird der Referenzwert nicht aus der jeweils letzten Beobachtung, sondern aus einer chronologischen Reihe dieser Informationen berechnet, wie bei der Verwendung eines einfachen gleitenden oder exponentiell geglätteten Mittelwerts, so ist für jeden Wert der Reihe die SMQ-Validity separat zu bilden, da andernfalls ein Informationsverlust die Folge wäre. Liegen nach drei Messungen beispielsweise die absoluten Abstände $d_0 = 0$, $d_1 = 0,3$ und $d_2 = -0,4$ vor, so wäre im Schnitt nach Berücksichtigung der Übererfüllung ein Abstand von 0 ausschlaggebend, obwohl möglicherweise eine SLA-Verletzung vorliegt. Werden hingegen die SMQ-Validity-Werte nachträglich wie die Messwerte zusammengeführt, so wäre unabhängig von der konkreten Mittelwertbildung ein Abstandsmaß verschieden von Null garantiert.

Berechnung der SMQ einer NFE als Abstandsmaß

Die Metaqualität von Diensten SMQ auf Ebene der nichtfunktionalen Eigenschaften liegt zwischen 0 im schlechtesten und 1 im optimalen Fall:

$$SMQ \in \mathbb{R}, 0 \leq SMQ \leq 1 \quad (3.1)$$

Für SMQ sei ein Vektor $s\vec{m}q$ mit n Komponenten smq_{QD} zur Repräsentation der Abstandsmaße in den jeweiligen Qualitätsdimensionen sowie ein korrespondierender Vektor \vec{w} mit ebenfalls n Komponenten zur Gewichtung gegeben.

Es gilt stets die Normierung der Gewichtung: $|\vec{w}| = 1$. Im vereinfachten Fall der gleichen Gewichtung gilt darüber hinaus: $\forall i, 1 \leq i \leq n : w_i = n^{-1}$

SMQ wird folglich als Skalarprodukt definiert:

$$SMQ = \langle s\vec{m}q, \vec{w} \rangle \quad (3.2)$$

Eine algorithmische Umsetzung kann effizient über die Map-Reduce-Methoden [34] der funktionalen Programmierung im λ -Kalkül erreicht werden:

$$SMQ = reduce(\lambda xy.x + y, map(\lambda xy.x * y, s\vec{m}q, \vec{w})) \quad (3.3)$$

Die Berechnung der Einzelwerte smq_{QD} wird individuell je nach Expressivität definiert. Jedes Abstandsmaß für Einzelwerte bildet sich aus dem Quotienten des gerichteten absoluten Abstands zwischen dem spezifizierten Wert nfe_{spec} und einem Referenzwert nfe_{ref} und der typgebundenen Definitionsbereichslänge der nichtfunktionalen Eigenschaft. Der absolute Abstand wird dabei als Differenz zwischen dem beobachteten und spezifizierten Wert so berechnet, dass eine Übererfüllung zu einem positiven Ergebnis führt, was durch einen Multiplikator dir erreicht wird. Der Multiplikator ist für implizit aufsteigend gerichtete NFE, also für die Operatoren $>$ und \geq , -1 , sonst 1 . Die resultierende Formel lautet damit:

$$smq_{QD} = \frac{dir * (nfe_{ref} - nfe_{spec})}{\|range(nfe)\|} \quad (3.4)$$

Listing 3.1: Berechnung der Teilmetrik SMQ-Coverage

```

1 def smqcoverage(nfpregs, nfpsds):
2     smqcov = 0.0
3     prios = 0.0
4     foreach nfpreg in nfpregs:
5         nfpsd = identify(nfpsds, nfpreg)
6         cov = covered(nfpreg, nfpsd)
7         if nfpreg.prio == required and not cov:
8             return 0.0
9         else:
10            smqcov += cov * nfpreg.prio
11            prios += nfpreg.prio
12    if prios == 0.0:
13        return 0.0
14    smqcov /= prios
15    return smqcov

```

Ist $smq_{QD} < 0$, dann gilt die Übererfüllung. Ist $nfe_{ref} \notin range(nfe)$, kann dies als Hinweis auf eine ungenügende Definitionsbereichsspezifikation angesehen werden. Andernfalls ist $smq_{QD} \leq 1$ garantiert.

Für Wertebereiche wird das Abstandsmaß als Durchschnitt aller kumulativen Abweichungen auf der Prozentskala definiert. Da kein einzelner Abstand $> 100\%$ sein kann, ist $smq_{QD} \leq 1$ garantiert.

Als Referenzwerte nfe_{ref} auf Dienstplattformen gelten dynamisch gesammelte Monitoringdaten nfe_{mon} und statische Einträge nfe_{req} im Anforderungskatalog *AK*.

Algorithmus zur Berechnung der SMQ-Coverage

Der Anforderungskatalog zur Überprüfung des Vorhandenseins von NFE-Angaben in den Dienstbeschreibungen wird als Vektor von nicht wertbehafteten NFE repräsentiert. Diese sollen im weiteren Verlauf als nfp_{req} bezeichnet werden. Ihre Struktur ist demnach eine Untermenge von nfp_{goal} . Sie enthalten durch den Wegfall der Wertangabe noch den Identifikator sowie eine Priorisierung im Bereich von 0-100%. Ein gesonderter Prioritätswert *required* sorgt dafür, dass ein Fehlen einer damit priorisierten NFE sofort zu einer SMQ-Coverage von 0 führt, um die betroffenen Dienste gezielt vorbehaltslos ausschließen zu können.

Die Berechnung der SMQ-Coverage erfolgt durch gewichtete Durchschnittsbildung über alle angeforderten NFE und die zugehörigen spezifizierten NFE wie im nachfolgenden Pseudocode 3.1 algorithmisch dargestellt. Die Übereinstimmung zwischen NFE wird über die Funktion *covered* als zentrale Routine unter Nutzung von Identifikatoren und Definitionsbereichen der NFE-Typen festgestellt.

Algorithmus zur Berechnung der SMQ-Validity

Die Berechnung der Abstandsmaße erfolgt ohne Darstellung der Gewichtung nach dem im Quelltext 3.2 abgebildeten Schema. Dabei dienen für jedes NFE-

Listing 3.2: Berechnung der Teilmetrik SMQ-Validity

```

1 def smqvalidity(nfpsds, nfpmons):
2     counter = 0
3     foreach nfpsd in nfpsds:
4         nfpmon = identify(nfpmons, nfpsd)
5         if nfpmon:
6             nfpval = dist(nfpsd, nfpmon)
7             smqval += nfpval
8             counter += 1
9     if counter == 0:
10        return 0.0
11    smqval /= counter
12    return smqval

```

Listing 3.3: Abstandsmaß für Einzelwerte

```

1 def dist(nfpsd, nfpmon):
2     return (nfpsd - nfpmon) / nfpsd.typerange

```

Trägerdokument die enthaltenen NFE sowie die für den Dienst vorhandenen Monitoringwerte als Eingabe der Berechnungsmethode.

Die zentrale Abstandsfunktion `dist` ist abhängig von der Expressivität definiert. Für Einzelwerte wird sie wie im Codeausschnitt 3.3 als Umsetzung der Berechnungsformel eingesetzt.

Für Wertebereichsangaben wird das ein modifiziertes Berechnungsschema angewandt. Dieses ist im Codeausschnitt 3.4 algorithmisch dargestellt.

SMQ-Wertaggregation auf Dokumentebene über Tableaus

Die Berechnung der SMQ für einzelne NFE als Vektorprodukt und Map-Reduce-Algorithmus dient als Grundlage für die Berechnung auf höheren Ebenen. Für

Listing 3.4: Abstandsmaß für Wertebereiche

```

1 def dist(nfpsd, nfpmon):
2     cfraction = 0.0
3     curanges = 0.0
4     dist = 0.0
5     foreach range in nfpsd.ranges:
6         curanges += range
7         fraction = getfraction(nfpmon, range)
8         cfraction += fraction
9         if cfraction < curange:
10            dist += curange - cfraction
11    return dist

```

die Betrachtung eines NFE-Trägerdokuments können bereits mehrdimensionale Abstandsmaße gebildet werden. Da nicht alle Werte in dieser Granularität benötigt werden, empfiehlt sich eine frühzeitige Projektion auf niederdimensionale Maße.

Zur Erläuterung soll ein Beispiel dienen. Gegeben sei eine Dienstbeschreibung als Träger für nfe_{spec} , ein dazugehöriger Monitoringreport sowie ein globaler Anforderungskatalog mit notwendigen NFE-Angaben jeweils als Träger für nfe_{ref} . Diese drei Dokumente bilden jeweils Dokumentvektoren \vec{doc} über die NFE, wie in Abbildung 3.13 vertikal dargestellt. Zwischen jeweils einem Spezifikationsvektor und einem Referenzvektor wird die SMQ über alle Qualitätsdimensionen berechnet.

NFE-Trägerdokumente			Vergleichsdokument
NFE	Dienstbeschreibung	Monitoring-report	
Preis	2,50 €		Preis
Urheber	"Entwickler"		Verfügbarkeit
Antwortzeit	<1,2 s	~1,4 s	Antwortzeit

Abbildung 3.13: Eingabe: NFE-Trägerdokumente und Anforderungsdokument

Horizontal müssen dafür alle NFE einander zugeordnet werden können. Neben fehlenden Einträgen sind auch unterschiedliche Bezeichner zu berücksichtigen. Es entsteht jeweils der Ergebnisvektor \vec{smq} mit den Abstandsmaßen über alle NFE in den um nicht vergleichbare NFE reduzierten Dokumentvektoren \vec{doc}_{smq} . Der Filteralgorithmus lautet:

$$\vec{doc}_{smq} = \text{filter}(\lambda x. \exists y = \vec{doc}_{ref_i} : x = y, \vec{doc}_{spec}) \quad (3.5)$$

Über alle in den erzeugten Vektoren \vec{doc}_{smq} vorkommenden NFE können nun für jede betrachtete Qualitätsdimension Abstandsmaße gebildet werden, wie in Abbildung 3.14 gezeigt. Zur Repräsentation kommt ein Tableau bzw. eine Matrix zum Einsatz, in deren Feldern die normierten Abstandsmaße und die nicht gebildeten Abstandsmaße vermerkt werden können. Die Gründe für nicht eingetragene Abstandsmaße sind abhängig von der Qualitätsdimension. So sind beispielsweise nicht alle NFE in ihrer Korrektheit überprüfbar oder in ihrer Vollständigkeit gefordert. Sowohl bezüglich der Qualitätsdimensionen (über alle NFE, spaltenweise) als auch bezüglich der NFE (über alle Qualitätsdimensionen, zeilenweise) lassen sich einfache und gewichtete Durchschnittswerte bilden. Jede Zeile entspricht dabei \vec{smq} , der Durchschnitt jeder Zeile SMQ .

Hierarchische Aggregation der Metaqualität

Die Metaqualität von Diensten kann auf fünf unterschiedlichen Ebenen $L0$ bis $L4$ angegeben werden, welche sich nach den betrachteten Artefakten und Entitäten richten. Orthogonal zu den phasenweise betrachteten NFE-Angaben werden SMQ-Angaben also ebenenweise betrachtet. Eine Zusammenführung zu einer Gesamtmetrik pro Dienst oder darüber hinaus bis hin zu einer qualitativen Bewertung eines Dienstmarktplatzes erfolgt hierarchisch, wie aus Abbildung 3.15

Qualitätsdimensionen

NFE	Vollständigkeit	Korrektheit	Präzision	Aktualität	Abstandsmaß über alle Dimensionen
Preis	1,0	n.g.	1,0	1,0	1,0
Urheber	n.g.	n.g.	1,0	1,0	1,0
Verfügbarkeit	0,0	k.A.	1,0	1,0	0,67
Antwortzeit	1,0	0,93	1,0	1,0	0,98
Ergebnisgüte	n.g.	n.g.	1,0	1,0	1,0
Abstandsmaß über alle NFE	0,67	0,93	1,0	1,0	

Abbildung 3.14: Ausgabe: SMQ-Tableau mit Spalten- und Zeilenzwischenwerten

ersichtlich wird. Als einschränkende Annahme wird dabei festgelegt, dass der Begriff *Dienst* sich auf eine bestimmte kontextfreie (zustandslose) Funktion mit wohldefinierter Ein- und Ausgabe bezieht. Sammlungen von Funktionen bilden somit keine eigene Ebene. Für häufig in Dienstbeschreibungen auftretende Bündel von Operationen, die teilweise untereinander implizite Aufrufreihenfolgen festlegen, ist somit eine Zerlegung in Einzeldienste mit gemeinsam genutzten Beschreibungsdokumenten notwendig.

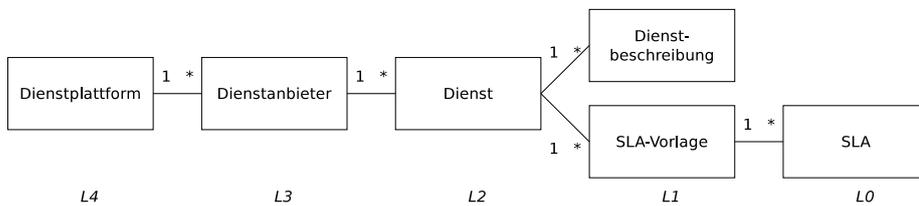


Abbildung 3.15: Hierarchische Aggregation der SMQ

Auf jeder Ebene L werden dazu gewichtete arithmetische Durchschnitte der n Metriken, mindestens jedoch einer, der darunterliegenden Ebene $L-1$ gebildet. Die Gewichte werden in einer Matrix w über alle Ebenen L und jeweils n_L zugehörigen Metriken ausgedrückt. Zur Vereinfachung wird eine Gleichgewichtung in w angenommen. Eine Gewichtung nach Faktoren wie der Höhe der SLA-Kompensation oder der Abrufpopularität von Dienstbeschreibungsdokumenten wird in dem vorgestellten Konzept nicht betrachtet, wäre aber als Erweiterung möglich und aufgrund der gewichtsbehafteten Formel problemlos einbindbar.

$$smq_{L0} = \langle \text{AbstandsmaessgemaessTableau} \rangle \tag{3.6}$$

$$smq_{L+1} = \frac{1}{\sum_{i=0}^{n_L} w_{Li}} * \sum_{i=0}^{n_L} smq_{Li} * w_{Li}, 0 \leq L < 4, n > 0; L, n \in \mathbb{N} \tag{3.7}$$

Für jede Ebene ergibt sich semantisch eine individuelle Interpretation der Metaqualität. Diese ist strukturell, nicht jedoch in ihrer Bedeutung, äquivalent

zur hierarchischen subjektiven Benutzerbewertung der Artefakte und Entitäten.

1. SLAs: Für jede SLA wird basierend auf Monitoringinformationen für jede enthaltene SLO die Metaqualität $smq_{contract}$ bestimmt. Dabei können neben den reinen Messwerten auch Informationen über die Auswirkungen auf SLA-Verletzungen einbezogen werden. Es ergibt sich ein Abstandsmaß, welches semantisch die Qualität im Sinne der Einhaltung der SLA beschreibt.
2. SLA-Vorlagen und Dienstbeschreibungen: Diese NFE-Trägerdokumente werden zumindest statisch überprüft. Die Metriken der von einer SLA-Vorlage abgeleiteten Verträge fließen in die Metriken der SLA-Vorlage mit ein. Es ergibt sich die Metaqualität $smq_{document}$.
3. Dienst: Die Metriken der zugehörigen SLA-Vorlagen und Dienstbeschreibungen ergeben zusammen die Metrik für den Dienst $smq_{service}$.
4. Dienstanbieter: Die Metaqualität aller Dienste eines Anbieters bestimmt die Metaqualität des Dienstportfolios des Anbieters $smq_{provider}$.
5. Dienstplattform: Die Metaqualität über alle Anbieter bestimmt die Metaqualität des Dienstangebots auf einer Plattform $smq_{platform}$.

3.3.4 Beispiel für die Berechnung von SMQ

Ein einfaches Beispiel für einen Dienst mit einer Dienstbeschreibung und einem SLA als NFE-Trägerdokumenten verdeutlicht die Berechnungsvorschrift für die Metaqualität. Die höheren Ebenen Dienstanbieter und Dienstplattform entfallen im Beispiel.

Die NFE Antwortzeit (rt) des Dienstes wird in aufeinanderfolgenden Aufrufen durch einen Monitor bestimmt. Die Messreihe lautet $rt_{mon} : i = [2, 2, 7, 8, 4, 2, 5]s$. Die NFE sei im Fall einfacher Wertangaben ungerichtet, also fix, mit $rt = 5s$ angegeben. Der Durchschnitt der Messungen beträgt hingegen $4,57s$. Das Abstandsmaß beträgt somit $0,43s$ im arithmetischen bzw. $0,91s$ im quadratischen Mittel.

$$\begin{aligned} rt_{sd} &= 5s \\ rt_{mon} : avg &= 4,57s \\ smq(rt_{sd}) &= 0,43s \end{aligned}$$

Im Fall von Wertbereichsangaben sei rt gerichtet in drei geschlossenen bzw. halboffenen Intervallen mit deren relativer Häufigkeit angegeben:

$$rt_{sd} \leq \begin{cases} [0; 4]s & 10\% \\ (4; 7]s & 75\% \\ (7; \infty]s & 15\% \end{cases}$$

Die absolute Häufigkeit der Messungen in den Intervallen beträgt hingegen

$$rt_{mon} \leq \begin{cases} [0; 4]s & 4 \\ (4; 7]s & 2 \\ (7; \infty]s & 7 \end{cases}$$

relativ somit auf ganze Zahlen gerundet

$$rt_{mon} \leq \begin{cases} [0; 4]s & 57\% \\ (4; 7]s & 29\% \\ (7; \infty]s & 14\% \end{cases}$$

Die kumulative Abweichung der Auftrittshäufigkeit beträgt

$$smq(rt_{sd}) \leq \begin{cases} [0; 4]s & 47\% \\ (4; 7]s & 11\% \\ (7; \infty]s & 0\% \end{cases}$$

im Mittel demnach 19,3%.

Für einen SLA sei die Angabe stets gerichtet mit $rt \leq 5s$ bzw.

$$rt_{sla} \leq \begin{cases} [0; 4]s & 10\% \\ (4; 7]s & 75\% \\ (7; \infty]s & 15\% \end{cases}$$

garantiert. Somit stellen nur die beiden Messwerte zu 7s und 8s eine SLO-Verletzung dar, ohne dass sich daraus notwendigerweise bereits eine SLA-Verletzung ergäbe. Es ergibt sich eine durchschnittliche Abweichung vom Einzelwert um 0,71s im arithmetischen und 0,52s im quadratischen Mittel sowie eine kumulative Abweichung von 0%.

3.3.5 Angabe der Metaqualität in Beschreibungsdokumenten

Der Zusammenhang zwischen Metaqualitätsinformationen und NFE-Trägerdokumenten erschließt sich sowohl bezüglich der Bildung als auch der Rückführung der SMQ-Metriken.

Die Ausdrucksstärke von SMQ-Coverage hängt von der Ausdrucksstärke der spezifizierten NFE ab. Die Abbildung 3.16 zeigt, inwieweit übliche Dienstbeschreibungssprachen und SLA-Formate in der Lage sind, eine SMQ-Coverage mit hoher Expressivität zu erreichen. Für un spezifizierte Formate wie WS-Agreement muss dabei mangels Mindestanforderungen an Implementierungen von einer Nichtangabe von Werten ausgegangen werden. Diese Einschränkung ist nur durch spezialisierte Formate wie WS-Agreement+ValueTypes oder WS-Agreement+USDL aufhebbar.

Jede SMQ-Metrik wird auf der untersten Ebene mit ihrer zugehörigen NFE-Angabe im korrespondierenden Trägerdokument in einer 1:1-Relation ausgedrückt. Aus diesem Grund folgt auch die annotierte Spezifikation von SMQ in Dienstbeschreibungsdokumenten dem Format der NFE-Spezifikation unter Ausnutzung der Annotierbarkeit der NFE-Angaben. Es entstehen somit neue eigenschaftsspezifische *Metaproperties*, für deren Auswertung spezielle Werkzeugunterstützung benötigt wird. Die aggregierten Metaqualitätswerte können hingegen unter geringem Informationsverlust als normale *Properties* auf der Ebene der betrachteten NFE angegeben und ebenso wie vorhandene NFE verarbeitet

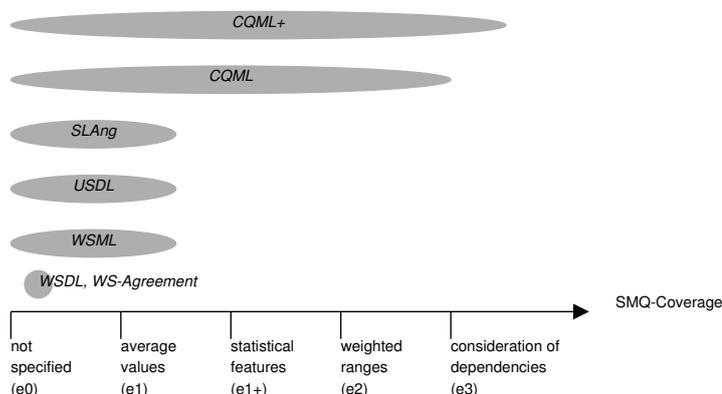


Abbildung 3.16: Sprachunterstützung für die Angabe von SMQ-Coverage

werden, um etwa darauf ein Ausschlusskriterium im Filter einer Dienstausswahl anzuwenden. Insbesondere fördert diese Art der Angabe von SMQ-Werten die weitere Analyse durch zusätzliche Metaproperties auf die SMQ-Angaben.

Die Metaqualität ist unabhängig von einem konkreten Beschreibungsformat zu betrachten. Um die mögliche Anwendbarkeit in dienstorientierten Umgebungen zu demonstrieren, stellt dieser Abschnitt die Angabe in zwei konkreten Sprachen CQML+ und WSML vor. Aufgrund der teilweise eingeschränkten Ausdruckskraft der Sprachen ist die vollständige Übernahme der symbolischen Angaben von $smq(nfe_{sd})$ nur bedingt möglich. Desweiteren werden menschenlesbare Metaqualitätsberichte als neuer Dokumenttyp und dynamischer Dienstarfakt *SMQ Report* analog zu Monitoring-Reports eingeführt, um die langfristige Verbesserung von Qualitätseigenschaften im Rahmen von Entwicklungsprozessen zu fördern.

Angabe in CQML+

CQML+ als Sprache zur Spezifikation der Qualität von Softwarekomponenten beinhaltet ein optimiertes maschinenlesbares XML-Format sowie die Angabe von Ressourcenabhängigkeiten. Die Sprache beinhaltet allerdings per se keine Notationsmöglichkeit für Annotationen auf Qualitätscharakteristiken, die ihrerseits zur Spezifikation von NFE genutzt werden. Eine Evaluierung des Abdeckungsgrades und eine Feststellung der Gültigkeit von CQML-Ausdrücken basierend auf einem Evaluationsframework für die Qualität von Modellierungssprachen wurde vom Autor zwar vorgeschlagen [56]. Das semiotische Evaluierungsframework ist unter dem Namen SEQUAL bekannt geworden, wurde jedoch noch nicht auf CQML angewendet.

Als Folge empfiehlt sich die Angabe von SMQ-Metriken in CQML+ als assoziierte Qualitätscharakteristiken, die durch einen abgeleiteten Namen von ihrer eigentlichen NFE gebildet werden. Die Assoziation ist dabei nicht durch die Semantik von CQML+ gedeckt und erfordert eine spezielle Auswertung. Beispielsweise kann die domänenspezifische Ebenheits-Eigenschaft der Ergebnisse eines Bildverarbeitungsdienstes wie im Quelltext 3.5 angegeben sein:

Die korrespondierende Metaqualitätsmetrik kann nun wie in der Spezifikation 3.6 annotiert werden.

Listing 3.5: Minimalspezifikation eines Dienstes in CQML+

```

1 quality smooth {
2   smoothness = good;
3 }

```

Listing 3.6: Metaqualitätsangaben in CQML+

```

1 quality smooth {
2   smoothness = good;
3   smoothness_confidence = 95%;
4 }

```

Angabe in WSML

WSML als Sprache zur semantischen Repräsentation von Dienstkonzepten ist syntaktisch flexibel gestaltet. WSML-Ausdrücke enthalten im Gegensatz zu CQML+ bereits die Möglichkeit der Metadatenangabe zu jeder NFE. Eine Spezifikation inklusive der Metaqualitätsinformationen kann über derartige Annotationen analog zum Codeausschnitt 3.7 werden.

Einschränkend gilt, dass diese Annotationen zwar spezifiziert und verarbeitet werden können, ihnen allerdings keine Metaqualitäts-Semantik gemäß der Sprachspezifikation zugrunde liegt. Somit müssen verarbeitende Anwendungen die Annotationen in ihrer Anwendungslogik interpretieren.

Angabe in Metaqualitätsberichten

Die Angabe der SMQ wird von Dienstnutzern nur dann sinnvoll interpretiert werden können, wenn sie auf einem genügend hohen Niveau aggregiert wird, etwa auf der Methoden- oder Dienstebene. Die Entwickler eines Dienstes haben hingegen ein Interesse an einer möglichst feingliedrigen Analyse der NFE-Angaben und somit an einer möglichst umfassenden Darstellung aller SMQ-Informationen. Um diese effizient von der Laufzeitumgebung in die Entwicklungsumgebung transferieren zu können, empfiehlt sich ein Zugriff via Web Service und eine Formatierung der SMQ-Informationen als XML-strukturierter Me-

Listing 3.7: Metaqualitätsangaben in WSML

```

1 namespace {smq _"urn:service:metaquality"}
2 webService
3   nfp
4     smoothness hasValue "good"
5     annotations
6       smq#confidence hasValue 95%
7     endAnnotations

```

Listing 3.8: *Metaqualitätsangaben in Berichten*

```
1  smqreport {
2    service "imageprocessing"
3    smq 0.81
4    method "info"
5      smq 0.98
6    method "process"
7      smq 0.64
8      nfp "responsetime"
9        smq 0.87
10     nfp "availability"
11       smq 0.12
12     nfp "security"
13       smq 0.92
14 }
```

taqualitätsbericht. Es folgt ein Beispiel eines solchen Berichts zur Qualität einer Dienstbeschreibung in Kompaktnotation im Listing 3.8. Der Bericht gibt Aufschluss darüber, dass der Dienst vermutlich korrekt arbeitet, die Dienstinstanz jedoch unter ungünstigen Hostingbedingungen angeboten wird. Die Berechnung erfolgt unter der Annahme gleicher Gewichtung aller Metriken.

3.3.6 Anwendungsfelder

Für die Metaqualität von Diensten lassen sich eine Vielzahl an Anwendungsfeldern anhand bisheriger Defizite in Dienstplattformen bestimmen.

- In großen Dienstangeboten werden statische Metadaten wie etwa Dublin-Core-Angaben zum Autor und zur Versionierung wahrscheinlich nicht von Hand gepflegt werden. Stattdessen werden diese NFE im Rahmen des Entwicklungsprozesses aus anderen Datenquellen hinzugefügt. Datenbankinhalte weisen jedoch aufgrund von Konsistenzproblemen fast immer eine Fehlerrate ungleich Null auf. Auch unzureichend spezifizierte Schemaeinschränkungen, Ein- und Ausgabefehler sowie generelle Fehler im Ablauf der Datenpflegeprogramme tragen dazu bei [80]. Das Wissen über solche Fehlerraten sollte in die Dienstbeschreibungen propagiert werden.
- Dienste, die nach dem Paradigma der modellgetriebenen Softwareentwicklung erstellt werden, können einige QoS-Eigenschaften direkt aus dem Modell ableiten. Dabei kann das Wissen über die zu erwartende Laufzeitumgebung mit einberechnet werden. Derartige Vorausberechnungen von Verhaltensweisen sind jedoch zumeist unvollständig. Es ist nahezu unmöglich, sämtliche Einflussfaktoren mit in die Berechnung einzubeziehen. In der Konsequenz kann es zu Diskrepanzen zwischen den vom Modell vorgegebenen und den gemessenen Werten kommen. In solchen Fällen ist es dann oftmals nicht klar, ob das Modell oder der Monitor die Fehlerursache beinhalten. Es kann zumindest aus der Verletzung von Dienstgütevereinbarungen die Aussage gewonnen werden, dass die Dienstbeschreibungen nicht hinreichend spezifiziert worden sind.

- In Dienst-Communities bewerten die Nutzer die von ihnen ausgehandelten Verträge und die von ihnen genutzten Dienste. Abhängig von der Aggregation der Bewertungen und Robustheit des Bewertungssystems gegenüber Manipulationsversuchen können diese Bewertungen ebenfalls wieder als Qualitätsmerkmale aufgefasst werden.

Zwei hauptsächliche Anwendungsfälle haben sich dabei für die Nutzung der Metaqualitätsangaben herausgestellt:

- **Unschärfe:** In einigen Szenarien wie der Dienstausswahl ist es wünschenswert, eine gewisse Unschärfe der NFE berücksichtigen zu können. Wenn beispielsweise ein Dienst eine Antwortzeit von 20 Millisekunden oder weniger aufweisen soll und ein Dienst eine durchschnittliche Antwortzeit von 20,12 Millisekunden angibt, kann das Wissen über die Auftrittswahrscheinlichkeiten der einzelnen Messergebnisse oder andere Details zur Modellierung verhindern, dass dieser Dienst aus der Ergebnisliste der Dienstausswahl ausgeschlossen wird.
- **Herabsetzung:** In anderen Szenarien hilft die Kenntnis über die Qualität von NFE, schlechte Dienste auszuschließen, die durch ihre inakkurat beschriebenen Verhaltensweisen Probleme bei der Ausführung verursachen.

3.3.7 Abgrenzung zu anderen Metriken

Zur besseren Einordnung und Abgrenzung zu ähnlichen dienstbewertenden Metriken wird die Metaqualität für Dienste anderen weit verbreiteten NFE gegenübergestellt.

Dienstqualität: Die Qualität eines Dienstes unterliegt gewollten und ungewollten Schwankungen. Die erste Gruppe umfasst Differenzierungsmerkmale wie Preismodelle, individuelle Vertragsterme, Priorisierungen und Lastbegrenzungsverfahren. Die zweite Gruppe umfasst Störungen auf Software-, System- und Verbindungsebene. Die Metaqualität liefert hingegen ein Maß dafür, wie umfassend, schnell und genau diese Schwankungen erfasst und in die Prozesse zur Dienstverarbeitung einbezogen werden können. In dieser Hinsicht existiert eine definitorische Unterscheidung zwischen den Metriken Dienstqualität und Metaqualität.

Reputation: Die NFE Reputation bestimmt das Ansehen eines Dienstes bei einem einzelnen Dienstkonsumenten oder einer Gruppe von Konsumenten. Als solches ist es eine subjektive, nichtlineare und schwer vorhersagbare NFE. Ein Dienst, der über eine lange Zeit alle Anforderungen erfüllt und anschließend kurzzeitig von einem schweren Zwischenfall betroffen ist, wird höchstwahrscheinlich für die darauffolgende Zeit eine schlechte Reputation aufweisen, die von der dann gelieferten Dienstqualität entkoppelt sein wird. Ein Beispiel dafür ist eine ohne vorherige Ankündigung durchgeführte und anschließend wieder zurückgenommene Preiserhöhung. Jedoch wäre die Metaqualität in diesem Fall sowohl während der Erhöhungsphase als auch nachher nicht betroffen, wenn die Dienstbeschreibung diese Änderungen akkurat widerspiegelt. Aus diesem Grund unterscheidet sich das objektive Maß Metaqualität von dem subjektiven Maß Reputation.

3.4 Verfahren zur inkrementellen Qualitätsverbesserung

Vertragliche Garantien auf Dienstgütekriterien sind abhängig von einer möglichst umfassenden und hochqualitativen formalen Beschreibung der NFE von Diensten. Viele Ansätze treffen keine Aussage über die Herkunft der NFE oder positionieren deren Erzeugung in den Dienstentwicklungsprozess. Die Praxis zeigt jedoch, dass die durch die Entwicklungsprozesse durchaus gegebenen Möglichkeiten der Modelltransformation und Annotation nur wenig genutzt werden. In der gängigen Praxis werden demnach die NFE oft willkürlich spezifiziert. Ein evolutionärer Prozess zu deren Pflege existiert nicht. Diese Problematik der Pflege von Metadaten spiegelt sich auch in anderen Bereichen wie dem semantischen Tagging von multimedialen Inhalten und persönlich genutzten Dokumenten wider, wie in den Grundlagen zur NFE-Modellierung erläutert wurde.

Ein bekanntes Verfahren zur Lösung des Problems ist die automatische oder manuelle nachträgliche Ergänzung fehlender oder die Korrektur ungültiger Metadaten und Annotationen. Beispiele hierfür finden sich beispielsweise in Methoden zur Anreicherung syntaktischer Dienstbeschreibung um Annotationen für grafische Schnittstellen für die Interaktion mit Diensten im WSGUI-Ansatz oder im dazu verwandten ServFace-Ansatz [32] sowie im Umfeld des Semantic Web [115]. Eine Ausweitung auf die Angabe von NFE wurde hingegen bisher nicht erprobt.

Das Konzept der koordinierten inkrementellen Qualitätsverbesserung soll somit die Zusammenführung der aus dem Entwicklungsprozess stammenden NFE-Spezifikationen mit den aus Monitoringinformationen und anderen Laufzeitquellen gewonnenen NFE sowie statischen Anforderungskatalogen bewirken. Hierbei wird ein integrierter zusammengesetzter Prozess geschaffen, der mit minimalen Aufwänden seitens der Nutzer im Internet der Dienste eine Qualitätsverbesserung sowohl für einzelne Dienste als auch für Kompositionen und Dienstportale bewirkt, deren Qualität indirekt von den Spezifikationen der genutzten und angebotenen Dienste abhängt.

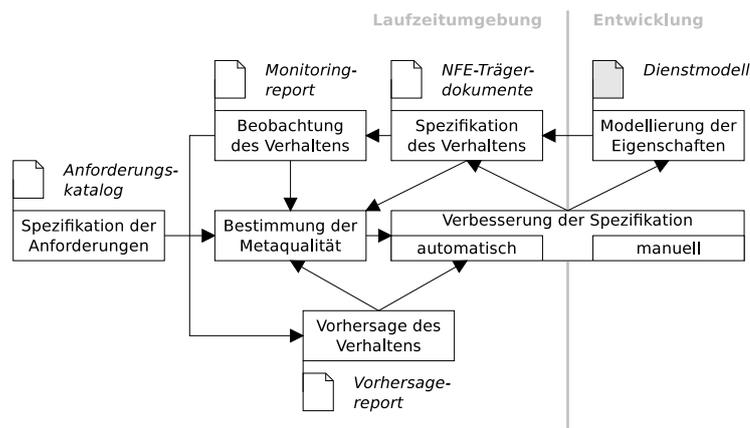


Abbildung 3.17: Konkreter Prozess der Metaqualitätsgewinnung, -rückführung und -verbesserung

Die Übersichtsgrafik 3.17 gibt einen Einblick in die gegenüber der Grafik 3.2 technisch und architektonisch konkretisierten Verfahrensabläufe zur Gewinnung und Verwendung der Metaqualität aller auf einer Dienstplattform angebotenen Dienste. NFE-Angaben in Dienstbeschreibungen und SLA-Vorlagen werden kontinuierlich auf Qualitätskriterien wie Vollständigkeit geprüft. Sie werden auch gegenüber weiteren Datenquellen wie Messergebnissen eines Monitors dynamisch auf Korrektheit geprüft. Zusätzlich können Vorhersagemechanismen sowohl als Messdatenquelle als auch zur Verbesserung der NFE-Wertangaben in Beschreibungen herangezogen werden. Ist dadurch die Metaqualität bestimmt, wird sie zunächst als Annotation auf die betreffenden NFE-Angaben zurückpropagiert. Die NFE-Angaben werden darüber hinaus automatisch auf der Dienstplattform oder manuell durch den Entwickler so lange angepasst, bis sie vollständig und akkurat das Verhalten des Dienstes beschreiben. Durch diesen iterativen Prozess konvergiert die Metaqualität schließlich gegen ihren Maximalwert.

Die Verfahren richten sich nach den in Grafik 3.17 enthaltenen Abläufen zur Gewinnung, Propagierung und Verbesserung der Metaqualität von Diensten. Sie werden an dieser Stelle detailliert als Einzelverfahren unabhängig von einer zugrunde liegenden Dienstplattform angegeben. Die Position im Gesamtprozess wird jeweils zum besseren Verständnis zu Beginn der Erläuterungen jedes Einzelverfahrens noch einmal in einer vereinfachten Prozessdarstellung hervorgehoben.

3.4.1 Gewinnung von Aussagen über NFE aus Monitoringdaten

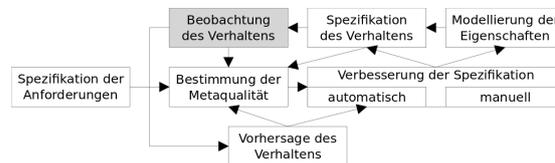


Abbildung 3.18: Einordnung in den Gesamtprozess

Monitoring auf Dienstplattformen ist ein parallel zur Dienstaufführung ablaufender Hintergrundprozess, der zum System, zu jedem Dienst und zu jedem Dienstaufwurf eine Menge an Messdaten zu funktionalen und nichtfunktionalen Eigenschaften liefert. Neben direkten Messungen durch Sensoren können weitere Daten wie Durchschnittswerte über die Zeit durch Aggregation dieser Messdaten erzeugt werden. Um die Metaqualität zu NFE bilden zu können, ist die Bestimmung der tatsächlich erzielten Dienstgüte durch Monitoringkonzepte als Voraussetzung notwendig.

Allerdings lässt sich nicht jede NFE stets durch automatisches Monitoring überprüfen (vgl. Abbildung 3.11). Die Eigenschaften werden daher orthogonal zu einer eventuell vorhandenen makroskopischen NFE-Klassifikation (vgl. Abschnitt 2.3.1) in drei Kategorien bezüglich der Überprüfbarkeit eingeteilt:

1. Monitoring ohne Plattformunterstützung. In diese Kategorie fallen Eigenschaften, die von außerhalb des Systems bestimmbar sind, wie beispielsweise die Verfügbarkeit und (bei Einsatz eines Proxy) die aufrufbezogenen

Größen Antwortzeit, Datendurchsatz und Erfolgsgrad. Diese Eigenschaften stellen eine Teilmenge von **QoS-NFE** dar.

2. Monitoring mit Plattformunterstützung. Eigenschaften, die nur durch direkten Zugriff auf die Dienstbringung bestimmt werden können, sind in dieser Kategorie zu finden. Beispiele hierfür sind der Ressourcenverbrauch und die Aufrufhäufigkeit von Diensten. Diese Eigenschaften bilden die komplementäre Teilmenge von **QoS-NFE**.
3. Kein Monitoring. Dies betrifft alle Eigenschaften der subjektiven Wahrnehmung, nichttechnische Eigenschaften wie Preise sowie aus Komplexitätsgründen nicht überprüfbare Angaben wie etwa den garantierten Abschluss der Datenweitergabe an Unbeteiligte in der Dienstbringungskette. Hierin werden also **Metadata-NFE** und **CPR-NFE** zusammengefasst.

Die Konfiguration eines Monitors erfolgt über Dienstbeschreibungen, SLA-Vorlagen und SLAs. Die darin angegebenen NFE werden auf ihre strukturellen Merkmale wie Typisierung, Einheiten und Grenzwerte geprüft. Ist eine NFE als automatisch kontrollierbar gekennzeichnet, ohne dass der Monitor diese Überprüfung jedoch durchführen kann, so muss dies in der Ausgabe des Monitors vermerkt werden, um diese Diskrepanz nicht zu Lasten des Dienstes auszulegen.

3.4.2 Vorhersage von NFE basierend auf Monitoringdaten

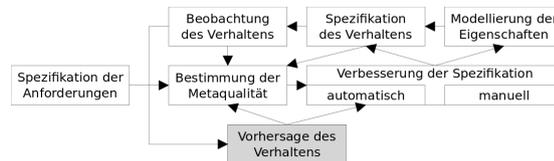


Abbildung 3.19: Einordnung in den Gesamtprozess

Die Heterogenität von Diensten bezieht sich sowohl auf deren Implementierung als auch auf deren Beschreibung. Je präziser ein Dienst beschrieben ist, desto bessere Aussagen lassen sich daraus über sein Verhalten und seine Eigenschaften ableiten. Die Angabe von NFE in Dienstbeschreibungen und SLA-Vorlagen reicht dabei von willkürlichen Festlegungen und Abschätzungen seitens des Entwicklers bis hin zu modelltheoretisch abgeleiteten Größen und Formeln. Der Zusammenhang zwischen diesen qualitativ unterschiedlichen Angaben und den tatsächlichen Laufzeiteigenschaften ist in Abbildung 3.20 dargestellt.

Aus den genannten Gründen ist die bloße Betrachtung der gemessenen und aggregierten Monitoringdaten nicht ausreichend für eine Aussage über das tatsächliche Qualitätsprofil eines Dienstes. Je nach Vorhandensein von zusätzlichen Dienstbeschreibungsmerkmalen lassen sich die Eigenschaften über ein vorhersagendes Tendenzmonitoring oder über eine analytische Vorhersage basierend auf Monitoring- und Modelldaten genauer bestimmen.

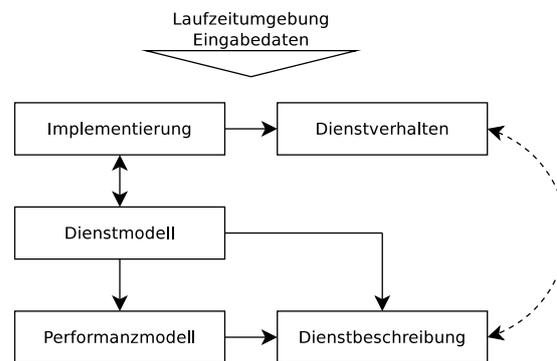


Abbildung 3.20: Problematik der inkonsistenten Dienstbeschreibungen

Tendenzmonitoring

Eine Problematik von nicht a priori bekannten NFE lässt sich durch die Frage ausdrücken, ob NFE überhaupt zur Laufzeit durch Beobachtung des Dienstes als „black box“ oder „grey box“ bestimmbar sind. Die Gefahr besteht darin, durch Beeinflussungen der Umgebung, der Eingabedaten oder anderer Dienste verfälschte Messwerte zu bekommen, die sich nicht direkt auf die gewünschte NFE abbilden lassen. Beispielsweise unterliegt die Ausführungszeit eines Dienstaufrufs Schwankungen, wenn im Betriebssystem höher priorisierte Prozesse aktiv werden.

Das *Tendenzmonitoring*, auch *Trendmonitoring* genannt, betrachtet Messwerte über einen Zeitraum unter Berücksichtigung äußerer Faktoren und hilft hierbei, solche Fehler zu reduzieren. Prinzipiell geht es dabei um die Frage, inwieweit die Menge aller beobachteten Dienstverhaltensweisen sich durch Subtraktion der erkennbaren Verhaltensweisen auf eine möglichst kleine Restmenge an nicht identifizierbaren oder chaotischen Verhaltensweisen reduzieren lässt. Im Umkehrschluss benötigt man eine möglichst hohe Zahl an erkennbaren Verhaltensweisen. Bezogen auf die Monitoringdaten und deren Darstellung als Funktion über die Zeit oder die Dienstaufrufe lässt sich das Problem also zurückführen auf die Ableitung von Formeln aus Messkurven. Die Ableitung ist trivial für invariante NFE, also konstante Werte, sowie linear von einer Größe abhängigen NFE. Sind komplexere Abhängigkeiten festzustellen oder liegen selbst im Falle der Linearität mehrere Ursprungsgrößen vor, so ist eine Ableitung selbst unter Einbeziehung mathematischer Approximations- und Interpolationsmethoden und anderer numerischer Verfahren nicht ohne weiteres möglich, da die Ergebnisfunktion erst in ihre Einzelbestandteile dekomponiert werden muss [1, 7].

Das Tendenzmonitoring überführt jede Messgröße nfe_{mon} in ein Derivat $nfe_{montrend}$.

Analytische Vorhersage durch Schätzer

Eine primäre Quelle für die Vorhersage von Diensteigenschaften ist die Implementierung des Dienstes. Allerdings ist die Analyse auf Basis von Quellcode oder Binärcode, beispielsweise durch statische interprozedurale Programmanalyse oder Tracing-Mechanismen, nicht trivial und in heutigen Ablaufumgebungen für Dienste nicht praktikabel.

Günstiger ist bei gegebener Möglichkeit die Analyse von Modellen zur Implementierung, welche bereits verifiziert worden sind. Insbesondere für prozessorientierte Dienste und für Softwarekomponenten existieren Verfahren zur Ableitung von Performanzmodellen als Spezialisierung genereller dynamischer Laufzeitmodelle zur Vorhersage des Verhaltens von Anwendungen [18]. Diese können im Kontext der Dienstaufführung als Schätzer genutzt werden, um ausgewählte NFE wie Antwortzeit oder Durchsatz vorherzusagen.

Im Fall spezifizierter Ressourcenanforderungen einer dienst anbietenden Softwarekomponente können NFE dynamisch von den zur Verfügung stehenden Ressourcen wie Prozessorleistung, freier Arbeitsspeicher und Netzwerkkapazität berechnet werden [66]. Erlaubt das System eine Ressourcenreservierung, lassen sich die so abgeschätzten Werte sogar fixieren, so dass der iterative Ableichsprozess entfällt.

Für die Vorhersage der Antwortzeit komponierter Dienste wird eine Performanz-Analyse eingebunden, wofür unterschiedliche Ansätze wie *PerMoTo* und *WPX.KOM* existieren [44, 26]. Dabei wird der dienstinterne Ausführungs- und Datenfluss auf ein Warteschlangensystem abgebildet. Für den Durchsatz kann eine aktive oder passive Bandbreitenanalyse durchgeführt werden. Für diesen Zweck existieren bewährte Verfahren wie Pathload, Yaz und Spruce, die je nach verfügbarem Wissen über die Leitungseigenschaften unterschiedlich brauchbare Ergebnisse liefern [39].

Für alle anderen NFE ohne spezialisierten Schätzer ist nur ein generischer Schätzer nutzbar, der sich auf das vorhersagende Monitoring und eventuell vorhandene Benutzerbewertungen stützt [120]. Dieser liefert je nach Komplexität direkt den abgeschätzten Wert zurück oder verwendet aus Gründen erhöhter Stabilität bei gleichzeitig geringerer Anpassungsgeschwindigkeit geglättete Durchschnitte [105].

Allerdings findet bei all diesen Verfahren eine teilweise Verschiebung des Problems statt, da nunmehr zwar bestimmte NFE berechnet werden können, allerdings in Abhängigkeit von per se nicht bekannten Eingangsvariablen wie zu erwartender Häufigkeit von Anfragen an den Dienst. Zumindest sind grundsätzlich untere und obere Schranken für die NFE ableitbar.

Dem Gewinn an Metaqualität durch den Einsatz von Schätzern stehen zwei Nachteile entgegen. Einerseits können im Vorfeld Kosten entstehen, wenn die Schätzer mit parametrisierbaren Modellen konfiguriert werden müssen. Dies ist beispielsweise bei der Performance Prediction der Fall. Andererseits können zur Ablaufzeit Kosten entstehen, da durch das Monitoring und den Analysevorgang Ressourcen wie Verarbeitungszeit und Speicher in Anspruch genommen werden. Nicht zu vernachlässigen ist neben diesen internen Qualitätseigenschaften des Monitors und der Schätzer auch deren externer Einfluss auf die Qualität der Vorhersagedaten und die Gesamtsituation des Systems. So verlassen sich etliche Mechanismen zur Vorhersage der Bandbreite auf aktive Prüfspitzen (*probes*), deren exzessiver Einsatz eine kritische Netzwerksituation noch verschlechtern und Vorhersageergebnisse verfälschen kann. Es ist demnach ratsam, die Qualitätseigenschaften aller am Verfahren zur Verbesserung der Metaqualität beteiligten Komponenten in die Gesamtbetrachtung mit einzubeziehen. Dies betrifft insbesondere die Exaktheit, die Laufzeit und den Störeinfluss der Komponenten [39].

Die Schätzer der analytischen Vorhersage überführen jede Messgröße nfe_{mon} in ein Derivat $nfe_{monpred}$. Der Ablauf der analytischen Vorhersage mitsamt der

3.4. VERFAHREN ZUR INKREMENTELLEN QUALITÄTSVERBESSERUNG 77

möglichen Fehlerstellen wird in Abbildung 3.21 gezeigt.

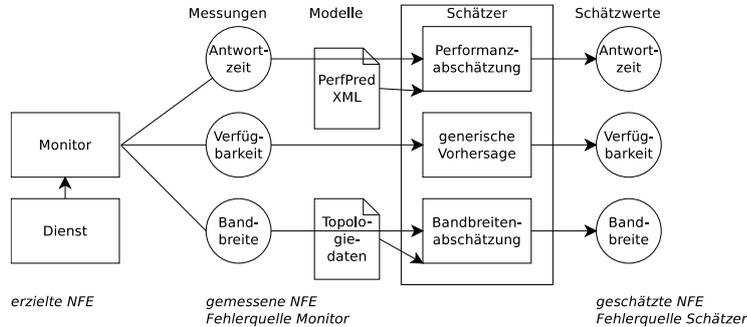


Abbildung 3.21: Analyse durch spezialisierte Schätzer mit möglichen Fehlerstellen

3.4.3 Bestimmung der Metaqualität

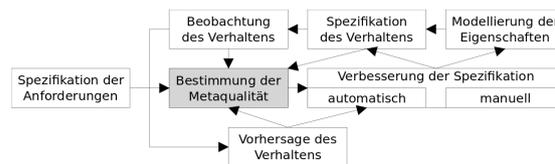


Abbildung 3.22: Einordnung in den Gesamtprozess

Im Unterabschnitt 3.3.3 wurde die SMQ-Berechnungsvorschrift entworfen. Diese ist ein zentraler Bestandteil des Prozesses zur Bestimmung der SMQ.

Zur Bestimmung der SMQ-Coverage werden sowohl die Dokumente mit Angaben nichtfunktionaler Eigenschaften (NFE-Trägerdokumente) als auch ein Anforderungskatalog als Vergleichsdokument in der Laufzeitumgebung benötigt. Diese Eingaben werden über die Berechnungsformeln in eine SMQ-Coverage-Metrik gewandelt, welche entweder direkt oder als Teilmetrik von SMQ in den Messdatenstrom zurückgegeben wird und somit für die Visualisierung oder NFE-basierte Dienstausswahl ohne weitere architektonische Ergänzungen direkt zur Verfügung steht.

Zur Bestimmung der SMQ-Validity werden zumindest Monitoringdaten, optional jedoch auch Vorhersagedaten herangezogen. Die weitere Nutzung erfolgt analog zur Teilmetrik SMQ-Coverage.

3.4.4 Rückführung und Nutzung der Metaqualität

Der Kreislauf der Verarbeitung von SMQ schließt sich mit der kurzfristigen Rückführung der Metrik in die Dienstbeschreibungsdokumente und der langfristigen Maximierung der SMQ durch eine dynamische Anpassung der NFE-Angaben an die tatsächlich erbringbaren Leistungswerte der Dienste gemäß Abbildung 3.23. Die dynamische Anpassung kann automatisch zur Laufzeit oder semiautomatisch nach Bestätigung des Dienstentwicklers erfolgen. Weiterhin hat der Entwickler die Möglichkeit, Informationen zu Modifikationen in die Entwicklungsumgebung zu laden, um zukünftige Versionen eines Dienstes von vornherein

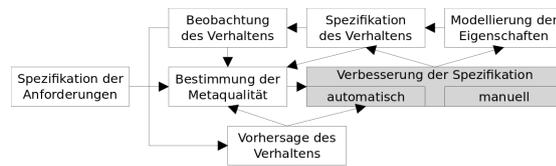


Abbildung 3.23: Einordnung in den Gesamtprozess

mit hochqualitativen Beschreibungen ausliefern zu können. Damit wird über die Erkennung von Qualitätsdefiziten hinaus eine Propagierung und ein Verbesserungskonzept in den Lebenszyklus integriert.

Nutzen der Rückführung und Maximierung der SMQ

Die Rückführung der SMQ-Metriken hilft vor allem dem Dienstanbieter, zwischen hoch- und niederqualitativ beschriebenen Diensten zu unterscheiden. Allerdings wird dabei unter der Annahme, dass eine zu niedrige Metaqualität in der Dienstsuche als Ausschlusskriterium genutzt wird, das verfügbare Dienstangebot reduziert.

Sofern Vorhersage- oder andere Verbesserungstechniken angewandt werden, fließen auch die aktualisierten NFE entweder automatisch in die dynamisch aktualisierten Beschreibungsdokumente ein, oder stehen zumindest für den Abruf in Entwicklungsumgebungen zur manuellen Übernahme bereit. In diesem Fall profitiert in erster Linie der Dienstanbieter, da mit den auf diese Weise realistischer gestalteten Dokumenten eine Nutzung der Dienste attraktiver wird. Indirekt profitiert ebenfalls der Dienstanutzer, da sich im Vergleich mit einer reinen Propagierung der SMQ das Dienstangebot nicht reduziert, sondern nur anders einordnet.

Beide Verfahren benötigen ein Konzept für dynamisierte deklarative Spezifikationen von NFE in Beschreibungsdokumenten, und somit auch eine darauf angepasste Laufzeitumgebung. Dies stellt eine Erweiterung gegenüber der üblichen statischen Verwaltung von Dokumenten in Registries und SLA-Verwaltungen dar. Der Adaptionsmechanismus zur Anpassung von Dokumenten soll dabei als *Adjustment* bezeichnet werden. Er agiert komplementär zur Adaption auf Dienst- oder Systemebene, also zu Verfahren wie Rekonfiguration von Diensten, Neubindung von atomaren Diensten in komplexen Prozessen und Ressourcenallokation. Im speziellen Fall der Aktualisierung nichtfunktionaler Angaben in einem Vertragsdokument ist jedoch bereits der Begriff Neuaushandlung etabliert, der dafür als Spezialisierung des *Adjustment* auch verwendet werden soll.

In Anlehnung an das vorausgesetzte NFE-Strukturmodell (siehe Abbildung 2.7) kann nun eine Modellverfeinerung vorgenommen werden, die eine Aufteilung der Metaeigenschaften in statische und durch die Metaqualitäts-Verfahren dynamisierte Eigenschaften vollzieht. Die Aufteilung wird durch Abbildung 3.24 veranschaulicht. Während der Bezeichner und die Einheit von NFE als statisch angenommen werden, sind Änderungen der Werte, Typen oder Wertebereiche und beliebiger Annotationen in der Methodik vorgesehen.

Abbildung 3.25 stellt sowohl die Rückführung als auch die Maximierung der SMQ noch einmal exemplarisch dar. Im Schritt (a) ist initial eine NFE **success** definiert, die die Wahrscheinlichkeit angibt, dass ein Dienstauftrag erfolgreich

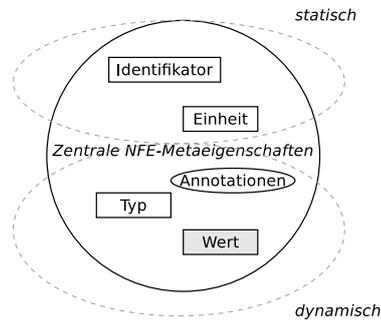


Abbildung 3.24: Statische und dynamische Metaeigenschaften

verläuft. Im Schritt (b) liefert das Monitoring nach einigen Aufrufen das Wissen darüber, dass die angegebene Wahrscheinlichkeit zu hoch angesetzt wurde. Gleichzeitig wird unter Ausnutzung von weiteren Informationsquellen wie der Auslastung des Systems eine Aufrufwahrscheinlichkeit geschätzt, die zwar etwas besser als die bisher gemessenen Werte liegt, aber immer noch den angegebenen Wert unterbietet. Der Schritt (c) enthält nun die Berechnung der Metaqualität bezogen auf die Metrik **success** und deren Propagierung in die Dienstbeschreibung. Nach weiteren Aufrufen wird die NFE-Angabe dynamisiert und an die auf Monitorwerten basierende Vorhersage angepasst. Dies führt in Schritt (d) zu einer höheren Metaqualität.

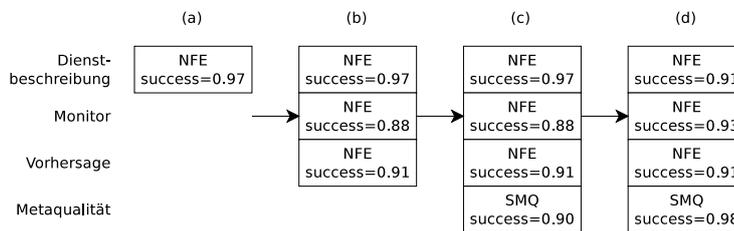


Abbildung 3.25: Rückführung und Maximierung der SMQ

Das technische Konzept zur Rückführung der SMQ in Dienstbeschreibungen, Vorlagen für Dienstgütevereinbarungen (SLAs) und aktive SLAs wird an dieser Stelle im Detail vorgestellt. Darauf folgt für die gleichen Dokumenttypen eine Erläuterung der dynamischen Anpassungen dieser Dokumente.

Rückführung in dynamische Dienstbeschreibungen und Dienstgütevereinbarungen

Die Metaqualität einer Dienstbeschreibung lässt sich auf Ebene der NFE oder auf Ebene der gesamten Beschreibung als Annotationen auf den NFE ausdrücken. Diese Rückführung muss durch die verwaltende Registry-Komponente unterstützt werden. Die Rückführung hat einen direkten Effekt auf die Auswahl von Diensten und indirekt auf daraus abgeleitete SLA-Vorlagen.

Die Metaqualität einer SLA-Vorlage berechnet sich aus der Wahrscheinlichkeit, dass aus dieser Vorlage abgeleitete Verträge über die gesamte Laufzeit eingehalten werden. Die Rückführung muss durch die verwaltende Komponente,

z.B. eine erweiterte Registry oder eine gesonderte SLA-Verwaltung, unterstützt werden. Sie hat einen indirekten Effekt auf die zukünftige Bildung von Verträgen.

Die Metaqualität eines aktuellen Vertrags berechnet sich aus der Verletzungshäufigkeit seiner Bestimmungen. Die Rückführung muss durch die SLA-Verwaltung unterstützt werden. Sie hat einen indirekten Einfluss auf die Bildung der Metaqualität der korrespondierenden Vertragsvorlage, sofern eine existiert, sowie auf den laufenden Vertrag.

Angepasste Dienstbeschreibungen

Die NFE-Angaben in Dienstbeschreibungen können automatisch oder semiautomatisch aktualisiert werden. Eine automatische Modifikation wird verzögerungsfrei anhand von Monitoringdaten (nfe_{mon}) bzw. davon abgeleiteten Vorhersagedaten ($nfe_{montrend}$, $nfe_{monpred}$) durchgeführt. Dafür bieten sich gleitende Mittelwerte an, um ein möglichst repräsentatives Bild der jeweils aktuellen Situation zu erhalten.

Die Dynamisierung von Dienstbeschreibungen erfordert konzeptionelle Erweiterungen in Komponenten zur Verwaltung der Beschreibungen (Registries). Insbesondere muss es möglich sein, in zwei Schritten zuerst für jede NFE den neuen Wert zu bestimmen und diesen anschließend programmatisch an die Registry zu übermitteln. Für eine semiautomatische Aktualisierung wird zudem ein interaktiver Prozess zur Einsicht und Übernahme der neuen Werte durch den Dienstanbieter benötigt.

Angepasste Vorlagen für Dienstgütevereinbarungen

Analog zu den angepassten Dienstbeschreibungen ist wiederum eine Entscheidung zwischen automatisch und semiautomatisch ausgeführten Aktualisierungen möglich. Für beide Fälle sind über SLA-Vorlagen hinaus weitere SLA-bezogene Dokumententypen notwendig, die an dieser Stelle eingeführt und erläutert werden sollen:

- SLA-Vorlagen-Vorschlag (*SLA Template Proposal/SLATP*)
- *SLA-Vorlagen-Metaeinschränkungen (SLA Template Meta Constraints/SLATMC)*

Sollen Aktualisierungen an SLA-Vorlagen nicht automatisch für zukünftige SLA-Aushandlungen zur Verfügung stehen, sondern erst durch den Dienstanbieter begutachtet und akzeptiert werden, sind Vorschläge für SLA-Vorlagen (SLATP) eine Lösung. Hierbei wird die Verwaltung von SLA-Vorlagen, also eine Registry oder ein SLA-Manager, um Einträge für noch nicht freigegebene SLA-Vorlagen ergänzt. Desweiteren muss der Anbieter die Möglichkeit bekommen, derartige Vorschläge mit der bereits existierenden Variante zu vergleichen und eine direkte Zustimmung, Zustimmung mit Änderungen oder Ablehnung vorzunehmen.

SLATP-Dokumente haben exakt den gleichen Aufbau wie akzeptierte SLA-Vorlagen. Eine Erläuterung der Struktur entfällt damit.

Listing 3.9: SLATMC-Dokument

```

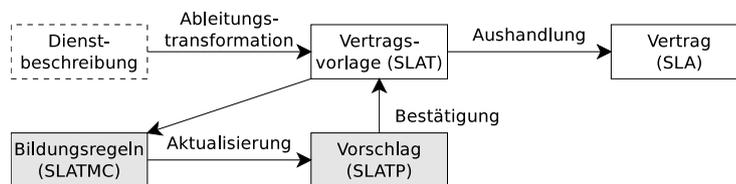
1 <slatmc>
2   <slot ref="responsetime" upperbound="1.2" lowerbound="0.8"
3     />
4   <slot ref="availability" value="0.9"/>
5 </slatmc>

```

Ein SLATMC-Dokument definiert, wie ausgehend von einer Dienstbeschreibung SLA-Vorlagen definiert werden. Es fließt in die ansonsten nicht profilbildende Transformation von NFE-Angaben in Dienstbeschreibungen zu SLO-Vorlagen in SLA-Vorlagen ein und ermöglicht damit die Bildung von Profilen durch Einschränkungen der Wertebereiche. Insbesondere können geschäftlich relevante Entscheidungen zu mehr oder weniger strikt kalkulierten angebotenen Dienstgütekriterien durch SLATMC-Dokumente ausgedrückt werden. Somit sind SLATMC-Dokumente im Gegensatz zu den eigentlichen SLA-Vorlagen nicht für den Dienstkonsumenten einsehbar, sondern in der Sichtbarkeit auf den Dienstanbieter beschränkt.

Das folgende Syntaxbeispiel 3.9 definiert die Transformation von NFE zu SLO-Vorlagen (SLOT). Die NFE **responsetime** wird als Wertebereich angegeben, aus dem der potentielle Konsument im Rahmen der SLA-Aushandlung einen garantierbaren Wert auswählen kann. Die NFE **availability** hingegen wird auf einen festen Wert gesetzt, der einen Sicherheitsabstand von 10% auf den öffentlich bekanntgegebenen Verfügbarkeitswert aufweist.

Die Erweiterung der Verarbeitung von SLA-Vorlagen um SLATP und SLATMC wird in Abbildung 3.26 zusammengefasst.

**Abbildung 3.26:** Vorschläge und Bildungseinschränkungen für SLA-Vorlagen

Angepasste aktive Dienstgütevereinbarungen

Werden ausgehandelte Garantien einer Dienstgütevereinbarung häufig verletzt, so ist die alleinige Bereitstellung neuer SLA-Vorlagen nicht ausreichend, da sowohl eine Terminierung des laufenden SLA als auch eine erneute Aushandlung mit Kosten verbunden sein kann. Stattdessen bietet sich eine Neuaushandlung von Vertragsbestandteilen an, sofern diese Möglichkeit im ursprünglichen Vertrag vorgesehen worden ist. Im Ergebnis steht dann ein modifizierter Vertrag, der besser an die Laufzeitsituation angepasst ist und somit zukünftig weniger Verletzungen aufweisen wird. Die Neuaushandlung als Adaptionmechanismus ist bereits Gegenstand existierender Forschungsarbeiten und soll an dieser Stelle als integraler Bestandteil des Konzepts aufgeführt werden.

3.4.5 Zusammenfassung der Verfahren

Ein Beispiel soll das Zusammenspiel der Verfahren erläutern. Gegeben sei ein Dienst, der gemäß der Top-Down-Methodologie aus einer WSDL-Datei entwickelt worden ist und keine weiteren Bestandteile außer der Implementierung und der syntaktischen Dienstbeschreibung aufweist. Nach bereits einem Dienstaufwurf lassen sich nun Aussagen über NFE über eine erweiterte Dienstbeschreibung wie WSML hinzufügen, die zudem initial eine vollständige Validität aufweisen. Der Entwickler kann anschließend diese WSML-Datei importieren, um sie weiter zu annotieren oder zu verfeinern. Das Beispiel ist angelehnt an die Übersichtsgrafik zu den Verfahren in Abbildung 3.27 als Ablaufsequenz dargestellt.

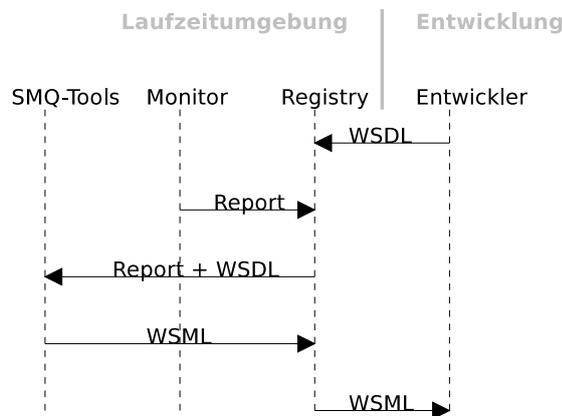


Abbildung 3.27: Beispielhafter Ablauf der Bestimmung und Nutzung der SMQ

Um die Einbindung der Verfahren in allgemeine IoS-Architekturen zu ermöglichen, sollen im nächsten Schritt die an den Abläufen beteiligten Komponenten und ihre Rollen sowie die damit verbundene dynamische Verarbeitung der Metaqualitätsangaben tragenden Dokumente beschrieben werden.

3.5 Einbindung der Verfahren in IoS-Plattformen

Sowohl die vorgestellten Verfahren als auch die dynamischen Dokumente müssen für den reibungslosen Betrieb in einer Dienstplattform verankert werden. Dazu wird in diesem Abschnitt sowohl ein dynamisches Dienstmodell vorgestellt als auch eine abstrakte Sicht auf notwendige Plattformfunktionalität gegeben, die alle Architekturen erfüllen müssen, sofern auf ihrer Basis Metaqualitätsangaben generiert und genutzt werden sollen. Schließlich werden dedizierte SMQ-Komponenten konzeptionell vorgestellt, mit denen die identifizierten Zielarchitekturen erweitert werden können. Eine konkrete Umsetzung einer Architektur für das IoS erfolgt im nächsten Kapitel.

3.5.1 TSM: Modell für handelbare Dienste mit dynamischen Dokumenten

Eine generalisierte Darstellung von Dienstleistungen auf einer Modellebene ist derzeit nicht vollständig möglich. Die Forschung zu Produkten und herkömm-

lichen Dienstleistungen im wirtschaftswissenschaftlichen Umfeld wurde durch die starke Verbreitung von SOA-Techniken um den Begriff Dienstwissenschaft („Service Science“) erweitert, welcher interdisziplinär Elemente aus der Informatik aufgreift. Dieser weit gefasste, gesellschaftsorientierte Begriff zur Durchdringung der IT-gestützten Wirtschaftswelt mit Dienstleistungen wird derzeit nicht einheitlich interpretiert. Es existieren weiterhin anbieterspezifische Varianten wie *Service Science, Management and Engineering* (SSME) [108], *Systems and Services Sciences* [82] und *Service Engineering* [14] mit eigenen Zielstellungen. Somit eignet sich der weitläufige Begriff Dienst aus dem Umfeld von Service Science nicht für die vorliegende Arbeit.

Auch der eher technisch angelegte und enger gefasste Begriff Internet der Dienste („Internet of Services“) wird derzeit zwar mit definierten Zielen verbunden, hat aber bisher ebenfalls noch keinen einheitlichen Dienstbegriff hervorgebracht. Selbst bei einer Einschränkung auf Web Services im Sinne von Softwarekomponenten divergieren die Vorstellungen und Anforderungen. Die Festlegung eines Dienstbegriffs und eine detaillierte Modellierung von Dienstleistungen ist jedoch für eine eindeutige Interpretierbarkeit von Forschungsergebnissen zu Diensten unverzichtbar.

Aus diesem Grund führt dieser Abschnitt ein Dienstmodell TSM (*Tradeable Service Model*) ein, welches sich einerseits an vorhandenen impliziten Vorstellungen orientiert, andererseits aber auch die notwendigen Eigenschaften für den Einsatz im Internet der Dienste aufweist. TSM definiert einen handelbaren konventionellen, elektronischen oder hybriden Dienst als physisch verteilbares digitales Paket, das aus heterogenen, teilweise dynamischen Artefakten wie Implementierung und Beschreibung besteht.

Neben der eigentlichen Dienstimplementierung im Fall von Web Services enthält ein TSM zwingend eine deklarative Beschreibung. Hierbei sind sowohl rein syntaktische Formate wie WSDL oder WADL möglich, als auch semantische Dienstbeschreibungen in WSML oder USDL oder weitergehende Spezialformate wie CQML+ zur Angabe nichtfunktionaler Eigenschaften. Desweiteren können Vertragsvorlagen nach WS-Agreement und ähnlichen Formaten beigefügt werden. Zur Steigerung der Attraktivität des Bezugs aus Dienstverzeichnissen können Informationen über passende Clients sowie grafische Abbildungen zur Funktionalität hinterlegt werden. Zudem sind weitergehende Annotationen wie etwa GUIDD für die automatische Formulargenerierung zur ad-hoc-Nutzung des Dienstes integrierbar. Insbesondere sind auch Metaqualitäts-Reports als qualitätsbezogene Annotationen möglich. Schließlich sind weitere Dokumententypen wie Dokumentationen im HTML-Format oder als PDF denkbar, die allerdings in dieser Arbeit nicht betrachtet und somit nicht explizit als Teil des TSM modelliert werden. Die äußere Struktur eines Dienstpakets mit Dokumenten und deren Kardinalitäten gemäß TSM wird somit zusammenfassend in der Abbildung 3.28 gezeigt.

Das Dienstmodell TSM unterstützt insbesondere die mehrstufige Beschreibung nichtfunktionaler Eigenschaften (NFE) und die Verknüpfung mit Anbietern und Benutzern durch Dienstgütevereinbarungen (*Service Level Agreements*, SLA). Eine semiformalisierte Darstellung der Angaben in ihren jeweiligen Trägerdokumenten ist als UML-Diagramm in Abbildung 3.29 zu sehen. Entnommen werden die Angaben sowohl den Dienstbeschreibungen als auch den Vertragsvorlagen im Dienstpaket. Damit wird die Berechnung der Metaqualität SMQ für alle Trägerdokumente auf den Ebenen der einzelnen Operationen, der Dienste

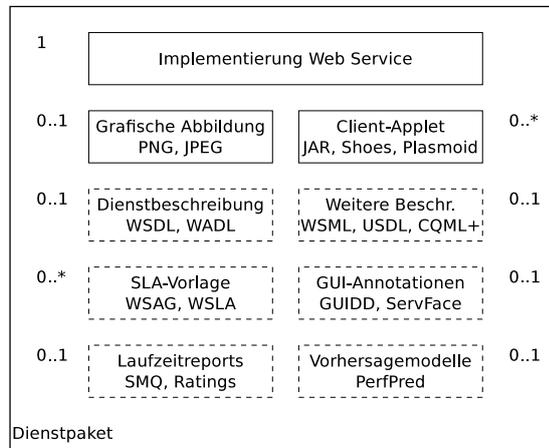


Abbildung 3.28: Mögliche Inhalte eines Dienstpaketes bei Konformität zum Dienstmodell TSM

und der Dienstpakete ermöglicht.

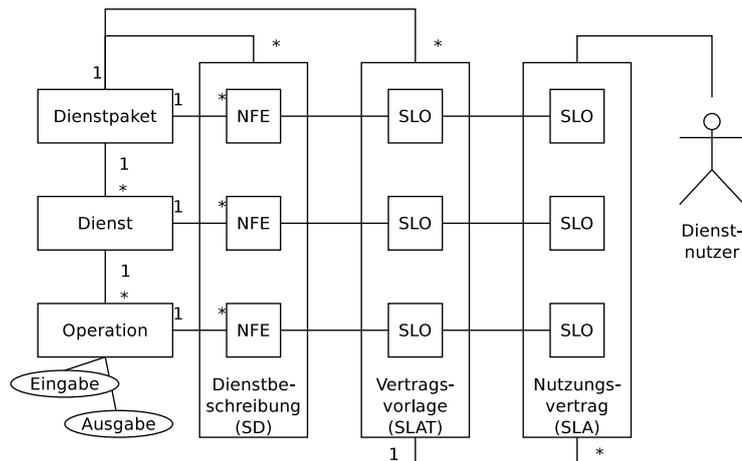


Abbildung 3.29: Die Struktur von NFE im Dienstmodell TSM

Abbildung von Diensten auf das Dienstmodell TSM

Um die Heterogenität angebotener Dienste auf Hostingplattformen unterstützen zu können, ohne jedoch die Komplexität der Plattformen unnötig erhöhen zu müssen, bietet sich eine Abstraktion der jeweiligen Merkmale in einem gemeinsamen Modell und damit eine Homogenisierung der wichtigsten Differenzierungsmerkmale an. Dies betrifft die Beschreibung eines Dienstes und darauf aufbauender Vertragsvorlagen und Verträge, die Installation (Deployment), den Aufruf (Protokolladapter), die Ausführung, sowie die Überwachung und Anpassung der Ausführung. Das Modell TSM eignet sich für eine Abstraktion aller zu einem Dienst gehörenden Artefakte.

Für jedes konkrete Dienstpaket muss eine Abbildungsvorschrift existieren, die die Projektion von Dateien auf TSM-Artefakte vornimmt. Dabei existieren sowohl deterministische als auch heuristische Projektionen.

Am Beispiel der verbreiteten verteilbaren Axis-Web-Services soll dies erläutert werden. Diese in Java entwickelten Dienste nutzen die Bibliothek Axis für die Umsetzung der SOAP-Kommunikation. Sie laufen als Servlets in einem Servletcontainer. Ihr Aufbau entspricht demnach der WAR-Spezifikation. Oftmals enthalten diese Archive die WSDL-Dateien der entwickelten Dienste. Dies ist jedoch nicht vorgeschrieben, so dass der statische Zugriff auf die Dienstbeschreibung nicht garantiert werden kann. Alternativ definiert Axis den Zugriff zur Laufzeit über spezielle URLs in der Form `http://serverpfad/servletpfad/services/dienstname?wsdl`.

Die heuristische Projektion verursacht unnötige Komplexität und erhöhten Ressourcenverbrauch in der Handhabung der Dienstpakete. Empfehlenswert sind demnach deterministische Projektionen durch klar strukturierte Dienstpakete wie beispielsweise SAR für USDL-beschriebene Dienste.

3.5.2 Anforderungen an Dienstplattformen

Die Modelle und Verfahren zur Verbesserung der Metaqualität auf Dienstplattformen lassen sich nur dann effizient anwenden, wenn die Plattformen bestimmte architektonische und funktionale Voraussetzungen erfüllen. Vereinzelt sind diese bereits erwähnt worden. An dieser Stelle sollen sie noch einmal zusammenfassend aufgezählt werden.

1. Dynamisierte NFE-Trägerdokumente: Sowohl Dienstbeschreibungen als auch SLA-Vorlagen und SLAs müssen derart verwaltet werden, dass jederzeit zur Laufzeit eine automatische oder semiautomatische Aktualisierung möglich ist.
2. Query-Schnittstellen: Der Zugriff auf die NFE-Trägerdokumente, inklusive dynamisch erzeugter Daten wie Monitoringreports und Benutzerbewertungen, muss über Web-Service-Schnittstellen möglich sein.
3. Homogenisierungsschichten: Die Heterogenität von Dienstpaketen und NFE-Trägerdokumenten muss durch eine Homogenisierung und somit eine Abbildbarkeit auf die konzipierten Modelle TSM und CPEM handhabbar gemacht werden.

3.5.3 Erweiterung von IoS-Plattformen um SMQ-Komponenten

Die Methodik zur inkrementellen Verbesserung der Metaqualität setzt eine erweiterbare Dienstplattform voraus. Üblicherweise sind dabei Plattformbestandteile mit Funktionen zur Dienstregistrierung, Suche nach Diensten, Vertragsgestaltung und überwachten Ausführung bereits enthalten (vgl. Abschnitt 2.5.4).

Die Umsetzung der konzipierten Lösung erfordert auf der Architekturebene die Einführung eines neuen Plattformdienstes zur Abfrage von sowohl statischen als auch dynamischen Daten zu einem Dienst. Sie erfordert außerdem auf der Prozessebene die Einführung einer Feedbackschleife. Sowohl die Komponente als auch der Prozess werden an dieser Stelle abstrakt dargestellt. Eine Umsetzung der Komponente folgt im nächsten Kapitel.

Einbindung in verteilte IoS-Plattformen

Wie in der Illustration 3.30 dargestellt, wäre aufgrund der losen Kopplung an die Schnittstellen der Plattformdienste auch ein autark betriebenes Portal zur Sammlung und Darstellung der Metaqualität über möglichst viele Plattformbetreiber hinweg vorstellbar. Damit wäre sogar die Bildung einer globalen Metaqualitätsmetrik für das gesamte Internet der Dienste möglich. Allerdings führt ein solches Konzept zu einer Vielzahl zusätzlicher Herausforderungen, die insbesondere die Sicherheit zur Rückführung der Werte in die Portale sowie das Betreiberkonzept betreffen. Als realistischeres Konzept wird daher die plattformintegrierte Umsetzung gewählt.

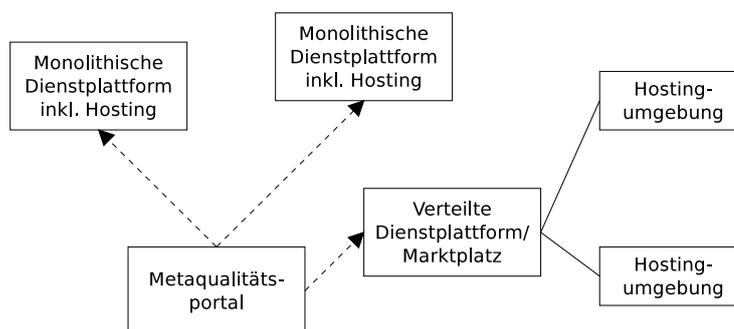


Abbildung 3.30: Entwurf für einen separaten Betrieb eines Metaqualitäts-Portals

Metadaten-Korrelationsdienst (MDCS)

Dieser Metadaten-Korrelationsdienst (*Metadata Correlation Service*, MDCS) dient als zentrale Abfrageschnittstelle zu spezialisierteren Metadaten Diensten wie Monitoringanfrageschnittstelle, Dienstbeschreibungsverwaltung, Benutzerbewertungsdienst und Vertragsverwaltung. Zur Handhabung heterogener Dienstarfakte wird als Teil des MDCS eine Abstraktionsschnittstelle für die Extraktion und Modifikation von NFE (*Service Abstraction Layer*) benötigt.

Die Funktionalität des MDCS umfasst eine zeit-, ereignis- und abfragegesteuerte Berechnung der Metaqualität von Diensten. Liegt diese Information noch nicht vor, wird sie periodisch oder bei erkannten Aktualisierungen der zugrundeliegenden Daten ereignisgesteuert, spätestens jedoch bei Eintreffen der ersten Abfrage, berechnet und in einem Zwischenspeicher vorgehalten. Für weitere analytische Abfragen steht sie somit direkt zur Verfügung. Die Größe des Zwischenspeichers ist üblicherweise begrenzt und beträgt einen Bruchteil der abfragbaren Informationsmenge. Es wird jedoch keine bestimmte Größe und keine spezielle Cacheverwaltungsstrategie vorausgesetzt.

Die errechnete Metaqualität kann und sollte als nichtfunktionale Eigenschaft in die Dienstbeschreibung rückgeführt werden. Dies bedingt eine Erweiterung des Service Abstraction Layers um auf heterogene Dienstbeschreibungsformate ausgerichtete Schreibfunktionalität zusätzlich zu den Parsern zur Extraktion der Eigenschaften. Eine solche Erweiterung ist derzeit nicht als Teil dieser Arbeit vorgesehen.

Desweiteren kann die Metaqualität sowohl durch Administratoren als auch durch Dienstentwickler programmatisch abgefragt werden. Administratoren ha-

ben ein Interesse daran, unter den angebotenen Diensten diejenigen selektieren zu können, die starke Abweichungen gegenüber ihrer beworbenen Leistungsfähigkeit bieten. Entwickler wiederum haben ein Interesse daran, ihre unterspezifizierten Dienste mit besseren Beschreibungen zu versehen. Aus diesem Grund sollte die Abfrage der Metaqualität in die Dienstentwicklungsumgebungen (*IDE*) der Entwickler möglich sein. Durch die Rückführung der Metaqualitätsdaten in die Dienstbeschreibungen als NFE wäre dies prinzipiell ohne zusätzliche WS-Schnittstellen möglich, indem stattdessen auf die Schnittstelle der Discovery zugegriffen wird. Dennoch bietet der Zugriff über die Schnittstelle des MDCS Vorteile durch eine geringere zu übertragende Datenmenge und weitere Optimierungen, und erspart die bereits erwähnte Erweiterung der Abstraktionsschnittstelle um Schreibzugriffe.

Der Metadaten-Korrelationsdienst (MDCS) ist eine abstrakte Erweiterung von Dienstplattformen im Sinne einer dienstorientierten erweiterbaren Middleware. Er führt dynamisch generierte Informationen über die Dienstausführung aus Monitoring und Dienstbewertung zusammen und gleicht diese Informationen mit statischen Informationen ab. Dabei werden Dienstbeschreibungen, SLA-Vorlagen und aktive SLA berücksichtigt. Der MDCS kontaktiert dazu die Registry und die SLA-Verwaltung, um eine Liste verfügbarer Beschreibungsdokumente für einen ausgewählten Dienst zu erhalten. Jedes Dokument wird über eine Abstraktionsschnittstelle so verarbeitet, dass alle Identifikatoren und Informationen über NFE extrahiert werden können. Für jeden Identifikator werden die Plattformdienste für die Abfrage von Monitoring- und Bewertungsinformationen abgefragt und die Bewertungen und Messungen gesammelt. MDCS führt anschließend einen namensbasierten Vergleich der NFE durch und berechnet über eine Distanzfunktion die Diskrepanz zwischen ermitteltem und ursprünglich angegebenem Wert. Die Ergebnisse werden in einer SMQ-Metrik für den Dienst zusammengefasst und für zukünftige Zugriffe gespeichert. Diese Methode erlaubt es dem MDCS, die Dienstbeschreibungsqualität und die Brauchbarkeit von SLA-Vorlagen über die Zeit zu verfolgen.

Für die Angabe der Aggregationsregeln ist die Verwendung einer ausdrucksstarken Regelsprache denkbar, um geschäftliche Belange wie die Kriterien für die Berechnung der Diskrepanz von Implementierungsdetails wie dem Web-Service-Zugriff und der Konfiguration des MDCS getrennt zu halten. Diese Separierung ist derzeit nicht Teil des angedachten Konzepts.

Die abstrakte Innenarchitektur von MDCS ist in Abbildung 3.31 dargestellt.

Prozess zur Sammlung und Aggregation von Rücklaufdaten

In autonomen Systemen, zu denen Dienstplattformen mehrheitlich gehören, werden Kontrollschleifen oft mit vier Phasen modelliert: Sammlung, Analyse, Entscheidung und Aktion [13]. Das Konzept des MDCS ist auf diese generische Herangehensweise abgestimmt und lässt sich somit in Ablaufprozesse auf den Plattformen integrieren. Messung und Aggregation durch Monitoring (Sammlung), NFE-Vorhersage (Analyse), Bestimmung der Metaqualität (Entscheidung), Rückführung der SMQ-Angaben und Anpassung der NFE-Angaben (Aktion) als Abschluss der Rückkopplung (*Feedback*).

Die Grafik 3.32 gibt einen Überblick über die Ansatzpunkte und Auswirkungen von kurz-, mittel- und langfristigen Strategien zur Qualitätsverbesserung von Diensten durch Feedbackschleifen. Zur Laufzeit werden Informationen zur

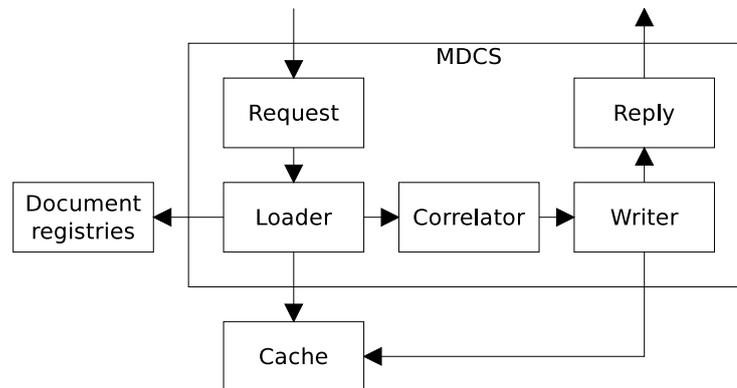


Abbildung 3.31: Prinzipieller Aufbau des Metadaten-Korrelationsdienstes

Dienstnutzung sowohl auf einer Dienst- als auch einer Dienstauftrufebene gesammelt. Jede Ausführung kann dabei an einen Vertrag gebunden sein. Messungen auf Dienstebene sind dabei als vertragsübergreifend zu verstehen. Vertragsverletzungen können zu sofortigen und kurzfristigen Korrekturen durch Laufzeitaaptionsmechanismen führen. Nach dem Auslaufen oder der expliziten Terminierung eines Vertrags kann der Dienstanutzer für diesen Bewertungen abgeben. Zusammen mit den aggregierten Monitoringergebnissen bilden sie Rücklaufdaten, die mit der Vertragsvorlage verknüpft werden, von welcher der zugehörige Vertrag abgeleitet worden war. Diese Rücklaufdaten können zu semiautomatischen Anpassungen der Dienstbeschreibungen und Vertragsvorlagen im Sinne mittelfristiger Dienstqualitätsverbesserungen dienen [105].

Bis zu dieser Stelle liegen bereits folgende Rücklaufdaten für jeden Vertrag vor: Eine Menge an gemeinsamen Messungen auf Dienstebene, eine Menge an vertragspezifischen Messungen auf Aufruf- bzw. Dienstinstanzebene, und eine Bewertung. Auf einer weiteren Aggregationsebene können nun die Rücklaufdaten über alle Verträge eines Dienstes zusammengeführt werden. Das Aggregat dient dabei der langfristigen Verbesserung sowohl der Qualitätsspezifikation als auch der Anpassung der Implementierung als Teil des Dienstentwicklungsprozesses. Zu diesem Zweck werden externe Zugriffsmöglichkeiten auf die Rücklaufdatenbasis aus den Dienstentwicklungsumgebungen (IDE) benötigt. Der konzeptionelle Ansatz des MDCS folgt dem Architekturstil der Plattformdienste, welcher sich für solche Aufgaben eignet [97]. Die Verwendung der sich ergebenden Dienstschnittstellen wird dabei für Innovations- und Entwicklungswerkzeuge empfohlen.

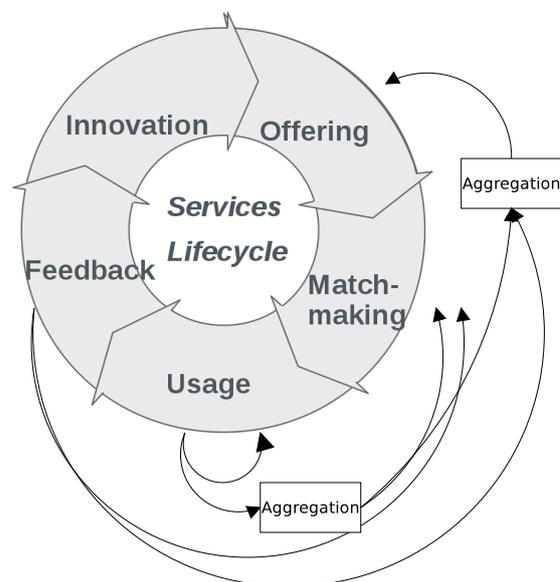


Abbildung 3.32: Feedbackschleifen im System

Kapitel 4

Plattform für das Internet der Dienste

Dieses Kapitel geht detailliert auf die Konzeption und Umsetzung einer Dienstplattform und ihrer modularen Plattformdienste ein. Die angestrebte Flexibilität der Plattform zielt auf ihre Nutzung als Basisinfrastruktur in unterschiedlichen anwendungsabhängigen Konfigurationen im Internet der Dienste. Die Berücksichtigung der Metaqualität von Diensten ist eine solche Anwendung, die sich über eine Erweiterung der Basisplattform realisieren lässt.

Die Konzeption der Architektur mit ihren Entwurfsrichtlinien, Anforderungen und Abläufen wird unter Berücksichtigung dieser Zielstellung zuerst dargestellt. Nach der Dekomposition des angestrebten Funktionsumfangs in einzelne Bausteine auf Systemebene folgt darauf eine Vorstellung der umgesetzten Bestandteile als modulare Plattformdienste. Auf Basis der Komposition der Plattformdienste wird die integrierte Basisdienstplattform im Sinne einer dienstorientierten Middleware vorgestellt. Ausgewählte über Erweiterungen und Konfigurationen gebildete Derivate der Plattform werden dabei ebenso erwähnt. Als eine spezialisierte Ausprägung werden schließlich erweiterte Plattformdienste zur Gewährleistung eines qualitativ hochwertigen Dienstangebots integriert. Diese setzen die vorgestellte Methodik zur inkrementellen Qualitätsverbesserung über die Metaqualität um.

Die Aufteilung des Kapitels wird durch die Abbildung 4.1 veranschaulicht.

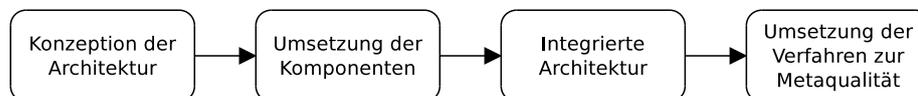


Abbildung 4.1: Aufteilung des Kapitels zu Konzept und Umsetzung der Dienstplattform

4.1 Konzeption der Architektur

Der Entwurf einer dedizierten Dienstplattform mit der Eignung als Basissystem für ein Internet der Dienste wird in diesem Abschnitt auf architektonischer und prozessorientierter Ebene beschrieben. Die Darstellung der Architektur gliedert

sich in die anforderungsbezogenen Entwurfsrichtlinien, die Beschreibung der abstrakten Plattformbestandteile sowie das angestrebte Zusammenspiel derer zu einer Gesamtarchitektur. Auf Basis der Zusammenstellung werden anschließend die benutzerbezogenen interaktiven Prozesse zur Bereitstellung und Ausführung von Diensten sowie systemseitig autonom ablaufende Prozesse demonstriert.

Die Konzeption der Architektur orientiert sich an existierenden Plattformen für das Internet der Dienste. Sie übernimmt den grundsätzlichen Dreiecksaufbau einer SOA, spezifiziert allerdings deutlich konkreter die zwischen Anbietern und Nutzern von Diensten befindlichen Systeme an der Spitze des Dreiecks. Dabei werden sowohl rollenbezogene Funktionen als auch systemtopologische Merkmale explizit betrachtet. Die resultierende abstrakte Architektur mit einer logisch zentralisierten Verwaltung und dezentralisierten Ausführungsumgebung wird in Abbildung 4.2 skizziert.

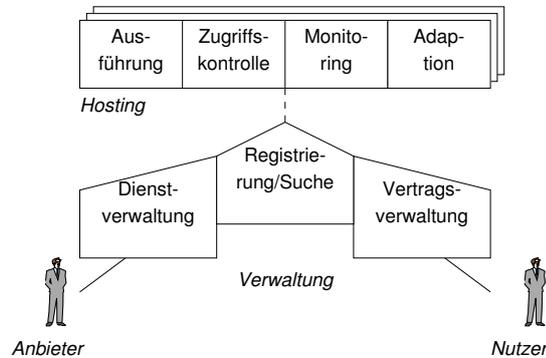


Abbildung 4.2: Abstrakte Architektur für die Basisdienstplattform

4.1.1 Entwurfsrichtlinien

Um die Anforderungen einer vertragsgebundenen und adaptiven Dienstausführung im Internet der Dienste erfüllen zu können, ist eine spezielle Infrastruktur notwendig. Eine mit den meisten in der Literatur und aktuellen Forschungsarbeiten vorhandenen Modellvorstellungen eines Internets der Dienste kompatible, jedoch wesentlich konkretere, Definition der Entwurfsrichtlinien ist in Abbildung 4.3 dargestellt. Diese primären und sekundären Anforderungen dienen als Rahmen für die Entwicklung der Architektur.



Abbildung 4.3: Primäre und sekundäre Entwurfsschwerpunkte

Zentrales Merkmal ist die Berücksichtigung von Dienstgütevereinbarungen vor, während und nach der Dienstausführung. Die Ausführung selbst sollte überwacht und anpassbar stattfinden.

Die Plattform muss in allen Phasen der Dienstbereitstellung und -nutzung eine nachvollziehbare und benutzerfreundliche Handhabung heterogener Dienste unterstützen. Eine Dienstplattform ohne ausdrückliche Einbeziehung von Benutzern in alle wichtigen Abläufe wird nur wenig Akzeptanz finden [19].

Die Notwendigkeit einer umfassenden Unterstützung von Heterogenität bezieht sich auf die von Dienstentwicklern bereits verwendeten Implementierungsansätze, die somit ohne Neuerwerb von Wissen weiterhin angewandt werden können [99]. Insbesondere müssen Dienste als abgeschlossene Entitäten in Paketform verarbeitet werden können, um im Gegensatz zu eher stationären Diensten in starren Umgebungen klassischer SOA-Architekturen eine flexible Verteilbarkeit und Handelbarkeit zu ermöglichen.

Um zusätzlich die vorteilhaften Eigenschaften von Web Services, beispielsweise die lose Kopplung und Komponierbarkeit, auf die Plattform selbst zu übertragen, wurde als Architekturstil eine modulare Plattform auf Basis einzelner Plattformdienste und weiterer Komponenten gewählt [97]. Diese Entscheidung fördert speziell auch die Wiederverwendung einzelner Funktionen in zukünftigen Forschungsprojekten.

Schließlich ist es für die Verbreitung und Akzeptanz einer solchen Plattform förderlich, wenn insbesondere bei Verwendung unterschiedlicher Implementierungsansätze die Installation und Konfiguration keinen hohen Zeitaufwand verursacht. Aus dem Grund wird eine Integration in die Installationsroutinen von Betriebssystemen mit möglichst hohem Automatisierungsgrad für die Konfiguration von Datenbankverbindungen, Zugriffs-URLs und funktionalen Merkmalen angestrebt. Ein positiver Nebeneffekt ist die einfache Erstellung verteilter Dienstumgebungen durch eine mehrfache zeitsparende Installation des gleichen Systems.

Als weitere sekundäre Anforderungen werden implizit eine hohe Skalierbarkeit der entwickelten Lösung durch verteilte Dienstaufführung sowie eine allgemeine Verfügbarkeit der entstehenden Umsetzung als Software angedacht. Zusätzlich zu der Systemintegrationsschicht soll optional eine Präsentationsschicht hinzugefügt werden können. Diese dient vor allem der Demonstrierbarkeit der Plattformfähigkeiten anhand einer desktopintegrierten, vorkonfigurierten Benutzerumgebung anhand von Beispielszenarien.

4.1.2 Anforderungen an Plattformdienste

Zur Umsetzung der Anforderung einer modularen Dienstplattform wird die angestrebte Funktionalität in Teilfunktionen dekomponiert. Jede Teilfunktion soll dabei durch einen eigenen Plattformdienst sowohl autark genutzt werden können als auch als Komponente im Gesamtsystem repräsentiert werden. Der Plattformdienst soll sowohl programmatischen als auch, falls sinnvoll, interaktiven Zugriff ermöglichen. Die modulare Aufteilung fördert die Flexibilität sowohl hinsichtlich der Zusammenstellung der Plattformdienste als auch hinsichtlich möglicher Ersetzungen durch funktionsäquivalente, aber leistungsfähigere Komponenten.

Die folgenden Teilfunktionen können für eine Basisplattform für handelbare Dienste identifiziert werden:

1. Bereitstellung von Diensten durch Anbieter. Diese Komponente nimmt ein Dienstpaket entgegen und installiert es in der Plattform. Sämtliche Artefakte im Paket müssen in die jeweils passenden Datenbanken übermittelt

werden. Der Anbieter muss die Möglichkeit bekommen, das Dienstangebot zu verändern oder den Dienst wieder zu entfernen.

2. Dienstsuche durch Konsumenten über die Menge aller angebotenen Dienste. Die Suche soll sowohl nach funktionalen als auch nach nichtfunktionalen Kriterien möglich sein.
3. Vertragsaushandlung durch Konsumenten zu einem ausgesuchten Dienst. Auf Basis von Vertragsvorlagen, die zu einem ausgesuchten Dienst gehören, soll die Einstellung von NFE-Grenzwerten möglich sein.
4. Dienstaufwurf durch Konsumenten mit einem zugehörigen oder ad-hoc generierten Client. Dabei soll vor der Ausführung die Existenz eines Vertrags zu dem aufgerufenen Dienst geprüft werden, sofern ein Vertrag erforderlich ist.
5. Überwachung und Anpassung von Diensten durch das System. Parallel zur Ausführung werden alle automatisch prüfbaren vertraglich vereinbarten NFE-Grenzwerte überprüft. Im Fall von Diskrepanzen soll versucht werden, die Ausführung so zu adaptieren, dass Vertragsverletzungen minimiert werden.
6. Vertragsverwaltung durch Konsumenten. Der Konsument soll auch abgeschlossene Verträge überprüfen, neu aushandeln und terminieren können.
7. Dienstbewertung durch Konsumenten zu einem abgelaufenen Vertrag. Jeder ausgehandelte NFE-Grenzwert soll dabei separat bewertbar sein, um Unstimmigkeiten eingrenzen zu können.
8. Zugriff auf alle Dienstartefakte im System. Dies ist nötig, um statistische Analysen durchführen und Erweiterungen anschließen zu können.
9. Dienstverbesserung durch Anbieter. Laufzeitinformationen sollen dem Anbieter helfen, seinen Dienst und die zugehörigen Artefakte kontinuierlich zu verbessern.

Die Umsetzung der Teilfunktionen als Plattformdienste wird nachfolgend durch eine Beschreibung der Nutzungsabläufe und durch konkrete Komponenten dargestellt.

4.1.3 Prozessabläufe

Zur Umsetzung eines Lebenszyklus von Dienstangebot und -nutzung findet zur Laufzeit ein Datenaustausch zwischen den Plattformdiensten statt. Die Zugriffe auf die Dienstschnittstellen auf der Dienstplattform erfolgen nicht willkürlich, sondern unterliegen gewissen Mustern, die mit einer Choreographie oder Orchestrierung vergleichbar sind. Die wichtigsten dieser Prozesse sowohl zwischen Benutzern und Plattform als auch innerhalb der Plattform sollen an dieser Stelle erläutert werden.

Prozesse mit Nutzerinteraktion

Prinzipiell lassen sich drei Prozesse definieren, die notwendig sind, um den Austausch und die Handelbarkeit von Diensten zu erlangen: Die Bereitstellung eines Dienstes durch einen Anbieter oder Entwickler, die Erstellung und Verwaltung eines Vertrages durch einen Benutzer, sowie der eigentliche Dienstaufwurf ebenfalls durch einen Benutzer.

Der auch als „Deployment“ bezeichnete Prozess zur Bereitstellung eines Dienstes deckt ausgehend von dessen Entwicklung die Installation und Bestimmung der Nutzungsbedingungen ab. Die Installation erfolgt als Dienstpaket entsprechend dem Modell TSM. Über die Laufzeit lassen sich darüber weiterhin Nutzungsdaten wie etwa die Popularität bestimmen und Aktualisierungen vornehmen. Sobald ein Dienstpaket installiert ist und die konstituierenden Artefakte in den sie verwaltenden Plattformdiensten hinterlegt sind, können bereits statische Analysen auf den Dokumenten vorgenommen werden.

Die Erstellung, Verwaltung und Bewertung von Dienstnutzungsvereinbarungen, auch als „Contracting“ bezeichnet, dient dem Nutzer zur Definition von Gewährleistungsanforderungen [100]. Im Rahmen der Erstellung werden entweder basierend auf den im Dienstpaket vorhandenen Vertragsvorlagen oder vorlagenfrei neue SLAs ausgehandelt, welche bei ihrer Fertigstellung an den Monitor übermittelt werden. Aufrufunabhängige NFE wie die Verfügbarkeit eines Dienstes können hiernach bereits gemessen und stehen anschließend für eine Analyse zur Verfügung.

Nach Abschluss eines Vertrages oder bei vertragsloser Aufrufbarkeit eines Dienstes kann der Dienstaufwurfprozess seitens des Benutzers initiiert werden. Dazu werden im Fall einer interaktiven Nutzung die im Dienstpaket hinterlegten Dienst-Client-Schnittstellen genutzt beziehungsweise Benutzerschnittstellen unter möglicher Einbeziehung ebenfalls hinterlegter UI-Annotationen automatisch generiert. Die Varianten der interaktiven Nutzung von Diensten liegen außerhalb des konzipierten Funktionsumfangs der Basisdienstplattform. Im Fall einer programmatischen Nutzung im Rahmen einer Dienstkomposition wird direkt die Dienstschnittstelle angesprochen.

Schließlich kann der Benutzer während der Vertragslaufzeit oder letztmalig nach dem Vertragsende eine subjektive Bewertung der zugesicherten Eigenschaften und der Gesamtzufriedenheit abgeben.

Autonome Prozesse

Die hier dargestellten Prozesse zur Ausführung, Überwachung und Anpassung von Diensten involvieren zwei Plattformbestandteile als Subarchitektur mit bereits in der Literatur bekannten Problemstellungen. Von besonderem Interesse sind das Zusammenspiel von Vertragsbildung und vertragsgebundener Ausführung einerseits [107] und von Monitoring und Adaption der ausgeführten Dienste andererseits [45].

Dienste werden abhängig von ihrer Implementierungstechnologie direkt auf dem Betriebssystem oder in einem verwaltenden Container ausgeführt. Die Ausführung wird durch einen Kontrollpunkt, etwa einen Proxy, auf das Vorhandensein eines gültigen Vertrags geprüft und kann darüber an weitere Prozesse wie eine hier nicht betrachtete Abrechnung angebunden werden [123].

Die vertragsgebundene Überwachung der Dienstaufwicklung und der dazu

verwendeten Umgebung findet demnach auf der System-, Dienstcontainer-, Dienst- und Dienstinstanzebene statt. Es werden dazu aktiv Messwerte aus Sensoren und Aggregatoren gesammelt und mit den vom Container gelieferten Werten und den instanzspezifischen Werten eines authentifizierenden Proxy kombiniert. Der Zugriff auf Monitoring- wie auch auf von Benutzern abgegebenen Bewertungsdaten ist verzögerungsfrei über Abfragedienste möglich.

In der präsentierten Architektur sind Schnittstellen für die Einbindung einer koordinierten Adaption auf der System-, Dienstcontainer-, Dienst-, Dienstinstanz- und Vertragsebene vorgesehen. Adaptionstrategien wie dynamische Ressourcenallokation, Neubindung und Neuplanung von Prozessen, Rekonfiguration atomarer Dienste [98] sowie eine Neuaushandlung von Verträgen lassen sich damit umsetzen.

Im Sinne einer inkrementellen Qualitätsverbesserung sind kurzfristig wirkende Adaptionstrategien nur begrenzt nützlich. Idealerweise werden die Adaptionmechanismen mit einer dauerhaften Erkennung und Manifestierung der real angebotenen Dienstgüte in den Dienstbeschreibungen und Vertragsvorlagen kombiniert [105]. Dieser bisher in keiner Dienstplattform verfügbare autonom agierende Prozess soll über die Verfahren zur inkrementellen Verbesserung der Metaqualität eingebunden werden, wofür die Basisplattform erweiterbare Schnittstellen enthält. Nach der folgenden Beschreibung der Umsetzung der Komponenten und Prozesse wird deshalb auch eine Beschreibung konkreter Bestandteile der um Metaqualitätsaspekte erweiterten Plattform erfolgen.

4.2 Umsetzung der Plattformdienste und Komponenten

Für die praktische Umsetzung der Verfahren zur Qualitätsverbesserung im Internet der Dienste wird vor allem eine lauffähige Basisinfrastruktur benötigt, mit der der Betrieb eines IoS möglich wird. Die im Rahmen der Arbeit iterativ entwickelte und umgesetzte Dienstplattform mit der Bezeichnung *SPACE* wird dafür als Vertreter ausgewählt. Die Plattform zeichnet sich durch eine hohe Modularität, nutzerzentrische Funktionalität, freie Verfügbarkeit, systemintegrierten Betrieb und die Unterstützung heterogener Dienste im partizipativen Internet der Dienste aus. Der Entwurf folgt den Prinzipien einer durchgängigen Zusicherbarkeit von Dienstgüteeigenschaften im Internet der Dienste [12]. Somit erfüllt die Plattform grundlegend die gestellten Anforderungen.

Die Modularität wird durch die Aufteilung der Funktionalität der Plattform auf einzelne Plattformdienste als Architekturmuster realisiert, wobei eine dienstorientierte Middleware entsteht. Es existieren sowohl rein programmatische Plattformdienste im Sinne von Web Services als auch solche, die aufbauend auf der Web-Service-Schnittstelle eine oder mehrere zumeist webbasierte Benutzerschnittstellen bieten. Darüber hinaus existieren einige Komponenten ohne Dienstschnittstelle, welche als Systemdienste im Hintergrund für zeit- oder ereignisgesteuerte Aktionen zuständig sind. Sowohl Systemdienste als auch rein programmatische Plattformdienste enthalten teilweise jedoch eine minimale Weboberfläche zu administrativen Zwecken.

Die nutzerzentrische Funktionalität wird durch viele semiautomatische Prozesse realisiert, in denen der Benutzer interaktiv Entscheidungen fällen kann. Als

Webanwendungen sind für Dienstanbieter die Dienstsuche *ConQo* [112] und die Anwendung zur Vertragsaushandlung und -verwaltung *Contract Wizard* [100] umgesetzt. Für Dienstnutzer steht zudem die Anwendung *Provider Wizard* zur Bereitstellung und Verwaltung von Diensten zur Verfügung.

Als rein programmatische Dienste und Komponenten existieren eine vereinheitlichte Hostingumgebung für heterogene Dienste (*Unified Hosting Environment*, UHE) [99], ein modular aufgebauter vertragsgesteuerter Monitor mit Sensoren und Aggregatoren mit dem Namen *Grand SLAM* [103], eine dazugehörige Monitoringdatenschnittstelle *Monitoring-as-a-Service* (MaaS) und ein spezielles Monitoringwerkzeug zur Durchsetzung von Systemrichtlinien [95]. Die vertragsgebundene Zugriffssteuerung erfolgt über einen authentifizierenden Proxy *Access Gate*. Mit *NFPcalc* steht exemplarisch die Anbindung eines rekonfigurierenden Adaptionsmechanismus zur Verfügung. Ein *Rating Service* zum Zugriff auf subjektive Bewertungen der Diensteseigenschaften komplettiert die programmatisch nutzbaren Dienste.

Nahezu jeder Plattformdienst ist autark ohne die jeweils anderen nutzbar, sofern die Konfiguration auf diese Situation angepasst ist. Im Sinne der Technologieunabhängigkeit von Web Services sind auch die Plattformdienste in unterschiedlichen Programmiersprachen und Frameworks realisiert. Es entstehen jedoch implizit Abhängigkeiten aufgrund gemeinsam genutzter Datenbankschemata. Auch sind einige der Dienste optional über eine nachrichtenorientierte Middleware (MoM) mit schematisch definierten Nachrichten integrierbar.

4.2.1 Architektur der Plattformdienste

In diesem Abschnitt werden die für diese Arbeit wichtigsten Einzelkomponenten detailliert präsentiert.

Bereitstellung von Diensten über den Provider Wizard

Der Plattformdienst *Provider Wizard* beinhaltet eine programmatische und eine interaktive Schnittstelle für die Bereitstellung und anschließende Verwaltung von Diensten auf der Dienstplattform. Die interaktive Schnittstelle ist als Webanwendung realisiert und in Menüs unterteilt.

Ein Installationsmenü ermöglicht das Hochladen von Dienstpaketen gemäß der Struktur von TSM. Die Dienstpakete werden zu ihrer Installation vom Provider Wizard an die Deployment-Schnittstelle einer Ausführungsumgebung übergeben. Eine solche Schnittstelle wird durch das *Unified Hosting Environment* bereitgestellt. Der Aufruf des Deploymentprozesses erfolgt als SOAP-Nachricht mit einem WS-Attachment-Anhang des Dienstpakets, oder alternativ als lokaler Prozessaufruf. Die interaktive Installation zeigt aufgrund der heuristischen Projektion von Dienstpaketbestandteilen auf TSM-Artefakte ein Bestätigungsmenü an und verläuft somit semiautomatisch. Die programmatische Installation über die Web-Service-Schnittstelle liefert hingegen einen simplen Erfolgsstatus zurück. Fehler in der Projektion müssen anschließend interaktiv korrigiert werden.

In einem Verwaltungsmenü werden dem Anbieter über alle Dienste statistische Informationen zum eigenen Angebot geliefert. Damit kann der Dienstanbieter sein Portfolio verwalten und die Popularität der Dienste einsehen. In diesem Menü werden laufzeitgenerierte Informationen wie abgeschlossene Verträge

und visualisierte Monitoringdaten eingebunden. Es bestehen somit Web-Service-Aufrufe an einen Monitordienst wie *Grand SLAM* und einen SLA-Manager.

Desweiteren dient Provider Wizard zur Aufnahme von anbieterspezifischen Personen- und Organisationsdaten, welche im Fall einer Vertragsbildung als juristisch notwendige Angaben genutzt werden. Sie können auch genutzt werden, um Vertragsvorlagen zu dynamisieren, in denen diese Informationen üblicherweise statisch hinterlegt sind. Die Angaben umfassen abhängig von der Anbieterform eine ladungsfähige Anschrift, Kontaktinformationen, eine Bankverbindung, einen Handelsregistereintrag sowie Steuernummern.

Implementiert ist die Anwendung als Python-Webanwendung unter Nutzung des Template-Frameworks Cheetah für eine Model-View-Controller-Architektur. Der Controller ist an die CGI-Schnittstelle eines Webservers wie Apache httpd angeschlossen und interagiert mit dem Client über HTTP. Die Oberfläche besteht aus internationalisierten HTML-Seiten mit einer austauschbaren Stildefinition als CSS, um Layout und Erscheinungsbild anpassen zu können. HTTP-Cookies werden verwendet, um Benutzerpräferenzen wie die Sprache zu persistieren. Der aktuelle Benutzer wird über HTTP-Authentifizierungsinformationen bestimmt, sofern diese vorliegen. Das Modell wird durch Datenbankeinträge repräsentiert, die abhängig von den HTTP-Anfragen verarbeitet werden. Es werden drei Datenbanken zur Verwaltung der Benutzer, der installierten Dienste und extrahierter Dienstbeschreibungen benötigt. Über die Schnittstelle DB-API 2.0 können nahezu beliebige SQL-Datenbanken dazu verwendet werden.

Die Abbildung 4.4 fasst die Schnittstellen von Provider Wizard grafisch zusammen.

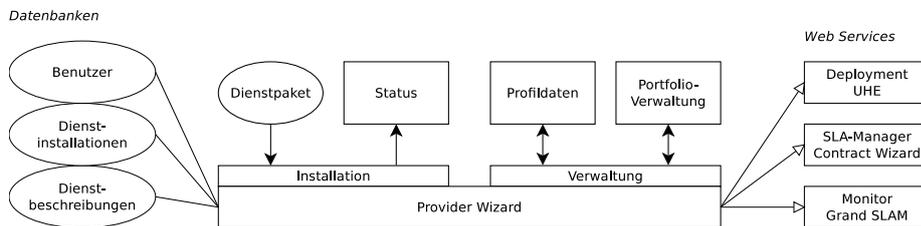


Abbildung 4.4: Schnittstellen von Provider Wizard

Registrierung und Abfrage von Diensten über ConQo

ConQo ermöglicht die Registrierung von und die Suche nach Diensten auf semantischer Basis unter Einbeziehung von nichtfunktionalen Präferenzen in unterschiedlichen Domänen wie Kontext, QoS und Reputation [112].

Jeder Dienst wird primär über eine Beschreibung nach der *Web Service Modelling Language* (WSML) repräsentiert. Die Registrierung erfolgt über Angabe einer Dienstontologie, welche sowohl ihre Domänenklassifikation als auch technische Konzepte von in ConQo bereits enthaltenen, aber ebenfalls zusätzlich registrierbaren Basisontologien ableitet. Jeder Dienst wird dabei mit einem menschenlesbaren Namen und einem eindeutigen IRI-Bezeichner registriert. Optional ist dabei die Angabe eines Logos, eines Dienstes Eigentümers und einer Kurzbeschreibung möglich. Die WSML-Datei kann für den Dienst unterschiedliche Profile, genannt Schnittstellen, anbieten. Dies kann zur Differenzierung von Leis-

tungsmerkmalen genutzt werden. An den Registrierungseintrag können dazu für jede Dienstschnittstelle weitere Dokumente wie etwa WSDL-Dateien, SLA-Vorlagen (SLAT) und GUI-Annotationen angehängt werden. Weitere Einträge sind als Verweise möglich. Damit können die Dienstimplementierung oder herunterladbare Clientanwendungen mit dem Dienst verknüpft werden. Durch den flexiblen Aufbau der Einträge werden die strukturellen Merkmale des Dienstmodells TSM durch ConQo umgesetzt.

Die Angaben zu NFE sind in ConQo dynamisch realisiert und können durch eine angebundene Komponente wie einen Monitor aktualisiert werden. Dabei wird eine weitere, aktualisierte WSML-Datei generiert und bevorzugt gegenüber der originalen, statischen Datei für die Bewertung der Dienstkriterien gegenüber einer Anfrage während der Dienstsuche verwendet.

Die Dienstsuche (Discovery) beginnt aus einer simplifizierten funktionalen Suche durch Auswahl einer Domäne über eine ebenfalls in WSML spezifizierte semantische Zielbeschreibung. Anschließend wird das Suchergebnis durch Berücksichtigung von nichtfunktionalen Anforderungen in der Zielbeschreibung des Benutzers dahingehend reduziert, dass sämtliche die Anforderungen nicht erfüllenden Dienste aus der Ergebnisliste aussortiert und die verbleibenden Dienste nach der Reihenfolge der Erfüllung aller Präferenzen sortiert werden. Das Ergebnis der Discovery ist eine Liste von Registrierungseinträgen.

In der Abbildung 4.5 wird der Funktionsumfang von ConQo als Plattformdienst über dessen Schnittstellen skizziert.

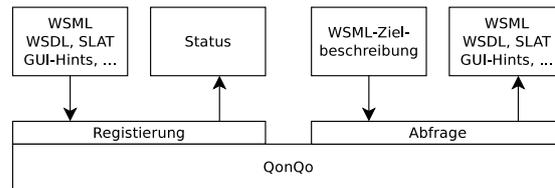


Abbildung 4.5: Schnittstellen von ConQo

ConQo ist als Java-Servlet mit einer SOAP-Schnittstelle (Matchmaker) und JSP-Servlet mit HTML-Oberfläche (ConQoCockpit) implementiert. Es nutzt dabei existierende Bibliotheken zur Verarbeitung der WSML-Dateien wie WSMO4J und WSML2Reasoner sowie als zentrale Komponente die *QoS-Enabled Service Discovery Component*, welche aus dem Forschungsprojekt DIP übernommen wurde¹. Der Matchmaker ist über JDBC an eine SQL-Datenbank angebunden. Die Architektur des ConQo-Matchmakers wird durch die Abbildung 4.6 erläutert.

ConQo enthält bereits eine Menge von Basis- und Domänenontologien, die unter dem Namen *WSMO4IoS* zusammengefasst sind. Die Struktur der Ontologie ist aus der Abbildung 4.7 ersichtlich.

Die Basisontologien definieren nichtfunktionale Qualitäts-, Kontext- und Reputationseigenschaften sowie als Teilontologie *WSAGBase* erweiterte Attribute für Metadaten, die für die Ableitung von Vertragsvorlagen aus den Dienstbeschreibungen notwendig sind. Die Qualitätsmerkmale sind anhand der Nutzung durch QoS-Monitoring in lokal und entfernt überwachbare Eigenschaften unterteilt. Die Basisontologien werden mit einer Domänenontologie kombiniert,

¹Data, Information and Process Integration: <http://dip.semanticweb.org/>

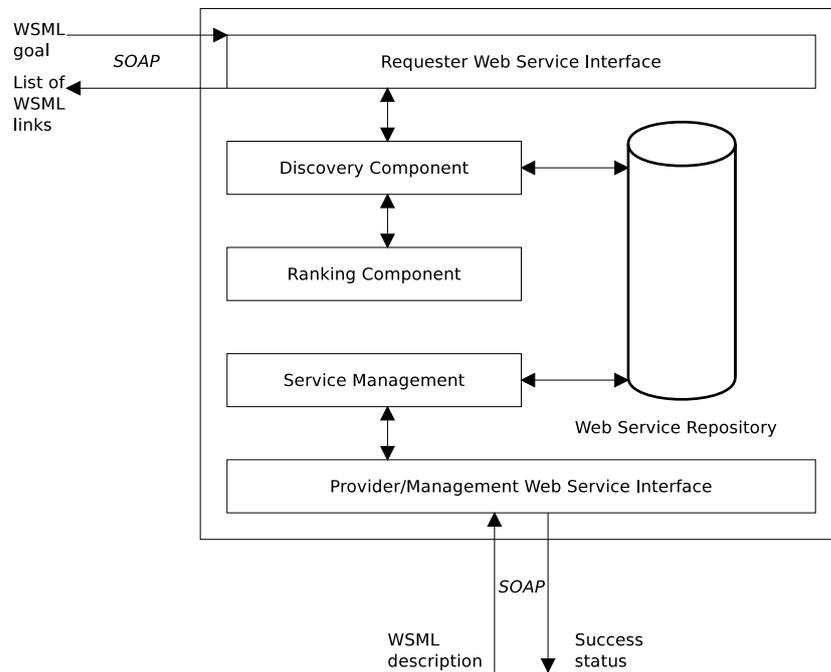


Abbildung 4.6: Kernarchitektur von ConQo

welche die funktionale Domäne des Dienstes repräsentiert. Zur zielgerichteteren Suche nach passenden Diensten lassen sich die Domänen dabei noch in Subdomänen untergliedern. WSMO4IoS enthält Domänendefinitionen für eine Vielzahl von Funktionen wie Hotelreservierungen, Druckdienste und UI-Services. Benötigt ein Dienst andere oder weitere Definitionen, müssen diese als Teil der Dienstbeschreibung im Dienstpaket vorhanden sein.

Vertragsverwaltung mit Contract Wizard

Ist eine vertragliche Bindung an einen Dienstanbieter erforderlich, so kann mit *Contract Wizard* ein Vertrag basierend auf einer Vorlage abgeschlossen, anschließend genutzt und überprüft und schließlich terminiert und bewertet werden [100]. Diese Anwendung besteht aus drei Teilkomponenten. Eine in Ruby imple-

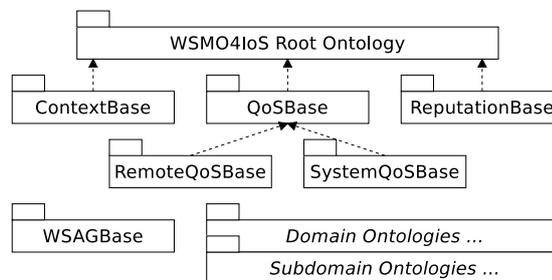


Abbildung 4.7: WSMO4IoS als Basisontologie für ConQo

mentierten CGI-Anwendung zur generellen Ablaufkontrolle beinhaltet auch die konkreten Schritte Vertragsvorlagen- und Tarifauswahl, Auktionsunterstützung, Vertragsabschluss und anschließende Vertragskontrolle. Eine zweite Anwendung auf Basis von Java-Servlets beinhaltet den zentralen Schritt der Vertragsaushandlung auf Basis einstellbarer Anforderungen an die NFE, sowie die Bewertung eines Vertrags nach Beendigung des Vertragsverhältnisses. Schließlich existiert eine ebenfalls als Java-Servlet implementierte Komponente zur Verwaltung von Verträgen als SLA-Dokumente.

Abbildung 4.8 zeigt die Schnittstellen von Contract Wizard.

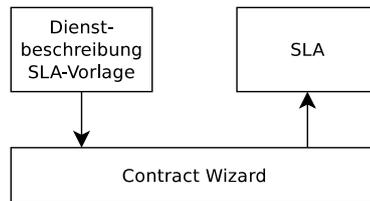


Abbildung 4.8: Schnittstellen von Contract Wizard

Dienstausführung über das Unified Hosting Environment (UHE)

Die eigentliche Dienstausführung obliegt dem UHE, welches speziell für die Verwendung heterogener Diensttechnologien entwickelt wurde [99]. Es zeichnet sich dadurch aus, unterschiedliche Dienstpaketformate gemäß TSM durch eine modulare Verarbeitung und anschließende Weiterreichung an einen spezialisierten Container zu unterstützen. UHE definiert Schnittstellen für die Installation, das Monitoring und die Adaption der Container und der darin enthaltenen Dienste.

Nachdem ein Dienstpaket über Provider Wizard in einem durch UHE verwalteten Dienstcontainer installiert wurde, können anschließend Dienstanfragen direkt an den Container gesendet werden. Diese Anfragen werden unter Umständen durch einen Proxy geleitet, nicht jedoch durch UHE, so dass hierbei kein Skalierbarkeitsproblem entsteht. UHE übernimmt vielmehr die administrative Homogenisierung von Installation, Monitoring und Adaption der Dienste.

Abbildung 4.9 zeigt die Schnittstellen von UHE.

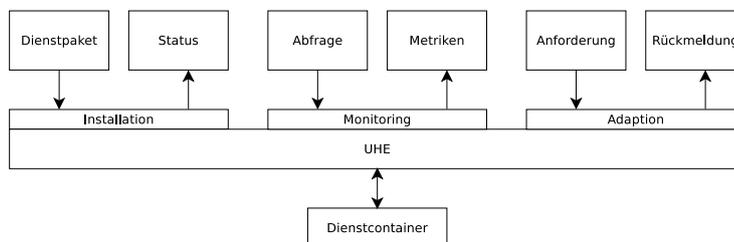


Abbildung 4.9: Schnittstellen der vereinheitlichten Hostingumgebung UHE

Dienstaufwurf und Dienstinstanzmonitoring mit Access Gate

Das Access Gate wird als Proxy vor die Container des UHE geschaltet, wenn Dienstaufrufe überprüft und protokolliert werden sollen. Hierbei wird sowohl eine Authentifikation des Aufrufers als auch eine Autorisierung des Aufrufs durchgeführt. Die Authentifikation erfolgt durch die aus einem WS-Security-Abschnitt in der SOAP-Nachricht ausgelesenen Anmeldeinformationen. Wird eine Autorisierung benötigt, so erfolgt dies durch die Überprüfung des Vorhandenseins eines Vertrags in der SLA-Verwaltung.

Gewonnene Logdaten können sowohl lokal abgelegt als auch zur weiteren Aggregation an den Monitor Grand SLAM übergeben werden. Das Access Gate liefert dabei Metriken auf der Aufruf- bzw. Dienstinstanzebene. Diese umfassen die Antwortzeit des Dienstes, den Durchsatz und den Aufruferfolg.

Abbildung 4.10 zeigt die Schnittstellen des Access Gate.

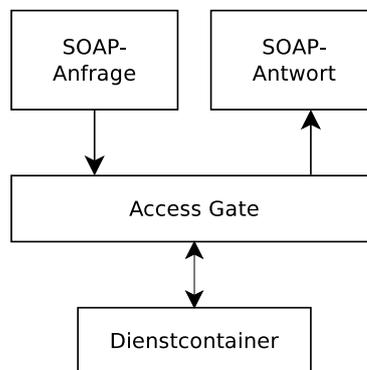


Abbildung 4.10: Schnittstellen des Proxy Access Gate

SLA-bezogenes Monitoring mit Grand SLAM

Durch Aushandlung im Contract Wizard hinterlegte Verträge werden zur kontinuierlichen Überprüfung an den Monitor Grand SLAM übergeben [103]. Dieser übernimmt nicht nur aufrufbezogene diskrete Messgrößen aus dem Access Gate, sondern führt auch eigenständig und aufrufunabhängig diskrete Messungen quasikontinuierlicher Metriken, etwa der Verfügbarkeit eines Dienstes, durch. Sämtliche Messergebnisse werden in einer Datenbank abgelegt.

Die Abbildung 4.11 stellt die Schnittstellen zu Grand SLAM dar.

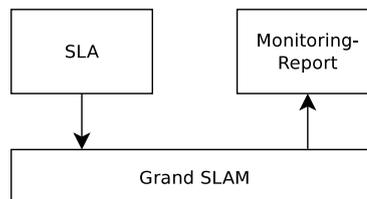


Abbildung 4.11: Schnittstellen von Grand SLAM

Abfrage von Monitoringdaten über Monitoring-as-a-Service (MaaS)

Der Zugriff auf die Monitoringdatenbank von Grand SLAM ist über einen dedizierten Abfragedienst realisiert. Dieser MaaS genannte Plattformdienst integriert Abfrage- und Optimierungsmethoden für einen effizienten und vorhersehbaren Zugriff auf die Monitor Daten.

MaaS besteht aus drei Java-Servlets. Der eigentliche Abfragedienst wird ergänzt um eine interaktiv nutzbare Weboberfläche für administrative Zwecke sowie eine Visualisierungsfunktion, die unter Zugriff auf den Abfragedienst Messdiagramme für die Einbindung in interaktive Anwendungen erzeugt.

Die Abbildung 4.12 stellt die Schnittstellen zu MaaS dar.

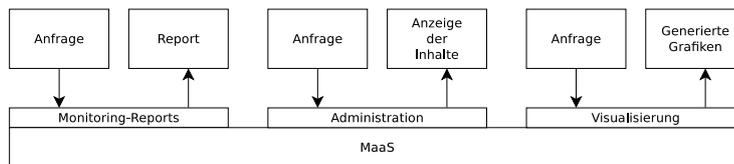


Abbildung 4.12: Schnittstellen von MaaS

Abfrage von Bewertungsdaten über einen Rating-Service

Die interaktive Bewertung von Verträgen und der programmatische Abruf dieser Bewertungen ist über den *Contract Wizard* realisiert. Allerdings sind gerade subjektive Bewertungen personengebundene Daten, die für die meisten Anwendungsfälle in der Form nicht verarbeitet werden können. Desweiteren unterliegen sie Abweichungen aufgrund der individuell unterschiedlich garantierten Eigenschaften in den Dienstnutzungsverträgen. Somit wird eine Aggregationsfunktionalität benötigt, die unterschiedliche SLA-Bedingungen berücksichtigt.

Der Rating Service stellt genau diese Funktionalität zur Verfügung. Er ermöglicht zu jeder Vertragsvorlage und zu jedem Dienst den Abruf der Bewertungen. Dabei werden die numerischen Werte normalisiert und als Durchschnittswert zusammengefasst, während die Textanteile zu einer Liste von Meinungen kombiniert werden.

Die Implementierung stellt einen REST-Dienst auf Basis von Ruby mit einer auf der Bibliothek Sinatra aufbauenden Schnittstelle dar. Die Funktionalität zum Abruf und zur Zusammenführung von Bewertungen ist in einem Modul gekapselt. Bei Eintreffen einer HTTP-Anfrage werden die Einzelbewertungen für den betreffenden Dienst von der Vertragsverwaltung *Contract Wizard* abgefragt. Sowohl die Einzelbewertungen als auch die aggregierten Ratings weisen das im Listing 4.1 enthaltene Format auf.

Die Bewertungen für NFE (`propRating`) und Kategorien (`catRating`) reichen aufsteigend von 1 bis 5. Ist eine bestimmte NFE nicht bewertet, wird ersatzweise die übergeordnete Kategorienbewertung genutzt, die ansonsten nicht zum Einsatz kommt. Die im *Contract Wizard* frei wählbare Kategorisierung kann somit an beliebige NFE-Klassifikationen angepasst werden, ohne eine Auswirkung auf die resultierende Bewertung zu haben.

Zur Reduktion der Arbeitslast und damit der Antwortzeit dieses Plattformdienstes kommt ein Cache zum Einsatz, welcher die Ergebnisse für eine feste Zeitdauer zwischenspeichert. Durch die Konfigurierbarkeit der Vorhaltezeit kann

Listing 4.1: Format der Vertragsbewertungen durch Benutzer

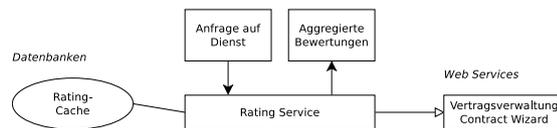
```

1 < ratingtemplateId="..." user="..." time="..." >
2 < categorycatKey="..." catRating="..." >
3 < propertypropName="..." propRating="..." />
4 </category >
5 <opinion >
6 ...
7 </opinion >
8 </rating >
9

```

ein Kompromiss zwischen Aktualität und Ressourcenverbrauch der abgerufenen Daten gefunden werden.

Die Schnittstellen zum Rating-Dienst sind in Abbildung 4.13 zu sehen.

**Abbildung 4.13:** Schnittstellen des Rating-Abfragedienstes

4.2.2 Einbindung der konzipierten Modelle TSM und CPEM

In den Abschnitten 3.2 und 3.5.1 sind die Modelle CPEM zur Repräsentation unterschiedlich ausdrucksstarker NFE im Lebenszyklus der Dienste und TSM zur Repräsentation verteilter heterogener Dienstpakete vorgestellt worden. An dieser Stelle soll die Nutzung der Modelle in den Einzelkomponenten der entwickelten Plattform als Grundlage für die spätere Erweiterung um Metaqualitätsfunktionen zusammenfassend erläutert werden.

Zugriff auf NFE nach CPEM

Der *Service Abstraction Layer* stellt eine vereinheitlichte Sicht auf NFE-Trägerdokumente wie Dienstbeschreibungen dar. Aufgrund der zahlreichen domänen- und anwendungsspezifischen Beschreibungsformate kann sich eine solche Abstraktionsschicht nur auf einige wenige Merkmale konzentrieren. Dies sind hauptsächlich die Beschreibungen der funktionalen und nichtfunktionalen Eigenschaften des Dienstes.

Für die Handhabung heterogener SLAs existieren desweiteren spezielle Abstraktionsschichten, die den Zugriff auf über NFE hinaus gehende Angaben wie Ablaufdaten oder allgemeine Dienstbestimmungen erlauben. Für das Monitoring mit Grand SLAM existiert eine Umsetzung in Form einer Java-Abstraktionsbibliothek namens „slalib“. Durch sie werden SLA-Dokumente auf ein internes Objektmodell abgebildet. Für die Vertragsgestaltung in Contract Wizard existieren einer Menge von XSL-Transformationen. Diese wandeln SLA-Dokumente in eine einheitliche XML-Syntax um.

Paketierung. Die dazu erbrachten grundsätzlichen Betrachtungen der Integrationsmöglichkeiten sowie technische Details zur Umsetzung der Integration können einem technischen Bericht entnommen werden [96].

Es existieren dabei zwei Hauptanwendungsfälle der Systemintegrationsschicht. Für den Aufbau verteilter Dienstinfrastrukturen ist es notwendig, auch kurzfristig neue Ausführungsserver einzubinden oder zentrale Dienste zu migrieren. Für den volatilen Betrieb als Demonstrator ist es hingegen notwendig, die Plattformdienste mit System- und Desktopdiensten zu integrieren.

4.3.2 Plattforminstanzen als Marktplätze

Schließlich existiert auf einer dritten Schicht über den Plattformdiensten und der integrierten Plattform eine Marktplatzintegration mit dedizierter grafischer Gestaltung auf einer einheitlichen Oberfläche und anwendungsübergreifender mehrbenutzerfähiger Authentifizierung. Ein solcher Marktplatz wird im Kapitel zur Validierung vorgestellt werden.

4.3.3 Plattformderivate

Durch die uniformen Schnittstellen sind diverse Erweiterungspunkte geschaffen worden, die prototypisch auch durch Anbindung weiterer Komponenten genutzt werden. Nach dem Baukastenprinzip können daraus spezialisierte Architekturen abgeleitet werden, was auch bereits in Einzelfällen geschehen ist. So ist auch die Erweiterung der Architektur um Bestandteile zur Verarbeitung der Metaqualität als abgeleitete Spezialisierung zu sehen, die im nächsten Abschnitt vorgestellt wird. Die bisherigen Erweiterungen und Integrationsstufen werden in der Abbildung 4.15 in der Übersicht gezeigt und nachfolgend erläutert.

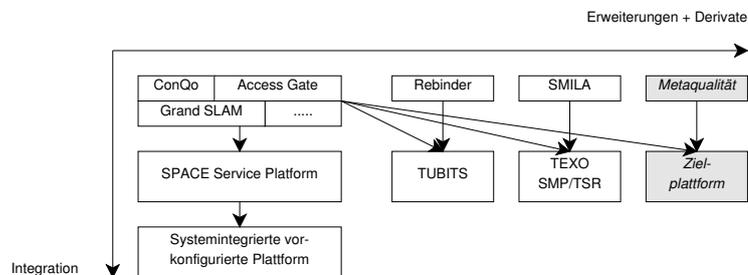


Abbildung 4.15: Abgeleitete und integrierte Systeme auf Basis von SPACE

TEXO-Laufzeitumgebung

Die Erprobung der Dienstplattform als Basistechnologie erfolgt im Rahmen des Forschungsprojekts THESEUS/TEXO³. Ein zentrales Modellmerkmal der TEXO-Architektur ist die alleinige Verwendung von USDL als Dienstbeschreibungssprache und einer dahinter liegenden pyramidal aufgebauten Dienstontologie zur Repräsentation von Wissen als juristische, geschäftliche und branchenspezifische Angaben [72]. Der Gesamtansatz von TEXO ist in der Abbildung 4.16

³Webseite des THESEUS-Programms: <http://www.theseus-programm.de/>

dargestellt. Die zentrale Laufzeitumgebung der TEXO-Plattform als Anpassung und Erweiterung von SPACE ist mit allen weiteren Architekturbestandteilen wie Dienstentwicklung und Dienstauslieferungsframework verknüpft.

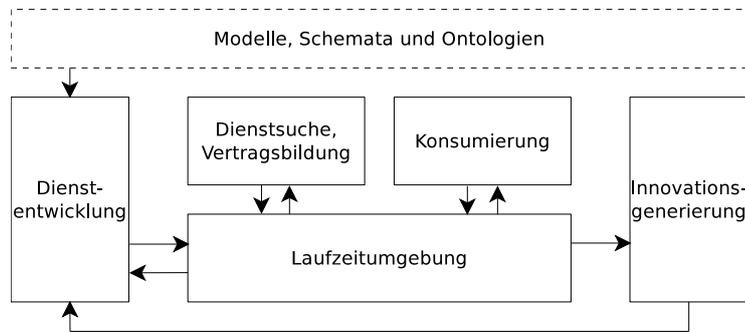


Abbildung 4.16: Gesamtansatz von TEXO mit Entwurfs- und Laufzeitkomponenten

- Die Entwicklungsumgebung *ISE Workbench* [91] dient zur modellgetriebenen Erstellung neuer Dienste. Sie besteht aus einer Vielzahl an Editoren und Transformatoren, welche sich in ihrer Anordnung an der ISE-Methodologie, angelehnt an dem Zachmann-Framework, orientieren [52]. Erstellte Dienste können direkt aus dem Editor auf die Laufzeitumgebung gebracht werden.
- Die Suche nach passenden Diensten ist in der *TEXO-Discovery* auf Basis des semantischen Suchintegrationsframeworks *SMILA* realisiert, welches als BPEL-Prozess die einzelnen Suchmechanismen orchestriert und deren Ergebnisse zusammenführt⁴. Eine Suche ist nach Volltext, Schlüsselwörtern und semantischen Konzepten möglich.
- Webanwendungen, mobile Geräte oder Unternehmensanwendungen interagieren mit den Diensten. Ein Codesign von Dienst und Client fördert dabei das Zusammenspiel.
- Auf Basis von Dienstbewertungen, häufigen Suchbegriffen und anderen Feedback-Informationen aus der Laufzeit lassen sich Ideen für innovative Dienstleistungen bilden und zu neuen konkreten Diensten entwickeln [87].

Die Abläufe auf der TEXO-Plattform werden durch ein Governance-Framework kontrolliert [48]. Dabei werden insbesondere die Anforderungen an den Lebenszyklus und die Granularität von Diensten sowie deren Interaktion mit Benutzern betrachtet [71].

Die TEXO-Laufzeitumgebung gliedert sich in einen zentralen Marktplatz (*Service Management Platform, SMP*) und dezentrale Ausführungsserver (*Tradable Services Runtime, TSR*). In TEXO werden Komponenten zur Dienstherstellung, -suche, -nutzung und zur Innovationsgewinnung teils ergänzend und teils ersetzend auf der Basisplattform aufgesetzt.

⁴SMILA-Projektseite: <http://www.eclipse.org/smila/>

Die Laufzeitumgebung von TEXO besteht aus zahlreichen Komponenten über SPACE-Plattformdienste hinaus, um eine Plattform mit höherer Funktionsvielfalt bilden zu können. Die Integrationspunkte sollen an der Stelle skizziert werden, um die Erweiterbarkeit der Plattform zu belegen.

- Ergebnisse des Rating-Abfragedienstes lassen sich in die TEXO-Komponente Semantic Backend hinterlegen. Dazu werden die XML-Daten über ein Skript `rating2onto` in RDF/XML-Instanzen des Rating-Moduls der TEXO-Dienstontologiepyramide gewandelt. Auf dieser Basis können Innovations- und Empfehlungswerkzeuge aus dem Projekt Vorschläge für die zukünftige Dienstnutzung erstellen.
- Die Vertragsverwaltung Contract Wizard ist neben der Nutzung des enthaltenen SLA-Managers auch die Nutzung des TEXO-SLA-Managers möglich [124]. Anstelle der MaaS-Visualisierung ist auch eine Nutzung des *SemaVis*-Frameworks für eine Visualisierung der Monitoringdaten zu jedem Vertrag über das Flex-Toolkit möglich [9]. Schließlich ist eine Plattform für kombinatorische Reversauktionen eingebunden, an die der Aushandelnde Beschreibungen für Dienstbündel zum Zweck der Preisreduktion übermitteln kann.
- Im Anbieterwerkzeug Provider Wizard kann alternativ zum Deploymentdienst des Unified Hosting Environments der TEXO-Deploymentdienst aufgerufen werden. Dieser Dienst installiert weitere Dienstartefakte in Datenbanken von Plattformdiensten und ruft anschließend den UHE-Deploymentdienst auf. Der TEXO-Deploymentdienst wird auch für das Deployment aus der ISE-Workbench heraus genutzt.
- Eine Performance-Prediction-Komponente liefert Laufzeitschätzungen für installierte BPEL-Dienste [44]. Die Schätzungen werden genutzt, um Vorschläge für Änderungen an SLA-Vorlagen zu generieren.
- Authentifizierte Aufrufe im Access Gate werden an den TEXO-SLA-Manager gebunden, um eine Autorisierung anhand der Existenz eines gültigen Vertrags durchzuführen.
- Messungen des SLA-Monitors Grand SLAM werden über eine nachrichtenorientierte Middleware an einen *SLA Violation Analyser* übertragen. Diese Komponente hat weitreichendere Möglichkeiten der Feststellung einer SLA-Verletzung auf Basis von horizontalen und vertikalen Abhängigkeiten zwischen Verträgen [65].

TUBITS - Basisinfrastruktur für handelbare Dienste

Eine weitere Nutzung als flexible Basisplattform erfolgt im Rahmen der *TU Dresden Base Infrastructure for Tradeable Services* (TUBITS) mit teilweise aus TEXO übernommenen und teilweise eigens entwickelten Bestandteilen. Die Dienstplattform SPACE wird für TUBITS um mehrere Komponenten der Entwicklungs- und Laufzeit erweitert:

- Editoren ermöglichen die Suche nach Diensten mit einem *Discovery Wizard* und die explizite Festlegung alternativer Dienste, um eine zeitintensive Alternativensuche zur Laufzeit auf Basis funktionaler Äquivalenz oder Ähnlichkeit vermeiden zu können.

- Benutzerbewertungen werden über ein Reputationssystem (*WebRat*) ausgewertet und fließen in die Dienstsuche in ConQo mit ein. Dazu sind erweiterte Basisontologien geschaffen worden, um solche Rückmeldungen abbilden zu können.
- Die Adaptionsmechanismen werden um eine Komponente für das erneute Binden von Diensten in BPEL-Prozessen (*Rebinder*) erweitert. Dadurch wird die Wahrscheinlichkeit von SLA-Verletzungen gesenkt [114].
- Die Benutzung von Diensten über zugeordnete und vertraglich zubuchbare Benutzerschnittstellen ist in TUBITS realisiert. Eine dynamische Dienstnutzung ist bei Vorhandensein einer syntaktischen Dienstbeschreibung wie WSDL oder WADL stets über die ad-hoc-Generierung von Formularen durch die Anwendung *Dynvoker* auch ohne spezielle Clientsoftware möglich [102]. Weitere Clients sind entsprechend der Vertragsbedingungen herunterladbar.

Im Gegensatz zu SPACE stellt TUBITS keine allgemein verfügbare Basisplattform dar, sondern dient der Evaluierung von Verfahren zur Qualitätssicherung durch Ausschluss fehlerhafter und nicht vertrauenswürdiger Dienste. Im Gegensatz zur TEXO-Laufzeitumgebung sind weniger Funktionen enthalten, dafür ist ein höherer Integrationsgrad erreicht.

4.4 Umsetzung und Integration der SMQ-Erweiterungen

Im Unterabschnitt 3.5.3 sind auf abstrakter Ebene SQM-Komponenten zur qualitätssteigernden Erweiterung von IoS-Plattformen vorgestellt worden. Neben der eigentlichen Dienstplattform liegen auch die konzipierten Erweiterungen zur Gewährleistung der Dienstqualität als implementierte Plattformdienste und Bibliotheken vor. Sie sollen an dieser Stelle erläutert werden.

4.4.1 Umsetzung der Dienstabstraktionsschicht SAL

Die Dienstabstraktionsschicht SAL wurde als Bibliothek für die Programmiersprache Python implementiert. Das Hauptmodul enthält alle benötigten Klassen zur Strukturdefinition sowie Verwaltungsfähigkeiten. Der Auffassung von heterogenen, paketierte, selbstbeschriebenen und handelbaren Diensten nach dem TSM folgend repräsentieren die Klassen die Modellbestandteile für Pakete, Dienste, Operationen, Beschreibungen und Eigenschaften. Ein Dispatcher und ein Cache sind verfügbar, um den Zugriff auf Dienstbeschreibungen und durch sie referenzierte Ressourcen transparent und effizient zu ermöglichen. Somit können sowohl lokale als auch entfernt befindliche Dateien, die über HTTP oder andere Protokolle abgerufen werden, analysiert werden.

Für jedes Format einer Dienstbeschreibung oder einer SLA-Vorlage existiert eine Parserklasse als sogenannter Format-Handler. Diese Handler können zu beliebigen Zeitpunkten installiert werden und werden anschließend automatisch genutzt, falls ihre Angabe zu den durch sie verarbeitbaren Formaten auf das aktuelle Dokument zutrifft. Derzeitig existieren acht derartige Handler:

1. Einfache Schlüssel-Wert-Paare für NFE als Dienstbeschreibung. Eine solche Syntax kommt zwar in keiner praktisch verwendeten Dienstbeschreibung vor. Sie ist allerdings einfach zu generieren und auszulesen und stellt somit ein ideales Format für Evaluierungszwecke dar.
2. Resource Description Framework (RDF) als semantische Beschreibungssprache. Aus diesen Dateien werden vor allem Dublin-Core-Metadaten extrahiert.
3. UISDL für Beschreibung für UI-Dienste [43]. Aus diesen Dateien werden die im UISDL-Format festgelegten Eigenschaften wie Lizenz, Preis und Zielplattform der Dienste ausgelesen.
4. WSDL als klassische Beschreibung der Syntax von Web-Service-Operationen. Während die semantische Annotation von WSDL-Strukturen über die Erweiterung SAWSDL möglich ist, beschränkt sich die SAL-Implementierung auf die Extraktion weniger technischer Elemente wie dem Namen, dem Endpunkt und des Transportprotokolls.
5. WSML für semantische Dienstbeschreibungen und NFE-Angaben. Da WSML konkrete Angaben zu NFE kennt, werden diese direkt ausgelesen.
6. WS-Agreement mit Erweiterungen als Vertragsvorlagenformat. In diesem Format werden alle garantierten NFE als *Service Level Objectives* (SLOs) erkannt.
7. Monitoring-Reports der Komponente Access Gate. Die Angaben korrespondieren zu den vertraglich garantierten NFE.
8. Ergebnisse der Performance-Prediction-Komponente. Diese Beschreibungen enthalten stets eine einzige NFE, nämlich die zu erwartende Performanz eines Dienstes in Sekunden.

Alle relevanten Diensteigenschaften werden aus den Beschreibungsdateien ausgelesen und in den instanziierten Modellklassen als Attribute repräsentiert. Dabei werden insbesondere die Expressivitätsstufen nach CPEM betrachtet. Nicht vorhandene Angaben (e_0) werden durch den Nullwert `None` repräsentiert. Einzelwerte (e_1) nutzen die vordefinierten Datentypen `string`, `float`, `integer` und `bool`. Für Wertebereiche existiert die Klasse `ValueRange` zur Aufnahme statistischer Merkmale.

Derzeitig ist nur der rein lesende Zugriff auf Dienstbeschreibungen über den SAL-Parser möglich. Zwar können die Objekte im Hauptspeicher verändert werden, eine Persistierung mit Unterstützung für die gleiche Varietät an heterogenen Formaten für Dienstbeschreibungen und SLA-Vorlagen würde jedoch weitreichende Erweiterungen benötigen. Diese sind im Rahmen dieser Arbeit nicht vorgesehen.

4.4.2 Umsetzung des Metadatenkorrelationsdienstes MD-CS

Der Metadatenkorrelationsdienst MDCS wurde in Python implementiert. Er beinhaltet für einen REST-Zugriff auf seine Funktionalität das Dienstentwicklungsframework `web.py`. Die an die SOAP-Schnittstellen der Dienstregistry sowie

4.4. UMSETZUNG UND INTEGRATION DER SMQ-ERWEITERUNGEN 111

der Monitoring-Abfrageschnittstelle MaaS durchgereichten Anfragen sind basierend auf der Python-Bibliothek *suds* implementiert. Anfragen an die REST-Schnittstelle der Dienstbewertungsabfrageschnittstelle wurden ohne externe Bibliotheken realisiert. Schließlich sind die Anfragen an die Abfrageschnittstelle der SLA-Verwaltung abhängig davon, ob diese für einen dedizierten SLA-Manager mit SOAP-Schnittstelle oder die einfache Abfrage aus der Datenbank von Contract Wizard konfiguriert worden sind. Derzeitig ist nur letztere Möglichkeit umfassend implementiert und getestet.

Die Architektur des MDCS mitsamt der enthaltenen verfahrensgebundenen Module und der angesprochenen Datenquellen für den Abruf von NFE-Trägersdokumenten wird in Abbildung 4.17 gezeigt.

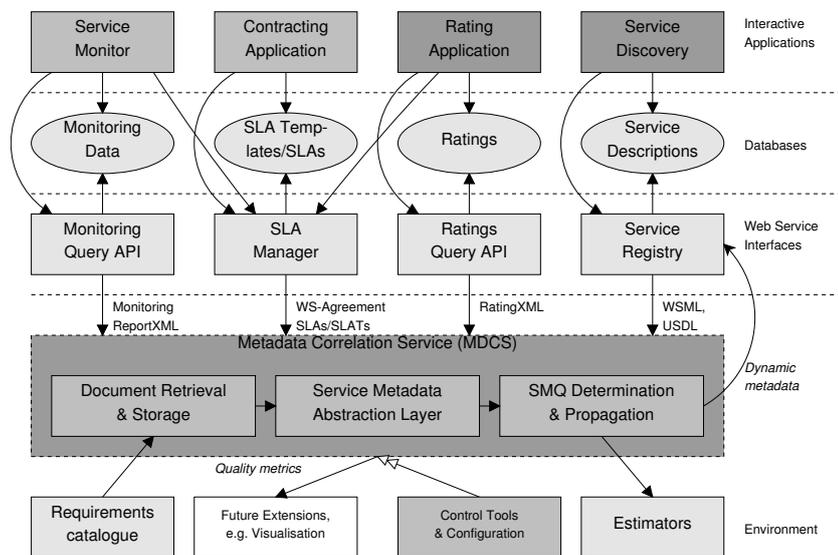


Abbildung 4.17: Architektur des Metadaten-Korrelationsdienstes

Eine beispielhafte Abfrage des MDCS zum Abruf und zur Vereinigung von statischen und dynamischen Informationen eines Dienstes ist nach dem folgenden Schema möglich:

```
1 http://localhost:8080/quality/EcoCalculator
```

Diese Abfrage weist den MDCS an, die Dienstbeschreibung und SLA-Vorlagen für den Dienst aus der Registry zu laden und sie zwecks einer statischen Qualitätsanalyse in SAL zu verarbeiten. Anschließend werden die korrespondierenden Dienst- und Vertragsbewertungen sowie Monitoringergebnisse abgerufen und zu den Beschreibungen korreliert. Für jede im SLA-Dokument gefundene NFE wird eine Korrelationsmetrik bestimmt. Die Berechnung dieser Metrik hängt von der Expressivität der NFE ab. Für Einzelwerte und einfach begrenzten Werten in den SLAs wird die Distanz als mittlerer quadratischer Fehler bestimmt. Schließlich wird daraus das folgende beispielhafte Resultat auf der höchsten Aggregatensebene generiert:

```
1 <quality>0.77</quality>
```

Dieser Wert repräsentiert eine Metrik für die Gesamtbeschreibungsqualität des Dienstes. Sein Wertebereich liegt zwischen 0 und 1. Je niedriger der Wert wird, desto mehr muss der Dienstentwickler dafür sorgen, dass die Abdeckung und Korrektheit der NFE durch zusätzliche Beschreibungsfacetten und möglicherweise Re-Engineering des Dienstes verbessert wird. Werte nahe 1 markieren einen akkurat und vollständig beschriebenen Dienst, der somit ein wahrscheinlicher Kandidat für automatische Dienstauswahlverfahren sein wird.

4.4.3 Gesamtsicht auf die erweiterte Plattform

Die um Metaqualitätskomponenten erweiterte Dienstplattform *SPACE* ermöglicht den qualitativ hochwertigen Betrieb eines Internets der Dienste. Abschließend soll dazu eine Architektursicht zur entwickelten Gesamtlösung präsentiert werden.

Die Gesamtarchitektur ergibt sich nicht nur aus einer bloßen Zusammenstellung von *SPACE* und *MDCS*, sondern auch durch eine vertiefte Integration mit zusätzlichen Verbindungen zwischen den Komponenten. Die Anzeige der angebotenen Dienste im Provider Wizard wird beispielsweise um eine Nennung der Metaqualität ergänzt. Eine solche Integrationstiefe ist selbst mit SOA-Techniken nicht automatisiert realisierbar. Es sind dazu folglich geringfügige Erweiterungen an der Dienstplattform nötig.

Die vollständige Architektur unter Berücksichtigung aller Akteure (Parallelogramme), Plattformdienste (weiße Ellipsen), Datenbanken (graue Ellipsen) und Dienstaufrufe (Pfeile) ist im Diagramm 4.18 abgebildet.

4.4. UMSETZUNG UND INTEGRATION DER SMQ-ERWEITERUNGEN113

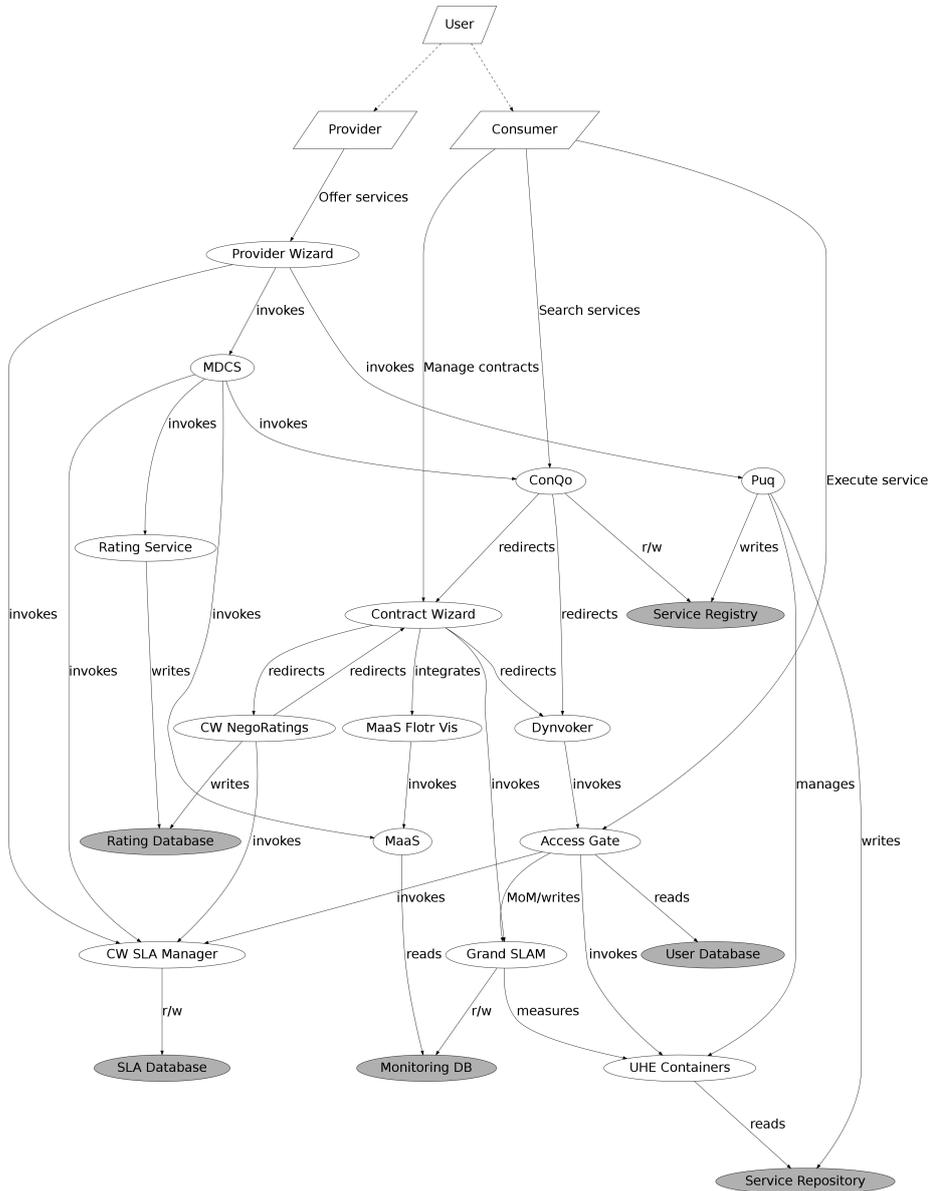


Abbildung 4.18: Gesamtarchitektur von SPACE mit Erweiterungen zur Metaqualität

Kapitel 5

Validierung von Methodik und Plattform

Die in den beiden vorherigen Kapiteln erarbeiteten Modelle, Konzepte und Plattformbestandteile sollen durch eine Validierung geprüft werden. Dabei muss zuerst die Auswahl eines geeigneten Verfahrens getroffen werden.

Eine formale Validierung auf Basis verifizierter Modelle wird aufgrund der hohen Dynamik aller Entwicklungen im Internet der Dienste als derzeit nicht praktikabel eingeschätzt. Prozess-Formalisierungsgrundlagen wie Bindungsgraphen in BPEL [15] und Choreographiemodelle im π -Kalkül [117] sind stets mit Einschränkungen verbunden, zumeist mit einer Restriktion der Syntax. Dies widerspricht dem heterogenen Charakter der auf der konzipierten Dienstplattform verwalteten Dienste. Zudem wird in dieser Arbeit ein holistischer Ansatz für eine verteilte Dienstplattform gegenüber den meist thematisch isoliert durchgeführten Arbeiten in Bereichen wie Dienstsuche und -bewertung, ad-hoc-Komposition oder SLA-Aushandlung verfolgt, wodurch sich die Komplexität der Modelle vervielfacht und die Fehlerfreiheit in den Modellen nicht mehr sichergestellt werden kann [2].

Eine rein simulative Validierung kann anhand generierter Beispieldatensätze vorgenommen werden. Es entstünden jedoch mit zunehmender Berücksichtigung möglicher Einflussfaktoren der Dienstkontraktierung und -ausführung zu viele unbekannte Variablen, so dass diese Form der Validierung für den vorliegenden Anwendungsfall ebenfalls als nicht praktikabel eingeschätzt wird. Desweiteren würde sich die Validierung auf die Korrektheit beschränken, während die Prüfung der Skalierbarkeit der entwickelten Dienstplattform unberücksichtigt bliebe.

Die experimentelle Validierung der Qualitätsverbesserungskonzepte an einem Realsystem bietet sich als Validierungsmethode an. Sie benötigt allerdings über die konzipierten und implementierten Plattformbestandteile hinaus weitere Vorbedingungen und Implementierungsarbeiten. Aufgrund einer erwarteten wirkungsvollen Überführung der Konzepte in die Praxis sind solche Arbeiten aber ohnehin nicht auszusparen. Aus dem Grund wird die experimentelle Validierung mit aktiven Benutzern auf einem Realsystem als Verfahren ausgewählt. Die dabei zu erwartenden Aktivitätsdefizite sollen über eine hybride Simulation der Vorgänge auf einer Replika des Realsystems ausgeglichen werden.

Einleitend werden daher Ziele und Anforderungen für die experimentelle Validierung vorgestellt und ein Komplexitätsmodell auf Basis mathematischer Matrizennotation erarbeitet. Anschließend wird die Erweiterung der Dienstplattform SPACE um eine für die Validierung im Feldversuch günstige einheitliche Oberfläche und ein Agentensystem sowie die Einbindung der erweiterten Plattform in eine Experimentierumgebung präsentiert. Schließlich erfolgt nach dieser Vorbereitung des Experiments die Durchführung auf Basis aufgezeichneter und durch Simulation verstärkter Benutzeraktivität und die Auswertung der Ergebnisse hinsichtlich der Skalierbarkeit und Korrektheit der Konzepte und der erweiterten Plattform.

5.1 Anforderungen an die Validierung

Die praktische Validierung der entwickelten Dienstplattform wird anhand einer beispielhaften Installation („Instanz“) durchgeführt. Diese weist jeweils eine Mindestanzahl für alle skalierbaren unabhängigen Parameter wie Benutzer und Dienste auf und kann somit aussagekräftige Daten über den Dienstbetrieb und die Dienstverwaltung liefern, sofern sich daraus auch eine Mindestaktivität an nutzer- und anbietertypischen Prozessen wie SLA-Aushandlungen oder Deployments als abhängige Parametergröße ergibt. Neben der Skalierbarkeit der Lösung soll mit Hilfe eines anhand typischer Merkmale zusammengestellten Szenarios auch die Korrektheit überprüft werden. Diese Überprüfung basiert auf Stabilitäts- und Konvergenzbetrachtungen der Metaqualität über die Zeit.

5.1.1 Skalierungsbetrachtungen

Zur Erreichung der Mindestanzahl an Benutzern, Diensten und Dienstbeschreibungen sowie Ausführungsplattformen sind jeweils spezielle Methoden anzuwenden, die nachfolgend erläutert werden. Mangels vergleichbarer Validierungsansätze wird als Zielgrößenordnung angestrebt, die Experimente auf 10 Ausführungsservern parallel zu betreiben, 100 Benutzer zu beteiligen, 1000 auf den Servern installierte Dienste anzubieten sowie 10000 registrierte entfernt befindliche Dienste zur statischen Prüfung hinzuzuziehen. Das Dienstangebot wird dabei aus speziell konzipierten Szenariodiensten und automatisch generierten Dienstschnittstellen für Anwendungsprogramme zusammengesetzt. Das Radar diagramm 5.1 visualisiert die Zielstellung bezüglich aller skalierbaren Basisparameter für die Validierung. Die Zielgrößen sind dabei nicht als konkrete Zahlen, sondern als realistische und bezogen auf einen Realbetrieb aussagekräftige Größenordnung zu verstehen.

Mindestanzahl an Diensten

Für die Erreichung einer möglichst hohen Zahl an Diensten wird in der Anfangsphase des Betriebs nicht mit einer entsprechenden Angebotsaktivität der Benutzer zu rechnen sein. Stattdessen werden Dienste in hoher Zahl mit einer möglichst einfachen Aufrufbarkeit vorinstalliert sein müssen. Dabei wird sowohl die Angebotstiefe durch speziell vorkonfigurierte Dienste mit besonderen Eigenschaftenmerkmalen als auch die Angebotsbreite durch eine Vielzahl einfacher Dienste abgedeckt werden.

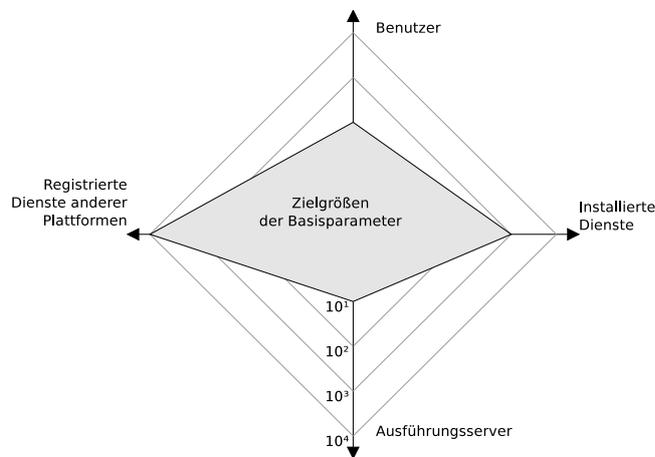


Abbildung 5.1: Zielstellung: Skalierbare Basisparameter

Als umfänglich aufbereitete Szenariodienste wurden ein *Fortune-Service* und ein *Sudoku-Service* mit einer SOAP-Schnittstelle realisiert. Die Funktion des Fortune-Dienst besteht aus einer parameterlosen Operation, die jeweils ein historisches Zitat zurückliefert. Der Sudoku-Dienst nimmt eine ASCII-Repräsentation eines beliebig vollständig gelösten Sudoku-Spielbretts entgegen und beantwortet die Anfrage mit einer vollständigen Lösung. Zu beiden Diensten sind jeweils Clients für verschiedene Ablaufumgebungen realisiert worden, um die Benutzerattraktivität zu steigern. Angeboten werden Clients für den Desktop, für die Kommandozeile und für den Webbrowser. Die Clients lassen eine Konfiguration des Endpunktes zu, um die Dienste vertragsgebunden über den Proxy Access Gate aufrufen zu können. Die Dienstpakete enthalten Beschreibungsdokumente in den Formaten WSDL, WSML und USDL sowie Vertragsvorlagen nach WS-Agreement. Die Implementierung besteht aus Dienstmodulen für den *Ruby Services Container* (Rusco) unter Nutzung der Bibliothek SOAP4R für die synchrone Kommunikation.

Exemplarische Implementierungen von Diensten auf Basis heterogener Frameworks bilden eine weitere Quelle. Diese Dienste dienen nicht zuletzt auch der automatisierbaren Überprüfung der Funktionstüchtigkeit aller Dienstcontainer im Rahmen des Monitoring. Hingegen sind die Beschreibungen dieser Dienste meist auf eine WSDL-Datei beschränkt. Als Technologien kommen hierbei beispielsweise BPEL-Prozesse zum Einsatz, die durch den Container *Apache Orchestration Director Engine* (ODE) ausgeführt werden.

Für die Gewinnung des Hauptteils der Dienste wurde ein Verfahren zu Serienfertigung hunderter Dienste aus existierenden Kommandozeilenprogrammen entwickelt. Dabei werden sowohl auf Ebene der Quellcodes die Verwendung von Kommandozeilenparsern als auch verfügbare Dokumentationen im Manpage-Format analysiert. Die Ergebnisse werden anschließend in ein deklaratives Zwischenformat überführt, welches durch den generischen SOAP-Wrapperdienst SOAPlab [93] in eine WSDL-Datei gewandelt wird. Somit werden die Kommandozeilenprogramme als Dienste von SOAPlab verwaltet und zugänglich gemacht. Da SOAPlab nur eine rudimentäre Registry bietet, werden parallel zur Generierung des Zwischenformats auch WSML-Dienstbeschreibungsvorlagen au-

Tabelle 5.1: Für die Evaluierung zusammengestellte Dienste

Dienste und Quelle	Anzahl	Besonderheiten
MPWS	544	Kommandozeilendienste via SOAPlab
Flug- und Reisebuchungen aus Praktika	51	Java-Servlets, BPEL-Prozesse
Videokonvertierung aus Praktika	20	Java-Servlets mit Axis2
Szenariodienste	2	Ruby-SOAPlets
Framework-Vergleichsdienste	8	BPEL-Prozesse, Python-Dienstmodule
Individuelle Angebote	6	Axis2-Module

tomatisiert ausgefüllt und zusammen mit einem Verweis auf die WSDL-Datei aus SOAPlab in der Registry ConQo hinterlegt. Dabei kommt eine angepasste Domänenontologie zum Einsatz, um Güteparameter des Wrapping-Prozesses abbilden zu können. Dadurch wirkt sich die Beschreibungsqualität der erkannten Kommandozeilenparameter direkt auf die Bewertung der Dienste in einer Dienstsuche aus. Die entstandene Lösung *Mass Production of Web Services* (MPWS) [58] liefert derzeit über 500 generisch aufrufbare Dienste. Diese Zahl kann durch weitere semiautomatische Analyseläufe erhöht werden.

Diese Menge an Diensten wird ergänzt um verfügbare Dienste auf Java-Basis, die im Rahmen studentischer Praktika entwickelt worden sind. Zusammen sind über 50 Dienste in den Domänen Flugbuchung, Einschreibung in Lehrveranstaltungen einer Universität, Wetterdienstabfragen und Videokonvertierung aus diesen Quellen nutzbar.

Als ergänzende Quelle dienen schließlich Dienstangebote, die im Verlauf des Experiments durch Dritte über das Dienstportal angeboten worden sind. Diese Möglichkeit wurde eingeführt, um das Experiment unter möglichst realistischen Bedingungen ablaufen zu lassen und die Aktivität auf dem Portal weiter zu steigern. Hervorzuheben sind hierbei vor allem die im Bereich *Internet Information Retrieval* produktiv genutzten Dienste des WebKnox-Portals ¹.

Die Dienste, ihre Herkunft und Besonderheiten sind in der Tabelle 5.1 zusammengefasst.

Alle Dienste sind, sofern sie nicht bereits mit einer semantischen Beschreibung aufwarten konnten, mit einer automatisch generierten WSML-Datei ausgestattet worden, welche nur generische NFE für nichtlokale Dienste wie Verfügbarkeit und Antwortzeit enthält. Über ein Transformationsverfahren sind daraus anschließend, wiederum nur in den notwendigen Fällen, SLA-Vorlagen im Format WS-Agreement+ValueTypes erzeugt und registriert worden. Somit weisen alle installierten Dienste mehrere NFE-Trägerdokumente auf. Dies ist der Vollständigkeit der Beschreibungen und somit der Metaqualität des Dienstangebots zuträglich. Andererseits wird durch die automatische Generierung und Anpassung von Dokumenten die Zuordnung der Metaqualität zu einem bestimmten Dienstanbieter gegenüber der Zuordnung zum Dienstportal wie bereits vermutet erschwert.

¹WebKnox-Informationportal: <http://www.webknox.com/>

Tabelle 5.2: Dienstbeschreibungen aus öffentlich zugänglichen Verzeichnissen

Quelle	Anzahl	Besonderheiten
Service-Finder	12248 (32110)	EU-Forschungsprojekt zur Dienstsuche in Web-2.0-Umgebungen
BioCatalogue	1027 (1071)	Produktivdienste der Biowissenschaften
Service-Repository	25 (37)	Testdienste
OPOSSum	1173 (1261)	Heterogene Beschreibungen aus den Geowissenschaften
XMethods	342 (365)	Testdienste
EBI	22 (298)	SOAPlab-Dienste der Biowissenschaften, mehrheitlich bereits in BioCatalogue enthalten

Mindestanzahl an Dienstbeschreibungen

Die Performance der statischen Qualitätsanalyse kann mit einer größeren Datenmenge als nur den Dienstbeschreibungen der installierten Dienste gemessen werden. Dazu werden weitere Dienstbeschreibungen aus öffentlich zugänglichen Dienstverzeichnissen in den Verzeichnisdienst ConQo importiert und anschließend auf die Vollständigkeit der NFE-Angaben überprüft.

Die Tabelle 5.2 zeigt die derzeitigen bekannten globalen Dienstverzeichnisse, die als Quelle für Dienstbeschreibungen in unterschiedlichen Formaten dienen können. Einige der Quellen stellen Abfrageschnittstellen bereit, über die neben den syntaktischen Dienstauftragsbeschreibungen auch Metadaten zu jedem Dienst abgerufen werden können. Die angegebene Anzahl an Diensten entspricht der Zahl der erfolgreich in ConQo importierten Beschreibungsdokumente bezogen auf die Gesamtzahl an extrahierten WSDL-Dateien.

Für den Import der Dienstbeschreibungen in die Registry und deren Anreicherung mit Metadaten ist der *Pica-Pica Service Description Crawler* entwickelt worden². Dieses Werkzeug enthält Zugriffsroutinen für alle betrachteten öffentlichen Dienstverzeichnisse sowie einfache Validierungs- und Duplikaterkennungsfunktionen. Da sich die inkrementelle Ergänzung der NFE-Trägerdokumente im Konzept dieser Arbeit auf nichtfunktionale Eigenschaften beschränkt, im Fall importierter Dienstbeschreibungen aber häufig die weiteren Dokumente wie SLA-Vorlagen fehlen, wurde Pica-Pica mit den Generatoren für WSML und darauf basierend WS-Agreement gekoppelt, die bereits in der Darstellung der Szenariodienste erwähnt worden sind. Die Architektur und der Ablauf des Crawlers sind in der Abbildung 5.2 dargestellt.

Mindestanzahl an Dienstbenutzern

Die hohe Zahl an Benutzern wird durch die visuelle Integration der Plattformdienste in ein Webportal mit Fokus auf soziale Netzwerke und Produkt- und Dienstangeboten realisiert. Hierbei wird angenommen, dass ausgehend von einer kleinen Zahl anfänglich angemeldeter Benutzer („Seeding“) bald weitere Be-

²Pica-Pica-Framework: <http://serviceplatform.org/wiki/Pica-Pica>

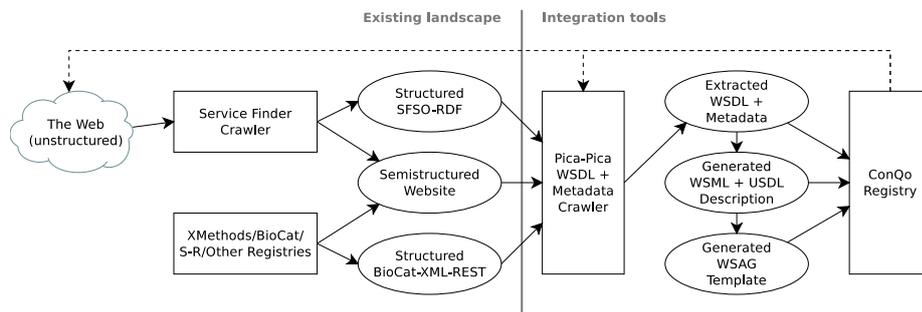


Abbildung 5.2: Architektur und Funktionsweise des Pica-Pica-Crawlers

Tabelle 5.3: Registrierte Benutzer

Quelle	Anzahl
TU Dresden	15
Vellore Institute of Technology	47
Kiewer Polytechnisches Institut	16
Sonstige	21

nutzer aus dem gesamten Web hinzukommen und so eine möglichst unvoreingenommene Bewertung der angebotenen Dienste vornehmen können.

Während der Vorbereitung der Validierungsarbeiten sind vor allem studentische Nutzer aus drei Universitäten (Technische Universität Dresden, Kiewer Polytechnisches Institut und Vellore Institute of Technology) angeworben worden. Insgesamt haben sich in dieser Zeit 99 Benutzer registriert, von denen einige indirekt über existierende Benutzer hinzugekommen sind.

In der Tabelle 5.3 wird die Zusammenstellung der Nutzer in der Evaluierungsumgebung erläutert.

Mindestanzahl an Ausführungsservern

Die SPACE-Plattform ist prinzipiell für die verteilte Dienstauführung unter einer logisch, nicht jedoch notwendigerweise physisch, zentralen Verwaltungsebene konzipiert. Neue Instanzen der Plattform lassen sich durch einen Bootvorgang des Live-Demonstrators SPACEflight in einer virtuellen Maschine oder auf dedizierten Servern innerhalb von Sekunden bilden. Die Zuordnung von installierten Diensten zu Ausführungsservern erfordert jedoch eine diesbezügliche Erweiterung der Basisplattform. Diese ist basierend auf der Verwaltungsschnittstelle für Ausführungsserver *Nodeman* entwickelt worden.

Für das Experiment kommen drei Ausführungsserver zum Einsatz.

Mindestanzahl an dynamischen Parametern

Weitere Parameter sind sowohl von den Basisparametern (Zahl an Benutzern, Diensten und Servern) als auch von der Aktivität der Benutzer abhängig. Diese sollen als dynamische Skalierungsparameter bezeichnet werden. Sie umfassen auf Providerseite die Anzahl der Deployments und auf Konsumentenseite die Anzahl der SLA-Aushandlungen und Dienstauführungen. Systemseitig wird dazu

eine Belastung durch Ausführungs- und Monitoringereignisse erzeugt. Schließlich wird das Gesamtsystem auch durch die Benutzerschnittstellen des Portals belastet, beispielsweise durch den Abruf von Webseiten durch Benutzer und Crawler von Suchmaschinen. Die Zugriffe auf den Webserver sollen an dieser Stelle nicht betrachtet werden. Das Radardiagramm 5.3 zeigt analog zum Diagramm 5.1 die geplanten Zielgrößen der dynamischen Parameter an. Als Technik zum Erreichen dieser Zielgrößen wird eine hybride Simulation in einer speziell vorbereiteten Experimentierumgebung durchgeführt, die im Anschluss an die theoretischen Betrachtungen zum Komplexitätsmodell vorgestellt wird.

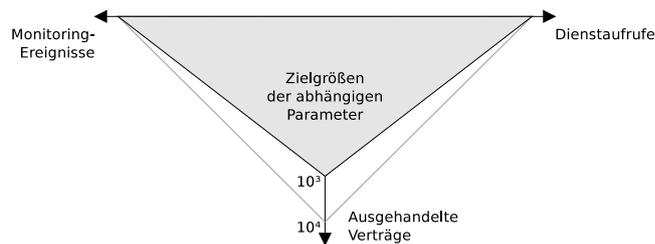


Abbildung 5.3: Zielstellung: Skalierbare abhängige dynamische Parameter

5.1.2 Komplexitätsmodell

Das Komplexitätsmodell stellt sich als eine Kombination der Anzahl von Dienstbietern und -nutzern, verfügbaren Diensten und dynamischen Parametern dar. Hierbei entstehen schwach besetzte Matrizen mit den Dimensionen *Dienste* und *Dienstnutzer*, deren Belegung durch die dynamischen Parameter wie den abgeschlossenen Verträgen oder den aufgerufenen Diensten bestimmt wird. Dies geschieht unter der Annahme, dass zwischen jedem Benutzer und Dienstanbieter höchstens ein Vertrag abgeschlossen wird, jedoch beliebig viele daran gebundene Aufrufe erfolgen können.

Die Komplexität der Berechnungen der Metaqualität bezogen auf m Dienstnutzer und n Dienste ist somit auf $m * n$ Verträge und bei k durchschnittlichen Aufrufen für jeden abgeschlossenen Vertrag auf $m * n * k$ vertragsgebundene Dienstaufrufe beschränkt. Jeder Aufruf produziert ein Monitoringereignis auf der Aufruf- bzw. Instanzebene zusätzlich zum instanzunabhängigen Monitoring, dessen Frequenz abhängig von vorhandenen Vertragsvorlagen und Verträgen ist. Unter der Annahme von l diskreten Zeitpunkten für das instanzunabhängige Monitoring über die Laufzeit des Experiments sind somit $m * n * (k + l)$ vertragsgebundene Monitoringereignisse zu berücksichtigen. Durch vollständige Eliminierung redundanter Ereignisse, beispielsweise der mehrfachen Prüfung der Verfügbarkeit jedes Dienstes für jeden Vertragspartner, wird die Zahl auf $m * n * k + n * l$ reduziert.

Es ist davon auszugehen, dass die Mehrheit der Benutzer nur eine geringe Zahl von Diensten nutzen wird. Dies wird nachfolgend durch die Aufstellung von Aktivitätsmatrizen verdeutlicht.

Aktivitätsmatrizen

Es lassen sich zwei Matrizen gemäß dem aufgestellten Komplexitätsmodell bilden. Die erste, genannt Anbietermatrix A , beinhaltet bezogen auf m Teilnehmer die Verteilung des Angebots von n Diensten. Dabei sind, wie in Abbildung 5.4 dargestellt, unterschiedliche Verteilungen denkbar. Es können im monopolartigen Extremfall alle Dienste von einem Anbieter kommen, im anderen Extremfall wird von jedem Anbieter nur ein Dienst zur Verfügung gestellt. Die Belegung der Elemente der Matrix gibt die Zahl der Beschreibungsdokumente an, die für eine statische Analyse in Frage kommen. Unter Beachtung der Annahme, dass jeder Dienst von genau einem Provider angeboten wird, gilt

$$\forall n a_{mn} = \begin{cases} > 0 & \text{falls } m \text{ Provider für } a_{mn} \\ 0 & \text{sonst} \end{cases}$$

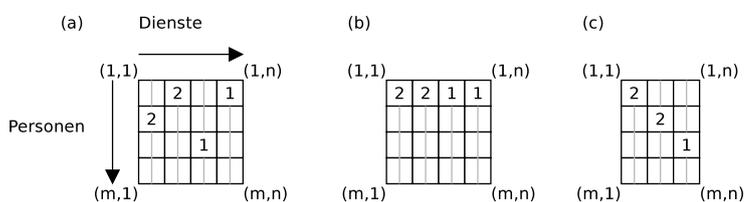


Abbildung 5.4: Matrix der angebotenen Dienste: (a) Normalbeispiel, (b) Monopol, (c) Gleichverteilung

Die zweite Matrix, genannt Benutzermatrix B , enthält die gleiche Grundgesamtheit bestehend aus m Teilnehmern und n Diensten, da die Nutzung und das Anbieten eines Dienstes nur Rollenunterschiede derselben Personen bedeuten. Die Elementbelegung gibt hier jedoch an, ob ein Vertrag abgeschlossen worden ist und wieviele Dienstaufrufe durch diesen Vertrag abgedeckt worden sind. Anonyme Dienstaufrufe sind separat zu betrachten und sind nicht Teil der Experimente. Die Benutzermatrix B ist schematisch in der Abbildung 5.5 dargestellt. Es lassen sich auch hier Extremverteilungen feststellen, wenn etwa nur einzelne Benutzer das gesamte Aufrufaufkommen verursachen oder einzelne Dienste eine Ausnahmepopularität erreichen.

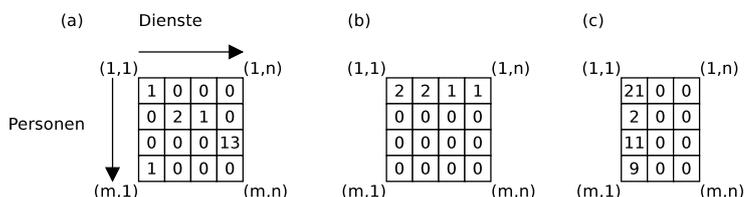


Abbildung 5.5: Matrix der konsumierten Dienste: (a) Normalbeispiel, (b) Power-User, (c) Ausnahmemedienst

Anforderungskatalog

Gemäß der Definition der SMQ-Coverage wird ein Katalog mit einer Liste notwendiger NFE-Angaben und ihrer Priorisierung benötigt. Für die experimen-

Listing 5.1: Anforderungskatalog für SMQ-Coverage

```
1 nfp_req = {  
2   dc:name : required  
3   dc:desc : 100%  
4   availability : 70%  
5   responsetime: 70%  
6 }
```

telle Validierung sollte dieser Katalog nur wenige NFE enthalten, da aufgrund der Dominanz von WSDL-Dateien unter den Dienstbeschreibungen die meisten Dienste nicht mehr bezüglich ihrer Metaqualität unterscheidbar wären. Der speziell für die Validierung erstellte Katalog soll als AK_{val} bezeichnet werden. Als unbedingt notwendig wird eine Namensangabe für einen Dienst betrachtet. Ebenfalls von hoher Priorität ist eine Beschreibung des Dienstes. Beide Angaben sind nicht messbare statische Metadaten. Desweiteren liefern viele, aber nicht alle Dienstportale eine messbare Verfügbarkeitsangabe zu Diensten mit, auch wenn diese nicht in der WSDL-Datei selbst angegeben ist. Hierfür wird eine mittlere Priorität definiert unter der Annahme, dass die SMQ-Prozesse der Dienstplattform ohnehin die fehlenden Verfügbarkeitsangaben ergänzen werden. Die Anforderungen nfp_{req} werden syntaktisch analog zu dem im Listing 5.1 exemplarisch erzeugten Katalog AK_{val} definiert.

Auswirkungen der Komplexität auf die Validierung

Durch die Validierung soll gezeigt werden, dass sich skalierbar auf viele Benutzer und Verträge das Konzept der inkrementellen Qualitätsverbesserungen in Diensthostingumgebungen realisieren lässt und einen belegbaren praktischen Nutzen bringt. Somit ist die Validierung auf zwei Experimente verteilt:

1. Eine Analyse des Overheads, der durch Messungen, Aggregationen und Korrelationen zusätzlich zur Ausführung entsteht. Dieser Overhead kann sowohl auf abstrakter Ebene bestimmt als auch unter Nutzung der vorliegenden Implementierung als Obergrenze angegeben werden.
2. Die Nutzung der errechneten SMQ-Werte zur langfristigen Verbesserung des Dienstangebotes. Durch eine Instrumentierung des Dienstangebotes mit bewusst fehlerhaft gestalteten Diensten soll gezeigt werden, dass diese nach einer bestimmten Zeit geringer bewertet und somit für die Dienstauswahl irrelevant werden.

5.2 Experimentierumfeld für die Validierung

SPACE ist aufgrund der Ausrichtung als erweiterbare Basisplattform nicht direkt für Demonstrations- oder Experimentierzwecke nutzbar. Vielmehr müssen Erweiterungen und Einstellungen zweckgerecht vorgenommen werden [104]. Zur Umsetzung der Anforderungen an die Validierungen werden zwei neue Bestandteile eingeführt:

- Ein Dienstportal als grafische Oberfläche zur Gewährleistung einer einfachen Benutzbarkeit und somit hohen Zahl an Nutzern. Das Portal soll zudem einen Grundumfang an verfügbaren Diensten aufweisen.
- Ein Simulationswerkzeug in Form eines Agenten zur Sicherstellung einer hohen Aktivität in der Dienstumgebung über das natürliche Nutzerverhalten hinaus. Hierbei soll ein Konzept für eine hybride Simulation auf Basis reeller Aktivitäten genutzt werden.

Diese beiden Erweiterungen werden nachfolgend beschrieben.

5.2.1 Dienstportal

Die Dienstplattform SPACE besteht aus einer Vielzahl an Plattformdiensten, die entweder keine Benutzeroberfläche oder nur auf Funktionalität getrimmte Administrationsschnittstellen beinhalten. Eine externe Validierung der Konzepte und Verfahren zur Metaqualität im Feldversuch mit Anwendern ist mit solchen Komponenten kaum durchführbar.

Um dem benutzerzentrischen Ansatz der Plattformdienste die notwendige Attraktivität und Akzeptanz zu verleihen, ist eine Erweiterung der Plattform um eine einheitliche Benutzerschnittstelle notwendig. Diese sollte die in den Plattformdiensten nicht enthaltene globale Funktionalität wie die Registrierung von Benutzern und das Ändern der Profilinformationen gleich enthalten.

Angelehnt an aktuelle Arbeiten zur Kollaboration und zur Schaffung von Communities sieht das Erweiterungskonzept dabei die Einführung eines sozialen Netzwerkes vor. Damit lassen sich klassische Ansätze zur Registrierung von Dienstleistungen in Branchenverzeichnissen, geschäftliche Kontaktnetzwerke und konventionelle SOA-Dienstverzeichnisse vereinheitlicht abbilden [101].

Zur Umsetzung des Dienstportals wurde die auf den Betrieb sozialer Netzwerke spezialisierte und umfassend konfigurierbare Webanwendung *Noosfero* gewählt. Diese auf dem Webframework *Ruby on Rails* aufbauende Anwendung beinhaltet bereits die grundlegenden Funktionen zur Anmeldung und Verwaltung von Benutzern, zur Bildung von Gruppen und zur Darstellung von Angeboten und Nachfragen zu Produkten und nichttechnischen Dienstleistungen. Auf dieses Modell lässt sich das TSM abbilden, wie in Abbildung 5.6 als Entity-Relationship-Diagramm gezeigt ist. Dabei werden angebotene Dienstpakete als Dienstleistungen und zugehörige Clients als bündelbare Produkte repräsentiert.

Die durch die Integration von SPACE-Komponenten erzeugten Instanzen von *Noosfero* lassen sich als soziale Dienstnetzwerke bezeichnen, um die Idee des partizipativen Internets der Dienste zu unterstreichen. Es existieren dabei zwei parallele Instanzen auf der gleichen Codebasis, die sowohl den globalen Betrieb im Rahmen der Validierung als auch den eingeschränkten internen Zugriff in der Lehre ermöglichen.

Die Integration der Dienstplattform in das Portal umfasst die folgenden Punkte:

Bereitstellung Umleitung des Benutzers auf die Dienstverwaltungsanwendung Provider Wizard, wenn eine neue Dienstleistung angeboten werden soll.

Dienstsuche Anzeige von Inhalten und Suchergebnissen nach Abfrage der ConQo-Discovery direkt im Portal.

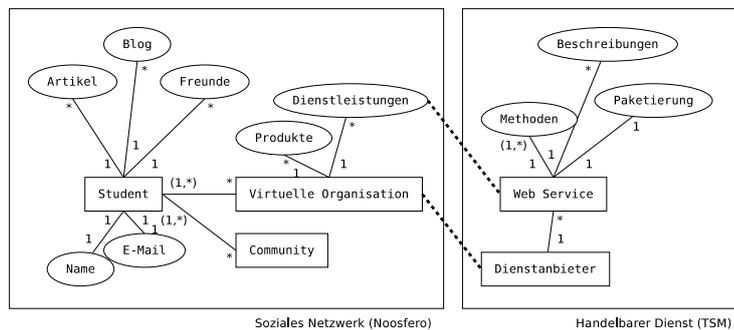


Abbildung 5.6: Konzept für die Einführung elektronischer Dienstleistungen in soziale Netzwerke

Vertragsverwaltung Verwaltung und Visualisierung von ausgehandelten Dienstnutzungsverträgen.

Benutzerdatenbank Bidirektionaler Abgleich zwischen den Benutzerprofilen des Portals und denen der Plattformkomponenten.

Zuerst wurde die Anbindung auf Basis von Verknüpfungen in HTML und programmatischen Aufrufen der Plattfordienste vorgenommen. Bei der geplanten großen Menge an Diensten hat sich die Nutzung der Web-Service-Schnittstellen jedoch als nicht performant genug herausgestellt. Aus diesem Grund ist die auf objektrelationalen Abbildungen (ORM) basierende SQL-Datenbank von Noosfero um Tabellen erweitert worden, die die Inhalte der ConQo-Registry periodisch übernehmen und für den direkten Zugriff über die ohnehin geöffnete Datenbankverbindung bereitstellen. Der Abgleich erfolgt über das Werkzeug *ConQoNoosferoUpdater*.

Zur weiteren Erhöhung der Attraktivität und Aktivität ist das Portal um ein monetäres Anreizsystem ergänzt worden. Durch sie erhält jeder angemeldete Benutzer ein initiales Guthaben in Spielgeldeinheiten, welches durch die vertragsgebundene Nutzung von Diensten reduziert und durch das Anbieten populärer Dienste erhöht werden kann. Derartige Anreizsysteme sind bereits erfolgreich in Prognosesystemen und Börsensimulationen eingesetzt worden, eine Nutzung auf Dienstmarktplätzen ist bisher hingegen noch nicht erfolgt. Eine Auswertung des Nutzungsverhaltens unter finanziellen Gesichtspunkten kann dennoch nur mit den gleichsam alle Domänen betreffenden Einschränkungen der Aussagekraft von Vorhersageergebnissen unter Einfluss von virtueller Währung erfolgen [88]. Die Spielgeldbank wurde als RESTful Web Service unter dem Namen *ServBank* implementiert. Der Dienst bietet Methoden zum Abruf und zur Durchführung von Transaktionen an, welche nur lokal seitens des Portals aufgerufen werden können. Die Transaktionen werden asynchron und tarifabhängig von vertragsgebundenen Dienstaufufen ausgelöst, indem aufgezeichnete Messdaten mit hinterlegten Vertragsinformationen periodisch durch ein *Transaktionstool* auf ausstehende Zahlungen untersucht werden. Die dadurch offensichtliche Möglichkeit, kurzfristig über das Guthaben hinaus Dienstaufufe tätigen zu können, wird aufgrund der vermutlich meist geschäftlichen Beziehungen der Vertragspartner mit einhergehender Forderung von Nachzahlungen zugunsten einer höheren Performance bewusst gestattet. Die Alternative, eine

eingehende Vertrags- und Tarifprüfung zum jeweiligen Aufrufzeitpunkt, würde sich nicht nur negativ auf die durchschnittliche Antwortzeit auswirken, sondern zusätzlich das Risiko eines Ausfalls des Proxydienstes Access Gate als zentrale Komponente erhöhen.

Um das zu erwartende Ungleichgewicht zwischen der ohne Vorbereitung möglichen Dienstnutzung und der mit doch erheblichem Aufwand verbundenen Dienstentwicklung abzuschwächen, sind als weiterer Anreiz Szenariodienste entwickelt worden, deren nichtfunktionale Eigenschaften durch alternative Implementierungen einfach überboten werden können. Zusammen mit den registrierten Benutzern und den installierten Diensten sowie den hinterlegten Beschreibungen externer Dienste stellen sie den Inhalt des Portals dar.

Die resultierende integrierte Kombination der Noosfero-Instanz Crowdserving mit der Dienstplattform SPACE und weiteren Komponenten wird durch Abbildung 5.7 vorgestellt.

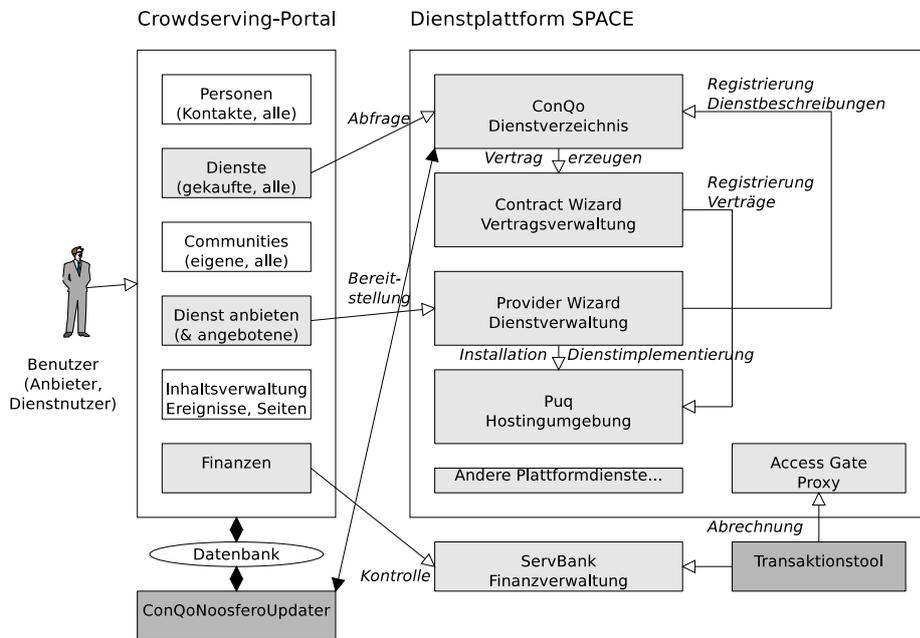


Abbildung 5.7: Integriertes Dienstleistungsportal aus sozialem Netzwerk und Dienstplattform SPACE

5.2.2 Hybride Simulation und Simulationsagenten

Die Erzielung einer möglichst hohen Aktivität von Benutzern hinsichtlich des Anbietens und Nutzens von Diensten, insbesondere auch des Abschließens und Bewertens von Verträgen, wäre trotz möglicher Anreize nur wenig aussagekräftig über das tatsächliche Skalierbarkeitspotenzial einer Plattform. Aus diesem Grund wird das Simulationswerkzeug SPACEagent eingeführt, welcher sich wie die Portaloberfläche für menschliche Benutzer auf die Schnittstellen der SPACE-Plattformdienste stützt. Die Simulation ist auf vielen Gebieten ein verbreitetes und anerkanntes Instrument zur Validierung, wird jedoch bisher für Problemstellungen in dienstorientierten Umgebungen und speziell im Internet der Dienste

nur sehr selten eingesetzt, etwa zur Überprüfung geschäftlicher Auktionsvorgänge [59]. Um die Vorteile der Planbarkeit einer Simulation mit den Eigenschaften einer produktiven Plattform verknüpfen zu können, bietet sich die Variante einer hybriden Simulation an [94, 60]. In dieser Simulationsart wird dual zum Real-system ein Replikat aufgebaut. Die Koexistenz und der Abgleich der Systeme sind in der Abbildung 5.8 zu sehen. Auf der weitestgehend isoliert ablaufenden replizierten Plattform werden Simulationstechniken genutzt, um künstliche Nutzeraktivitäten zu generieren, die wiederum reelle Systemaktivitäten auslösen und damit weitere Einträge in den Datenbanken der Plattform erzeugen. Dadurch können Benutzer, Dienste, Verträge, Aufrufe und Monitoringdaten abhängig vom vorherigen oder parallelen Verhalten auf dem Realsystem erzeugt werden.

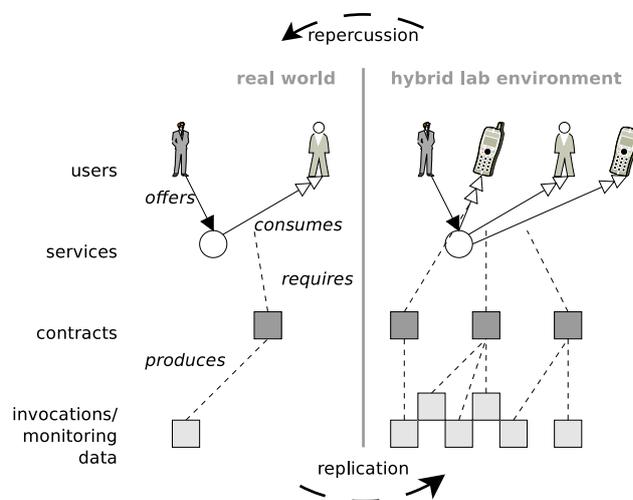


Abbildung 5.8: Abläufe der hybriden Simulation

Der Replikationsprozess kann grobgranular durch die Kopie und erneute Instanzierung eines Abbilds der virtuellen Maschine der Dienstplattform vorgenommen werden. Möglich ist jedoch auch eine feingranulare Replikation durch das Kopieren einzelner Dateien und die Sicherung und Wiedereinspielung der Inhalte von SQL- und XML-Datenbanken. Das feingranulare Verfahren ist wesentlich flexibler. Es erlaubt beispielsweise die zwischenzeitliche Anonymisierung der Benutzerdaten und weitere Modifikationen für die Aufbereitung und Publikation von Momentaufnahmen der Plattformaktivitäten. Desweiteren wird in dem Verfahren ein wiederholter Abgleich mit dem Realsystem aufwandsarm ermöglicht. Durch die Snapshot-Fähigkeiten moderner Dateisysteme und Virtualisierungstechniken ist es in beiden Verfahren möglich, zu jedem beliebigen Zeitpunkt in sich konsistente Momentaufnahmen zu generieren.

Auf der replizierten Plattform werden Schlüsselmetriken des IoS wie etwa die Zahl registrierter Benutzer, angebotene Dienste, ausgehandelte Dienstgütevereinbarungen (SLAs) und überwachte Dienstauführungen gesammelt, ausgewertet und durch Simulationen rückwirkend in gleicher oder modifizierter Abfolge und Frequenz eingespeist. Einstellbar sind dabei vor allem die Wartezeiten zwischen den Aktivitäten, die bei Beachtung kausaler Abhängigkeiten für die Simulation stark verkürzt werden können. Abhängigkeiten sind in konven-

tionellen Dienstplattformen kaum von Bedeutung. In rückgekoppelten Plattformen hingegen haben vorherige Dienstaussführungen durch geänderte NFE-Trägerdokumente, abgegebene Bewertungen und andere dynamische Faktoren wie langfristig adaptierte Dienstimplementierungen einen starken Einfluss auf spätere Nutzungsmuster und müssen aus dem Grund beachtet werden. Die in dieser Arbeit betrachtete Simulationstechnik beschränkt sich auf die Erzeugung neuer Verträge und Dienstaussführungen. Das simulierte Anbieten neuer Dienste wäre ungleich komplexer und vermutlich auf die Permutation numerischer Daten wie die Wertzuweisungen nichtfunktionaler Eigenschaften in Dienstbeschreibungen beschränkt. Weitere Einschränkungen resultieren aus den Unterschieden der Konnektivität der Ursprungsplattform und der Replika zu den aufrufenden Clients. Somit müssten Netzwerkeigenschaften komplementär simuliert werden. Desweiteren ist die Replikation bisher nur für Einzelplattformen ausgelegt. Eine Übernahme von Daten aus verteilt arbeitenden Dienstlandschaften ist schon aufgrund der möglichen unterschiedlichen Eigentumsverhältnisse und Verantwortungsbereiche sowohl organisatorisch als auch technisch schwierig und derzeit noch Gegenstand der weiteren Forschung.

Für die Generierung neuer Metriken werden Agenten eingesetzt, die das menschliche Verhalten bezüglich der Anforderungen an Dienste und SLAs simulieren. Die Aufgaben der Agenten umfassen die Auswahl von Diensten aus der Registry, die nachfolgende Aushandlung von SLAs und die Nutzung des resultierenden Vertrags zur Ausführung einer vertragsgebundenen Dienstaushandlung. Während der Laufzeit der Simulation werden durch die Plattform weitere Monitoringdaten unabhängig von den Agenten generiert, beispielsweise durch die stetige Überwachung der Verfügbarkeit von Dienstendpunkten durch regelmäßig gesendete HTTP-Prüfanfragen.

Der wesentlicher Vorteil des hybriden Ansatzes ist der durch die zu erwartende Zunahme an Dienstplattformen zukünftig reduzierte Simulationsanteil bei gleichzeitig höherem Anteil an reell beobachteten Aktivitäten. Der Vorteil einer Nutzung von Agenten für die hybride Simulation ist die Rückwirkung der diskreten Simulationsereignisse auf die Plattform, da der Simulator die gleichen programmatischen Plattformschnittstellen nutzt wie die über eine Benutzerschnittstelle angebotenen menschlichen Benutzer.

Der Aufbau des SPACEagent ist aus der Abbildung 5.9 ersichtlich. Er besteht aus einem in Python implementierten Hauptprogramm, mehreren als Threads gestarteten Agentenobjekten und für die Ausführung der Agententätigkeit notwendigen externen Werkzeugen. Das Hauptprogramm verwaltet die Agenten und verarbeitet die Konfigurationsdaten, die unter anderem die Endpunkte der genutzten SPACE-Plattformdienste enthalten. Die externen Werkzeuge können durch ihre adressräumliche Trennung in unterschiedlichen Programmiersprachen entwickelt werden. Die für das Experiment umgesetzten Werkzeuge umfassen ein Ruby-Skript zur automatischen Aushandlung von Verträgen sowie einen dienstspezifischen automatischen Aufrufer. Die Entwicklung eines generischen Aufrufers ist mit heutigen Techniken hingegen aufgrund beliebig komplexer funktionaler Schnittstellen mit ungenügender semantischer Beschreibung nur bedingt im Rahmen der Erzeugung von Testfällen möglich [4, 5]. Die Einschränkung für das Experiment lautet somit, dass die Dienstaufrufe nur für eine vorher bestimmte Teilmenge der verfügbaren Dienste hybrid simuliert werden können. Aus Praktikabilitätsgründen wird diese Teilmenge zudem klein gegenüber dem gesamten Dienstangebot und somit nur eingeschränkt repräsentativ sein.

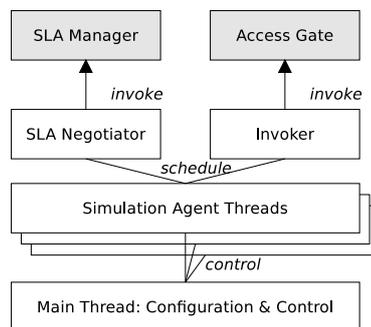


Abbildung 5.9: Simulationswerkzeug SPACEagent

5.2.3 Gesamte Simulationsumgebung

Die um die Metaqualitätsanalyse, die Portalumgebung und die Experimentierwerkzeuge erweiterte Validierungsarchitektur auf Basis der SPACE-Plattform ist in der Grafik 5.10 abschließend dargestellt. Als Werkzeuge kommen neben den Simulationsagenten auch die vorgestellten Tools Pica-Pica als Web-Service-Beschreibungs-Crawler, die Transformatoren für die Generierung von WSML und WS-Agreement, ConQoNoosferoUpdater für die Portalintegration sowie das MPWS-System zum Einsatz.

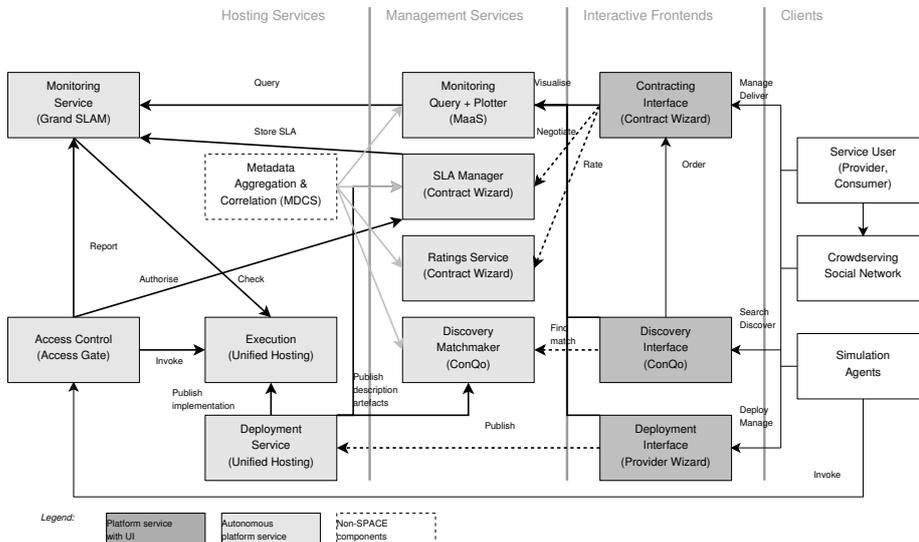


Abbildung 5.10: SPACE mit Portal- und Agentenerweiterungen sowie dem Metadaten-Korrelationsdienst

Neben der rein architektonischen Betrachtung soll an dieser Stelle auch auf die konkrete Instanziierung mitsamt der für das Experiment wichtigen Datenquellen und Abläufe hingewiesen werden. Schematisch ist die Durchführung in Abbildung 5.11 zu erkennen.

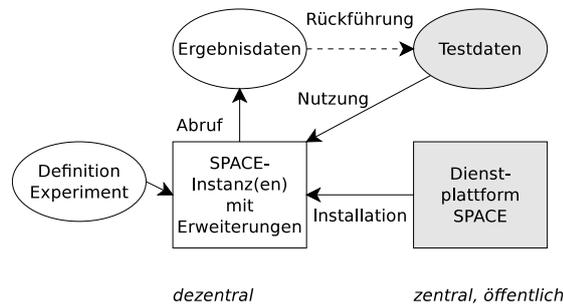


Abbildung 5.11: Durchführung des Experiments

5.3 Experimentelle Abläufe und Ergebnisse

Das öffentliche soziale Dienstnetzwerk als Einstiegspunkt in das Experimentiersystem auf Basis der Dienstplattform SPACE ist unter dem Namen *Crowdserving*³ Mitte Dezember 2009 gestartet und ab Anfang Januar 2010 zunehmend bekannt gemacht worden. Der Verlauf der Entwicklung aller Kennzahlen in der Dienstplattform ist im Diagramm 5.12 ersichtlich. Die durchgeführten Experimente auf Basis dieser Daten begannen dabei bei einer Konfiguration der Aktivitätsmatrizen von $m = 99$ Personen und $n = 87$ installierten Diensten mit Stand vom 19. Mai 2010. Zum Startzeitpunkt waren noch keine Beschreibungen externer Dienste hinterlegt, und nur einer der installierten Szenariodienste war mit zwei SLA-Vorlagen für feststehende und verhandelbare QoS-Eigenschaften versehen. Zum Zeitpunkt der Replikation aller Daten für die kontrollierte Durchführung von Experimenten in der replizierten Plattforminstanz am 27. Mai 2010 waren $m = 100$ Personen registriert und $n = 471$ Dienste installiert. Von 35318 WSDL-Beschreibungen als Ergebnismenge des Pica-Pica-Crawlers waren 15547 ergänzend dazu in die Registry eingetragen. Bis zum Schlußtag der Beobachtungen auf dem Realsystem am 22. Juni 2010 erhöhte sich die Zahl der Benutzer geringfügig auf $m = 103$.

Im Vergleich mit den geplanten Größenordnungen für die Kennzahlen liegt die erreichte Ausbaustufe des Dienstportals teilweise unter und teilweise über den Planungen. Die Abbildung 5.13 verdeutlicht auf logarithmischer Skala den Grad der Erfüllung der Plankennzahlen. Trotz der nicht vollständig erreichten Wachstumsziele stellen diese Daten eine für die Skalierbarkeitsuntersuchungen brauchbare und für experimentelle Untersuchungen im Internet der Dienste repräsentative Grundlage dar, insbesondere auch durch die Möglichkeit der Ergänzung im Rahmen der hybriden Simulation. In den folgenden Abschnitten werden daher der Verlauf der Experimente und die daraus gewonnenen Ergebnisse dargelegt, bevor anschließend eine Auswertung hinsichtlich der Validierungskriterien erfolgt.

5.3.1 Ausbau der Grundkonfiguration

Zur Gewährleistung der Überwachung aller registrierten Dienste hinsichtlich ihrer nichtfunktionalen Eigenschaften wurde das Werkzeug `wsdl2wsag` zur Generierung von SLA-Vorlagen aus WSDL-Dateien auf alle über den Pica-Pica-

³Crowdserving-Portal: <http://beta.crowdserving.com:3000/>

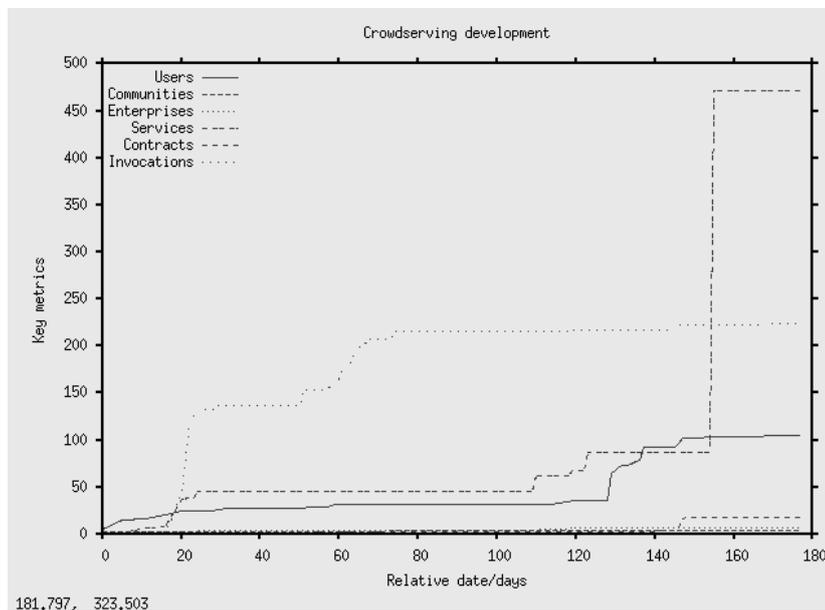


Abbildung 5.12: Wachstum aller Kennzahlen im Crowdserving-Portal während der Validierungsphase

Crawler bezogenen Dienstbeschreibungen mit Ausnahme des Service-Finder-Katalogs angewendet. Dabei wurden die von den Ursprungsportalen extrahierten assoziierten Metadaten zur Verfügbarkeit ausgewertet und als SLO-Angaben in die SLA-Vorlagen eingetragen. Von 35318 WSDL-Dateien konnten im ersten Lauf lediglich 14317 erfolgreich genutzt werden. Die Komplementärmenge bestand aus komplexeren WSDL-Strukturen, die mehrere Dienste oder Dienste mit mehreren Endpunkten beschreiben. Ihre eindeutige Abbildung auf das Format WS-Agreement+ValueTypes erfordert Änderungen an dem Format zur Aufnahme mehrerer zu überwachender Endpunkte pro Vertrag. Eine solche Änderung widerspräche der Unterstützung existierender heterogener Formate. Aus diesem Grund ist als zweckmäßige Lösung die Nutzung des jeweils ersten gefundenen Endpunktes in `wsdl2wsag` implementiert worden. Damit stieg die Zahl der erfolgreich generierten WSAG-Dateien auf 33156 und die Quote auf 93,9%. Der Generierungsvorgang für diese Menge wurde auf dem Testsystem in 120 Minuten und 2 Sekunden abgeschlossen. Das Resultat der Generierung ist ein deutlicher Ausbau der Grundkennzahlen der Experimentierumgebung hinsichtlich der zur Überprüfung der Skalierbarkeit des Monitors notwendigen SLA-Vorlagen.

5.3.2 Durchführung der Evaluierungs-Sprints

Bis Mai 2010 wurden mehrere Evaluierungs-Sprints mit realen Benutzern zur Verbesserung der Ausgangslage für die eigentlichen Evaluierung durchgeführt, um über die erreichten Grundkennzahlen hinaus weitere Kennzahlen wie ausgehandelte SLAs und aufgerufene Dienste steigern zu können. Die Evaluierungs-sprints sind nach dem folgenden Schema in fünf Teilschritten durchgeführt worden:

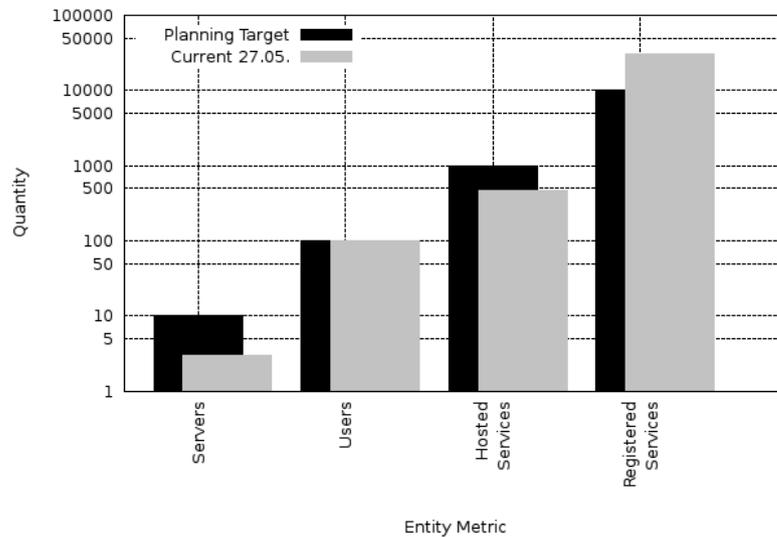


Abbildung 5.13: Vergleich Plankennzahlen zu erzielten Kennzahlen

1. Kontinuierlicher Aufbau des Netzwerks durch Gewinnung von Benutzern und Installation von Diensten, wie vorangehend erläutert
2. Auswahl einer Evaluierungsgruppe
3. Durchführung des Sprints zur Dienstsuche, SLA-Aushandlung und Dienstauftrag innerhalb weniger Stunden
4. Replikation der Dienstplattform und hybrid simulierte Weiterführung des Experiments
5. Nachbearbeitung und Auswertung der Ergebnisse

Stellvertretend soll ein Evaluierungssprint an dieser Stelle detailliert dargestellt werden. Es wurden 44 Studenten des Vellore Institute of Technology (VIT) in Tamil Nadu, Indien, zur Teilnahme ausgewählt. Die Aufgabenstellung umfasste die Auswahl eines in WSML, WSDL und USDL beschriebenen Szenariodienstes mit einer fixen und einer variablen Vertragsvorlage im WS-Agreement-Format. Drei Teilnehmer sind durch ungültige E-Mail-Adressen ausgeschlossen worden. Weitere 28 Teilnehmer haben entweder den Dienst nicht gefunden oder sind noch vor Abschluss der SLA-Aushandlung ausgeschieden, so dass insgesamt nur 13 SLAs ausgehandelt wurden. Fünf Teilnehmern gelang es auf Basis dieser Verträge, den Dienst vertragsgebunden auszuführen.

Die beobachteten Aktivitäten und gesammelten Daten wurden genutzt, um eine Instanz des Simulationsagenten zu parametrisieren. Die zeitlichen Abstände zwischen den Aktivitäten und damit die Phasen der Inaktivität wurden minimiert, und zufällige zeitliche Abweichungen von den ursprünglichen Aktivitäten in einem Toleranzbereich von ± 10 s eingestellt. Dadurch wurden die Frequenz und die probabilistische Verteilung der Aktivitäten auf die Plattformreplika als aktivitätsverstärkte logische Weiterführung der Benutzerstudie abgebildet.

Listing 5.2: Log eines Durchlaufs des Simulationswerkzeugs *SPACEagent*

```
1 $ ./spaceagent --probability 0.60
2 SpaceAgent: Create new agent agent_0
3 [Agent agent_0] starting...
4 [Agent agent_0] sleep...
5 SpaceAgent: Check
6 SpaceAgent: Check
7 [Agent agent_0] invoke service
8 Invoke FortuneService at http://192.168.0.8:9000/
   FortuneService...
9 SpaceAgent: Check
10 [Agent agent_0] sleep...
11 SpaceAgent: Check
12 [Agent agent_0] create SLA
13 SpaceAgent: Check
14 Fetch SLA templates...
15 * FortuneServicePremium Flexible
16 Create and register offers ...
17 * FortuneServicePremium Flexible -> FortuneServicePremium
   Flexible|1274986761.26182
18   -> FortuneServicePremium Flexible|1274986761.26182
19 Negotiate SLAs ...
20 * FortuneServicePremium Flexible|1274986761.26182
21   -> Negotiation finished. Agreement is now active!
22 [Agent agent_0] sleep...
23 SpaceAgent: Check
```

5.3.3 Durchführung der Simulation

Auf Grundlage der gesammelten Grundkennzahlen zum Systemausbau und durch die Sprints erreichten Aktivitätskennzahlen wurde ein Systemreplikate erzeugt und ein Datenbankabgleich durchgeführt. Anschließend wurde die hybride Simulation darauf angewendet, um die Aktivitätsmetriken zu steigern. Durch die daraus resultierende erhöhte Systembelastung lassen sich die Grenzen der SPACE-Plattformdienste und der darauf aufsetzenden Werkzeuge zur Bestimmung, Propagierung und Verbesserung der Metaqualität experimentell bestimmen.

Zur Veranschaulichung der Vorgänge der Simulation wird zuerst exemplarisch der Simulationsagent mit einer geringen Gesamtaktivitätswahrscheinlichkeit bezogen auf den Szenariodienst *FortuneService* ohne Vertragsvorlage und den Szenariodienst *FortuneServicePremium* mit Vertragsvorlage aufgerufen. Die Logausgabe 5.2 des Agenten zeigt an, wann welcher Thread einen SLA aushandelt oder einen Dienstaufwurf durchführt. Erkennbar ist auch der Hauptthread, welcher in regelmäßigen Abständen überprüft, ob die Agententhreads noch aktiv sind oder nach Ablauf ihrer Tätigkeit eingesammelt werden müssen.

Bezogen auf die ermittelten Aktivitätskennzahlen im Dienstportal wurde die durchgeführte Simulation in der Replika mit unterschiedlichen Aktivitätswahrscheinlichkeiten für SLA-Aushandlungen und Dienstaufwürfe konfiguriert. Das Verhältnis der beiden Wahrscheinlichkeiten zueinander gleicht dem Ver-

hältnis der Kennzahlen. Die absoluten Wahrscheinlichkeiten enthalten die Inaktivität der meisten Benutzer. Somit betrug $P(\text{negotiation}) = 13/41 = 0,317$ und $P(\text{invocation}) = 5/41 = 0,122$. Die Zahl der Threads wurde auf 100 konfiguriert. Somit lautet die Aufrufparametrisierung des Simulationsagenten:

```
1 $ ./spaceagent --agents 100 --probability:negotiation 0.317
   --probability:invocation 0.122
```

Das resultierende Wachstum der simulierten Kennzahlen innerhalb einer Experimentdauer von 25 Minuten wird durch das Diagramm 5.14 verdeutlicht. Erkennbar ist die vom Realsystem übernommene Ursprungsconfiguration sowie die unterschiedlichen Wachstumsgeschwindigkeiten aufgrund der Parametrisierung.

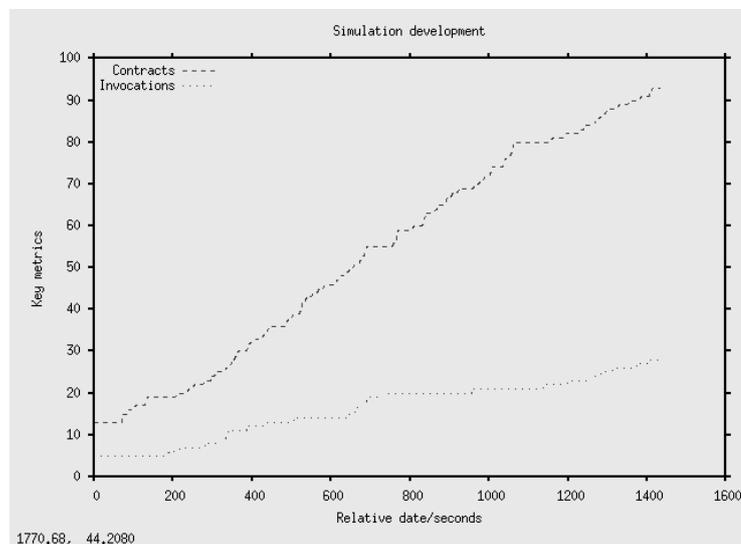


Abbildung 5.14: Wachstum der simulierten Kennzahlen

5.3.4 Bestimmung und Verbesserung der Metaqualität

Zur initialen Bestimmung der Metaqualität wurde das MDCS-Werkzeug `mcs-local` verwendet, welches eine für Experimentierzwecke günstige exakte Parametrisierung des Metadaten-Korrelationsdienstes ermöglicht. Damit soll an einem isolierten Beispiel die Vorgehensweise demonstriert werden, bevor die Ergebnisse des Experiments über alle Dienste erläutert werden.

Das dargestellte Ablaufbeispiel zeigt exemplarisch an, wie die Metaqualität für den Szenariodienst *FortuneServicePremium* über eine Abfrage der SPACE-Plattformdienste bestimmt wird. Der Dienst ist dabei mit einer Vertragsvorlage, einem abgelaufenen Vertrag, einem vorher durchgeführten Aufruf und einer abgegebenen Bewertung versehen. Unter Nutzung des spezifischen Anforderungskataloges AK_{val} wird die SMQ-Coverage ermittelt. Hierbei erfolgt eine Abwertung aufgrund der nicht vorhandenen NFE-Angaben in der WSML-Dienstbeschreibung sowie eine weitere, geringe Abwertung der fehlenden Angabe der Antwortzeit im Vertrag. Die Bestimmung der SMQ-Validity erfolgt auf Basis

der abgerufenen Monitoringdaten. Im Beispiellog 5.3 und 5.4 stimmt die Verfügbarkeit mit der vertraglichen Zusicherung überein, so dass die SMQ-Coverage maximiert wird. Die Gesamtmetriken SMQ wird allerdings durch die in der ersten Iteration nicht durchführbare Validierbarkeit der Antwortzeit abgewertet.

Da zu Beginn keine vollständige Abdeckung der geforderten NFE durch den SLA angenommen werden kann, werden Proxy-Monitoringdaten hinzugenommen, die SLA-unabhängig ermittelt und für die dynamische Eintragung über einen Vorhersagemechanismus genutzt werden. Im gegebenen Fall sieht ein globaler SLATMC vor, die Monitoringdaten mit einem Puffer von 10% als SLAT-Grenzwerte einzutragen. Somit wird die Antwortzeit auf 2.224s festgelegt. Als Wertebereichsgrenzen werden basierend auf Intervallschritten 0s und 10s bestimmt. Die Dienstbeschreibung wird hingegen mit den tatsächlichen Durchschnittswerten dynamisch aktualisiert.

Nach Aushandlung eines neuen Vertrags und einigen erneuten Aufrufen liefert der Monitoringdienst leicht abweichende Werte wie im Log 5.5 gezeigt.

Die Antwortzeit ist in der SLA-Vorlage und im SLA vorhanden und liegt nach Überprüfung durch den Monitor im gültigen Bereich. Auch in der Dienstbeschreibung sind die Werte vorhanden. Allerdings sind sie aufgrund der direkten Übernahme der Einzelwerte anstelle statistisch ermittelter Wertebereiche einer permanenten Abwertung in der SMQ-Validity unterworfen. In dem im Log 5.6 nachvollzogenen Fall weicht die nicht konstante Antwortzeit allerdings nur minimal ab, so dass bezogen auf den Wertebereich die SMQ-Validity für diese NFE 0.999895125 und damit gerundet 1.0 beträgt.

5.4 Auswertung des Experiments

Insgesamt sind die folgenden Metriken als Grund- und Aktivitätskennzahlen im Experiment gebildet worden:

- 103 registrierte Benutzer
- 2 angebotene Dienste mit Vertragsvorlage und 469 ohne Vertragsvorlage
- 15547 registrierte und 19771 weitere nicht angebotene externe Dienste, davon 33156 mit Vertragsvorlage
- 94 ausgehandelte SLAs, davon 81 simuliert
- 35 Dienstaufrufe, davon 30 simuliert
- 189965 Monitoringdatensätze nach 15718 Sekunden (12,09 pro Sekunde), sowie zusätzlich 35 aufrufbezogene Einträge
- 2 Vertragsbewertungen

Die Aushandlung von Verträgen mit dem Szenariodienstanbieter ergab vor der Simulation eine Abdeckung von 13%. Während der Simulation steigerte sich der Wert auf 94%. Aufgrund der Beschränkung auf einen Dienst entspricht dieser Wert auch gleichzeitig der Auslastung des Komplexitätsmodells. Hinsichtlich der Dienstaufrufe wird die Zahl der durchschnittlichen Dienstaufrufe pro Vertrag, mit $k = 0,37$ bestimmt.

Listing 5.3: Log eines Aufrufs des Metadaten-Korrelationsdienstes MDCS

```

1  $ ./mdcs-local --requirements required/akval.ini --config
      replica.ini FortuneServicePremium
2  Loading configuration from replica.ini ...
3  Running local MDCS for service FortuneServicePremium ...
4  // fetch requirements from required/akval.ini
5  ** fetch description(s) from http://192.168.0.8:8080/
      Matchmaker/
6  ** fetch contract(s) from http://192.168.0.8:8080/SLAManager
      /
7  ** fetch monitoring data from http://192.168.0.8:8080/maas/
8  ** fetch proxy measurements from http://accessgate.vtsr/
9  ** fetch ratings from http://192.168.0.8:4567/
10 ** fetch contracts from http://contractwizard.vtsr/
11 ## maas, using SOAP, WSDL: file:mdcs/wsd1/MaaSWebService.
      wsdl
12 -- SLA Fortune Service Sample Template
13 >> Availability
14     pred= None avg= 1.0
15 >> responseTime
16     pred= None avg= None
17 ## conqo, using SOAP, WSDL: file:mdcs/wsd1/ConQo-
      ClientAccess.wsdl
18 -- Service FortuneServicePremium [ urn:FortuneServicePremium
      .wsml#FortuneServicePremium ]
19 ## proxy, using HTTP: http://accessgate.vtsr//
      prozyaggregator.php?service=FortuneServicePremium
20 -- throughput 754.47275
21 -- responsetime 2.02170875
22 -- success 1
23 ## sla manager, using SOAP, WSDL: file:mdcs/wsd1/CWNR-
      SLAManager.wsdl
24 -- SLAT FortuneServicePremium Flexible
25 -- SLA FortuneServicePremium Flexible|1274983523346
26 ## ratings, using HTTP: http://192.168.0.8:4567/ratings/
      service/FortuneServicePremium
27 -- Rating Availability 5.0
28 -- Rating reponseTime 5.0
29 ## contracts, using HTTP: http://contractwizard.vtsr//
      get_contract.rhtml
30 -- SLA 17 finished
31 -> FortuneServicePremium Flexible

```

Listing 5.4: Log des Metadaten-Korrelationsdienstes MDCS (Fortsetzung)

```

1  ## loading requirements catalogue
2  == Package requirementscatalogue [requirementscatalogue] ==
3  -- Service dummy [dummy] --
4  -- Description [ini] --
5  -- Property responsetime/None --
6  Measured value (average): ranges
7  -- Property availability/None --
8  Measured value (average): singlevalue
9  ##> http://192.168.0.8:8080/Matchmaker//getInterfaceDoc.jsp?
    iri=urn%3AFortuneServicePremium.wsml%23
    FortuneServicePremium&doctype=WSDL
10 == Package servicedescription [servicedescription] ==
11 -- Service dummy [dummy] --
12 -- Description [wsml] --
13 SMQ-Coverage [SD] 0.58823529411764708
14 SMQ-Validity [SD] None
15 SMQ-Coverage [SLAT] 0.79411764705882348
16 SMQ-Validity [SLAT] 1.0
17 SMQ 0.84558823529411764

```

Listing 5.5: Logausschnitt aus MDCS-Aufruf: SLA-Vorlagen

```

1  -- SLA Fortune Service Sample Template
2  >> Availability
3     pred= None avg= 1.0
4  >> responseTime
5     pred= None avg= 2.02066

```

Listing 5.6: Logausschnitt aus MDCS-Aufruf: Ergebnis

```

1  SMQ-Coverage [SD] 1.0
2  SMQ-Validity [SD] 1.0
3  SMQ-Coverage [SLAT] 1.0
4  SMQ-Validity [SLAT] 1.0
5  SMQ 1.0

```

5.4.1 Aussagen zur Korrektheit

Der Nachweis der Korrektheit aller beteiligten Komponenten und Algorithmen zur Berechnung der Metaqualität soll über eine Stabilitäts- und Konvergenzbeurteilung erfolgen.

Die Stabilität der Metaqualitätsangaben ist eine notwendige, wenn auch nicht hinreichende Größe. Sind Metaqualitätsangaben instabil, so deutet dieser Umstand darauf hin, dass qualitative Schwankungen der Dienstauführung zu übermäßig hohen Abweichungen von den in den NFE-Trägerdokumenten spezifizierten Werten führen und diese demnach relaxiert werden müssen (Notwendigkeit). Sind die Angaben hingegen stabil, so ist dadurch allein noch nicht sichergestellt, ob sie nicht insgesamt additiv verschoben sind oder multiplikativ gestreckt oder gestaucht sind (fehlende Hinreichbarkeit). Vermutet wird, dass die Metaqualitätsangaben nach anfänglich chaotischem Verhalten schnell in einen geringen Schwankungsbereich überführt werden können, wie in Abbildung 5.15.

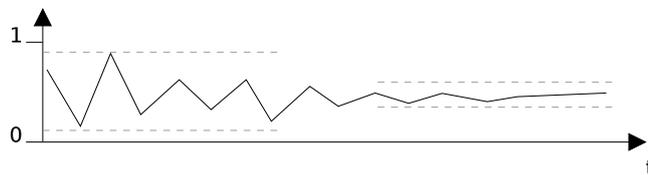


Abbildung 5.15: Vermutete Entwicklung der Stabilität der Metaqualitätsangaben

Die Konvergenz der Metaqualitätsangaben bedingt neben der Rückführung der Werte auch die Anpassung der NFE-Angaben. Idealerweise konvergiert die Gesamtmetaqualität stets gegen 1, da sämtliche fehlenden NFE-Angaben aufgefüllt und abweichende NFE-Angaben korrigiert würden. Vermutet wird, dass durch in der Praxis auftauchende kurzzeitige Schwankungen dieser Wert weder kurz- noch langfristig erreicht wird. Die Vermutung wird in der Abbildung 5.16 skizziert.

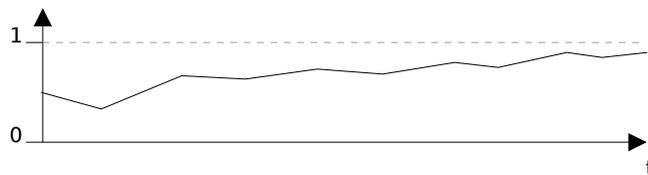


Abbildung 5.16: Vermutete Entwicklung der Konvergenz der Metaqualitätsangaben

Nach 1000 Durchläufen der Metaqualitäts-Korrelationsdienstes MDCS bezogen auf einen exemplarischen Dienst ergibt sich über alle errechneten Metaqualitätswerte das Diagramm 5.17 unter Beinhaltung der Konvergenz- und Stabilitätseigenschaften der SMQ-Validity. Die SMQ-Coverage wird hierbei nicht betrachtet, da sämtliche Ergänzungen im ersten Durchlauf erfolgen können. Die SMQ-Validity wird je nach Vorhersagetechnik auf die Durchschnittswerte, den Gesamtwertebereich und den Interdezilbereich berechnet.

Bei Nutzung des jeweils letzten Messwertes als neuer NFE-Angabe ohne Anwendung gleitender Mittelwertbildung oder sonstiger Glättungsverfahren tritt wie erwartet fast immer eine Abweichung ein, so dass die SMQ-Validity selten

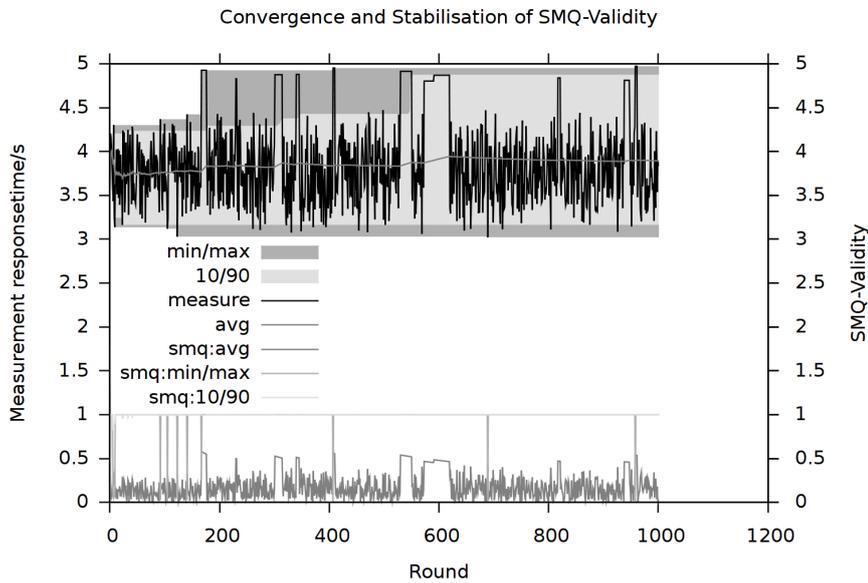


Abbildung 5.17: Tatsächliche Entwicklung der Konvergenz und Stabilität der Metaqualitätsangaben

über 0,5 steigt. Eine Konvergenz ist in dem Fall nicht erkennbar. Wesentlich bessere Übereinstimmungen lassen sich mit Wertebereichen erzielen. Die Konvergenz gegen den Wert 1,0 wird bereits nach 10 Durchläufen erreicht. Die Stabilität der SMQ-Angabe mit Interdezil-Bezug manifestiert sich in einer auf die letzten 300 Durchläufe bezogene maximale Abweichung von $\epsilon = 0.0014$, bezogen auf die letzten 100 Durchläufe sogar nur $\epsilon = 0.0011$. Meist noch bessere Werte, dafür allerdings stellenweise auch wesentlich höhere Abweichungen, weist die SMQ-Angabe mit Bezug auf den Gesamtwertebereich auf.

Auf der vom Crowdserving-Portal ausgehend replizierten Experimentierplattform konvergiert die Metaqualitäts-Teilmetrik SMQ-Coverage über alle Dienste, und damit die SMQ-Coverage des Portals, unter Nutzung des Anforderungskataloges AK_{val} gegen den Wert 0,795. Auffällig sind vor allem die nicht automatisch nachtragbaren Antwortzeiten aufgrund der geringen Zahl der realen und nicht simulierbaren Aufrufe. Diese bestimmen auch die Geschwindigkeit der Konvergenz, da alle übrigen Werte aus dem Anforderungskatalog bereits nach einer statischen Analyse von Dienstnamen und -beschreibungen und einem Durchlauf des Monitorings mit nachgetragenen Verfügbarkeitswerten vervollständigt sind. Die höhere Gewichtung der statischen Merkmale in AK_{val} ist in diesem Fall förderlich.

Die SMQ-Validity konvergiert über die betrachtete Teilmenge der Dienste gegen den Wert 0,986. Als Gesamtmetriek ist die Metaqualität des Portals aufgrund der nur teilweisen Berechnung der SMQ-Validity nicht bestimmbar, allerdings unter der Annahme der Gleichverteilung und Repräsentanz der ausgesuchten Teilmenge als Stichprobe über die Grundgesamtheit aller registrierten Dienste auf den Wert $SMQ(Crowdserving) = 0,89$ abschätzbar.

Tabelle 5.4: Zeitbedarf SPACE-Plattformdienste bei Aufruf durch MDCS

Plattformdienst	Zeitanteil
ConQo-Registry	0, 147s
MaaS	0, 533s
SLA-Manager	0, 176s
Ratings	10, 29s

5.4.2 Aussagen zur Skalierbarkeit

Die Skalierbarkeitsuntersuchungen beziehen sich sowohl auf die durchgeführten Experimente insgesamt als auch auf die Einzelkomponenten während der durchgeführten Prozessschritte wie beispielsweise der Abfrage von Dienstbeschreibungen aus der Registry oder die Durchführung der Dienstüberwachung.

Monitor Grand SLAM

Die Startphase des Monitors umfasst das Einlesen und Interpretieren der registrierten SLAs. Das Laden von 33156 SLA-Vorlagen mit einer Gesamtdateigröße von 263 MB wurde auf dem Testsystem in 572,2 Sekunden abgeschlossen. Dies entspricht einer Verarbeitungszeit von 16,4 Millisekunden pro Datei. Deutlich weniger performant ist der Aufruf des Parsers zur Extraktion der SLO-Angaben und Aktivierung der Mess- und Aggregationsbundles für jeden zu überwachen Parameter. Mit zunehmender Zahl an Monitoringtasks sinkt die Performanz deutlich. Dieser Vorgang dauerte für die ersten 883 SLAs 1510 Sekunden und somit 1,7 Sekunden pro Datei. Erst nach 15598 Sekunden waren 1000 SLAs verarbeitet, dies entspricht 15,6 Sekunden pro Datei. Sowohl Implementierungsentscheidungen als auch Systembedingungen wie die Zahl der gleichzeitig in die Verarbeitung einbeziehbaren Dateien tragen zu dem Ergebnis bei.

Die Skalierbarkeit der Arbeitsphase des Monitors wird durch den internen Scheduler maßgeblich beeinflusst. Dieser führte auf dem Testsystem nahezu konstant 12,1 Verfügbarkeitsmessungen pro Sekunde durch. Dies entspricht einem Zeitbedarf von 82,7 Millisekunden für jede Messung.

Metadaten-Korrelationsdienst

Ein Profiling des MDCS über 100 Durchläufe ermittelt die zeitliche Verteilung des Datenabrufs über die SPACE-Plattformdienste. Die Ergebnisse der benötigten Zeit pro Durchlauf bezogen auf einen exemplarischen dienst sind in der Tabelle 5.4 zusammengefasst. Als Aussage für die Skalierbarkeit lässt sich eine maximale Obergrenze definieren, bis zu welcher die Zahl der Dienstbeschreibungen, SLA-Vorlagen, SLAs und Monitoringreports zeitnah behandelt werden können. Ausschlaggebend sind dafür die Richtlinien der jeweiligen Plattforminstanz über die Frequenz der Anwendung der Metadatenverfahren. Auffällig ist die sowohl absolut als auch im Vergleich zu den anderen Schnittstellen hohe Abrufzeit des Rating-Dienstes. Es empfiehlt sich daher die Deaktivierung des Einflusses der Benutzerbewertungen auf die Metaqualität in zeitkritischen Umgebungen bis zur Verfügbarkeit einer optimierten Implementierung.

Dienstverzeichnis ConQo

Parallel zu MDCS ist auch das Synchronisationswerkzeug *ConQoNoosferoUpdater* ein Aufrufer des Verzeichnisdienstes ConQo, sofern eine Portalintegration erreicht werden soll. Zum Zweck der Messung ist das Werkzeug isoliert aufgerufen worden. Für 15547 registrierte Dienste benötigte der SOAP-Aufruf zur Rückgabe aller Dienste und ihrer Dokumente 122 Minuten. Der anschließende Synchronisationsvorgang mit der lokalen Datenbank von Noosfero beanspruchte weitere 28 Minuten. Dies entspricht einem Zeitbedarf von 578,9 Millisekunden pro Dienst.

Die sich an den Abruf der Daten durch MDCS anschließende Registrierung von aktualisierten WSML-Dienstbeschreibungen in der ConQo-Registry benötigte für 15547 registrierte Dienste inklusive Aktualisierung einer assoziierten Daten wie WSDL oder WSAG 732 Minuten. Dies entspricht einem Zeitbedarf von 2,82 Sekunden pro Dienst. Dieser erhöhte Zeitbedarf gegenüber einer Abfrage ist mit der Analyse der referenzierten Basis- und Domänenontologien aus WSMO4IoS zu erklären. In der Konsequenz ist es derzeit nicht ratsam, die Propagierung der Metaqualitätsdaten sowie die dynamische Anpassung von NFE häufiger als notwendig durchzuführen. Stattdessen empfiehlt sich bei niedriger Aktivität eine ereignisgebundene Durchführung als Neuberechnung der Metriken für einen Dienst. Bei hoher Aktivität hingegen empfiehlt sich die Zusammenfassung mehrerer Ereignisse in einer zeitlich fixierten Durchführung. Die Richtlinien zu den Umfängen und Zeitpunkten der Metaqualitätsvorgänge bestimmen dabei die DQ-Dimension Aktualität. Über diese in der Dissertation nicht weiter betrachtete Dimension können sich Plattformbetreiber im Wettbewerb voneinander über die Metaqualität hinaus abgrenzen.

Kapitel 6

Zusammenfassung

Eine umfängliche Berücksichtigung der qualitativen Zusammensetzung von beschreibenden Artefakten im Internet der Dienste über dafür ausgelegte Dienstplattformen steigert die Erlebnisqualität für die Benutzer der angebotenen Dienstleistungen. In dieser Dissertation sind die existierenden Verfahren zur Erkennung, Propagierung und Verbesserung von Qualitätsmetriken in dienstorientierten Architekturen in Hinblick auf diese gesteigerten Anforderungen systematisch in Verfahren eingebunden und erweitert worden.

Das konzeptionelle Resultat ist eine feingranulare, phasenübergreifende Berücksichtigung der Metaqualität von Diensten als abstrakte Methodik und architekturgebundener Prozess.

Das praktische Resultat der Arbeit ist eine modulare und offene Plattform für die Bereitstellung und Nutzung von Diensten. Die Skalierbarkeit der Plattform ist in weiten Teilen untersucht und bis auf ausstehende Optimierungspunkte nachgewiesen worden. Die Erweiterbarkeit der Plattform wurde mit einer Umsetzung der Konzepte zur Metaqualität sowie mit einer Anbindung eines Dienstleistungsportals und einer Experimentierumgebung im Rahmen der Validierung belegt.

Zum Abschluss soll das Verwertungspotenzial der Resultate skizziert und ein Ausblick auf konzeptionelle Erweiterungen über die in dieser Arbeit explizit festgelegten Einschränkungen hinaus gegeben werden.

6.1 Verwertungspotenzial

Die in dieser Dissertation erbrachten Konzeptions- und Umsetzungsarbeiten lassen sich sowohl im akademischen als auch im wirtschaftlichen Bereich verwerten. Aufgrund der derzeitigen Dynamik der Entwicklung des Internets der Dienste wird der Beitrag der Dissertation hierfür skizziert.

Auf wirtschaftlicher Seite stellt das im Rahmen der Validierungsarbeiten dieser Dissertation entstandene Dienstportal Crowdserving eines der ersten Webangebote dar, auf dem sowohl technische als auch nichttechnische Dienstleistungen reeller Unternehmen gesucht und genutzt werden können. Die Basisdienstplattform SPACE als zugrundeliegende Infrastruktur wird von verschiedenen Internetdiensteanbietern (ISPs) auf einen praktischen Einsatz überprüft. Eine Instanz der Plattform ist zudem im THESEUS-Innovationszentrum in Berlin installiert

und zur intuitiven Erfassung der operativen und qualitativen Aspekte handelbarer Dienste öffentlich nutzbar ¹.

Auf akademischer Seite steht mit dem Dienstportal Servomat ein interaktives und kollaboratives Werkzeug zur qualitätsfokussierten Bereitstellung von Diensten zur Verfügung. Neben systemseitigen Überprüfungen können Studenten und Betreuer über das System kommentieren und Verbesserungen vorschlagen. Mit den Experimentiererweiterungen zur SPACE-Plattform, speziell dem Open-Access-Repository, dem Simulationsagenten SPACEagent und dem Metadaten-Korrelationsdienst, lassen sich zukünftige Forschungsfragen im Internet der Dienste wesentlich einfacher experimentell lösen. Schließlich dient die modulare Struktur der SPACE-Plattform anderen Forschungsprojekten als Technologielieferant. Dies wurde bereits in der Nutzung der Discovery-Komponente ConQo im Projekt CRUISe erprobt ². Hierbei werden Dienste zur Repräsentation und Auslieferung komponierbarer Benutzerschnittstellen (*UI Services*) mit Hilfe der deklarativen XML-Sprache UISDL registriert. Die darin enthaltenen nichtfunktionalen Eigenschaften können mit dem in dieser Dissertation entwickelten SAL sowohl direkt als auch indirekt über eine Abbildung auf die jeder Registrierung zugrunde liegenden WSMML verarbeitet und qualitativ bemessen werden.

6.2 Ausblick

Im Verlauf dieser Arbeit sind bewusst Einschränkungen vorgenommen worden, um einen zeitlich und thematisch eng gesetzten Rahmen nicht zu überschreiten. Eine Priorisierung und Ergänzung der erforschten Konzepte und umgesetzten Prototypen wird für Folgearbeiten empfohlen. Ein wissenschaftlich verwertbares Potenzial wird in diesen Erweiterungen vermutet:

- Wechselwirkung zwischen kurzfristig zur Laufzeit eines Dienstes durchgeführten Adaptionenmechanismen und Berechnung der Metaqualität. Hierbei könnte eine konzeptionell getrennte Angabe der mit und ohne Adaptionenmechanismen garantierbaren Qualitätsstufen auf Basis annotierbarer nichtfunktionaler Eigenschaften erfolgen. Dies erlaubt im Sinne der wirtschaftlich flexiblen Nutzung von Diensten eine getrennte Preisgestaltung und Abrechnung für komplexe Adaptionenverfahren als Bestandteil von Dienstbündeln.
- Verfeinerte Berechnung der Metaqualität unter Berücksichtigung des zeitlichen Verlaufs, verfügbarer Kontextdaten und der Eingabedaten zu Dienstaufufen. Mit dieser Erweiterung entstünde die Möglichkeit, Störeinflüsse wie temporäre Überlastungen oder fehlerhafte Plattformbestandteile zu berücksichtigen sowie die Qualität anspruchsvoller Dienstbeschreibungsformate wie CQML+ detaillierter zu überprüfen.
- Verteilter Abgleich von Metaqualitätsinformationen, die aus möglicherweise unterschiedlichen Implementierungen der Berechnungsalgorithmen, mindestens jedoch aus unterschiedlichen Systemen resultieren. Über eine

¹THESEUS-Innovationszentrum: <http://www.theseus-programm.de/innovationszentrum/>

²Forschungsprojekt CRUISe: <http://www.cruise-projekt.de/>

Zusammenführung und einen Abgleich der Daten können sich Drittanbieter als Metaqualitätsprüfer etablieren und die Dienstanutzer deren Ergebnisse mit den weiteren verfügbaren vergleichen. Eine Automatisierung des Vergleichs ließe sogar zu, bei starken Abweichungen möglicherweise fehlerhafte Metaqualitätsquellen zu erkennen und auszuschließen.

Literaturverzeichnis

- [1] E. Acar and B. Yener. Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1):6–20, January 2009.
- [2] S. Agarwal. *Formal Description of Web Services for Expressive Matchmaking*. PhD thesis, Universität Karlsruhe (TH), May 2007.
- [3] E. Al-Masri. The QWS Dataset, 2007.
- [4] X. Bai, W. Dong, W.-T. Tsai, and Y. Chen. WSDL-Based Automatic Test Case Generation for Web Services Testing. In *Proceedings of the IEEE International Workshop on Service-Oriented System Engineering (SOSE)*, pages 215–220, October 2005. Beijing, China.
- [5] C. Bartolini, A. Bertolino, E. Marchetti, and A. Polini. Towards Automated WSDL-Based Testing of Web Services. In *6th International Conference on Service-Oriented Computing (ICSOC)*, pages 524–529, December 2008. Sydney, Australia.
- [6] J. Belizki, S. Costache, and W. Nejdl. Application Independent Metadata Generation. In *Proceedings of the ACM 15th CIKM (Conference on Information and Knowledge Management) International ACM Workshop on Contextualized Attention Metadata*, 2006. Airlington, Virginia, USA.
- [7] J. Bell, N. Snooke, and C. Price. Functional decomposition for interpretation of model based simulation. In *Proceedings of 19th International Workshop on Qualitative Reasoning*, pages 192–198, May 2005.
- [8] R. Berbner. *Dienstgüteunterstützung für Service-orientierte Workflows*. PhD thesis, Technische Universität Darmstadt, 2007.
- [9] N. Bhatti and S. H. Weber. *Handbook of Research on Social Dimensions of Semantic Technologies and Web Services*, chapter Semantic Visualization to Support Knowledge Discovery in Multi- Relational Service Communities. Information Science Reference, May 2009.
- [10] G. J. Brahnmath, R. R. Raje, A. M. Olson, M. Auguston, B. R. Bryant, and C. C. Burt. A Quality of Service Catalog for Software Components. In *Proceedings of the Southeastern Software Engineering Conference*, pages 513–520, April 2002. Huntsville, Alabama, USA.

- [11] I. Brandic, S. Venugopal, M. Mattess, and R. Buyya. Towards a Meta-Negotiation Architecture for SLA-Aware Grid Services. Technical Report GRIDS-TR-2008-10, The University of Melbourne, Australia, August 2008.
- [12] I. Braun, S. Reichert, J. Spillner, A. Strunk, and A. Schill. Zusicherung nichtfunktionaler Eigenschaften und Dienstgüte im Future Internet of Services. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 31(04/08):225–231, 2008.
- [13] Y. Brun, G. D. M. Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, and M. Shaw. *Engineering Self-Adaptive Systems through Feedback Loops*, volume 5525 of *LNCS*, pages 48–70. Springer-Verlag, 2009.
- [14] H.-J. Bullinger, K.-P. Fähnrich, and T. Meiren. Service Engineering – Methodical Development of New Service Products. *International Journal of Production Economics*, 85(3):275–287, September 2003.
- [15] M. Bundgaard, A. J. Glenstrup, T. Hildebrandt, and E. Højsgaard. Formalizing WS-BPEL and Higher Order Mobile Embedded Business Processes in the Bigraphical Programming Languages (BPL) Tool. Technical report, IT University of Copenhagen, May 2008.
- [16] S. Cai, Y. Zou, B. Xie, and W. Shao. Leveraging Robust Service Evaluation by Introducing the Web of Trust. In *2009 IEEE International Conference on Cloud Computing (CLOUD)*, pages 190–197, September 2009. Bangalore, India.
- [17] J. Cardoso, M. Winkler, and K. Voigt. A Service Description Language for the Internet of Services. In *Proceedings First International Symposium on Services Science (ISSS)*, March 2009. Leipzig, Germany.
- [18] S. Casolari, M. Andreolini, and M. Colajanni. Runtime prediction models for Web-based system resources. In *16th IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems (MASCOTS)*, pages 1–8, September 2008. Baltimore, Maryland - USA.
- [19] J. Cateano, P. Santos, P. Justino, L. W. Goix, P. Renditore, M. Demartini, P. Falcarin, R. Martín, A. Martínez, R. Fernández, C. Baladrón, J. Aguiar, and B. Carro. Introducing the user to the service creation world: concepts for user centric service creation, personalization and notification. In *16th IST Mobile and Wireless Communications Summit*, July 2007. Budapest, Hungary.
- [20] D. Chalmers, N. Dulay, and M. Sloman. Towards Reasoning About Context in the Presence of Uncertainty. In *1st International Workshop on Advanced Context Modelling, Reasoning and Management*, September 2004. Nottingham, UK.
- [21] V. Christophides, R. Hull, G. Karvounarakis, A. Kumar, G. Tong, and M. Xiong. Beyond Discrete E-Services: Composing Session-oriented Services in Telecommunications. In *Proceedings of the Second International*

- Workshop on Technologies for E-Services (TES)*, pages 58–73, September 2001. Rome, Italy.
- [22] M. Comerio, F. D. Paoli, M. Palmonari, and I. Toma. Flexible Service Offering with Semantic Policies. In *Proceedings of the 2nd Non-Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC)*, November 2008. Dublin, Ireland.
- [23] P. Corte and D. Debora. NEXOF Reference Architecture - RA Specification 1.0. Technical report, NESSI, November 2009.
- [24] B. Coskun and B. Sankur. Robust Video Hash Extraction. In *Proceedings of the European Conference On Signal Processing (EUSIPCO)*, 2004. Wien, Österreich.
- [25] C. Dovrolis and P. Ramanathan. A Case for Relative Differentiated Services and the Proportional Differentiation Model. *IEEE Network*, 13(5):26–34, September-October 1999.
- [26] J. Eckert. *Cross-organizational Service-based Workflows - Solution Strategies for Quality of Service Optimization*. PhD thesis, TU Darmstadt, Fachgebiet Multimedia-Kommunikation, October 2009.
- [27] J. P. (ed.). RFC 739: Transmission Control Protocol (TCP), September 1981.
- [28] A. A. et al. Web Services Agreement Specification (WS-Agreement), 2007. Grid Resource Allocation Agreement Protocol (GRAAP) WG.
- [29] H. L. et al. Web Service Language Agreement (WSLA) Language Specification, 2003. IBM Corporation.
- [30] W. T. et al. SLA@SOI Framework Architecture. Technical report, SLA@SOI, June 2009.
- [31] M. Feldmann, G. Hübsch, C. Liebing, and A. Schill. Anwendung dienstorientierter Ansätze im Kontext der Erstellung von Benutzungsschnittstellen für serviceorientierte Applikationen. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 33(02/10), 2010.
- [32] M. Feldmann, G. Hübsch, T. Springer, and A. Schill. Improving Task-driven Software Development Approaches for Creating Service-based Interactive Applications by using Annotated Web Services. In *5th International Conference on Next Generation Web Services Practices*, November 2009. Prague, Czech Republic.
- [33] S. Fisher. The Architecture of the Apex Platform, salesforce.com’s Platform for Building On-Demand Applications. In *Companion to the proceedings of the 29th International Conference on Software Engineering*, 2007. Minneapolis, Minnesota, USA.
- [34] J. Fisseler, G. Kern-Isberner, C. Beierle, A. Koch, and C. Müller. *Algebraic Knowledge Discovery Using Haskell*, pages 80–93. 2007.

- [35] P. G. S. Florissi. *QoSME: QoS Management Environment*. PhD thesis, Columbia University, New York City, USA, 1996.
- [36] A. U. Frank. Data Quality—What Can an Ontological Analysis Contribute? In *8th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Spatial Accuracy)*, June 2008. Schanghai, China.
- [37] G. Frankova, D. Malfatti, and M. Aiello. Semantics and Extensions of WS-Agreement. *Journal of Software*, 1(1):23–31, 2006.
- [38] T. Friedrich. Fusion von Web Services mit lokalen Diensten. Master’s thesis, TU Dresden, April 2008.
- [39] E. Goldoni and M. Schivi. End-to-End Available Bandwidth Estimation Tools, An Experimental Comparison. In *Proceedings of Traffic Monitoring and Analysis Workshop (TAM)*, April 2010. Zürich, Schweiz.
- [40] O. Günther, G. Tamm, and F. Leymann. Pricing web services. *International Journal of Business Process Integration and Management*, 2(2):132–140, 2007.
- [41] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. WSMX - A Semantic Service-Oriented Architecture. In *Proceedings of the International Conference on Web Service (ICWS)*, 2005. Orlando, Florida - USA.
- [42] L. Hunter, M. Bada, K. Cohen, H. Johnson, W. B. Jr., and P. Ogren. Ontology Quality Metrics, October 2007.
- [43] G. Hübsch and C. Liebing. A Description Language For User Interface Services. In *To be published at IADIS Intl. Conference WWW/Internet*, October 2010. Timisoara, Romania.
- [44] O. Hühn, C. Markl, and M. Bichler. On the predictive performance of queueing network models for large-scale distributed transaction processing systems. *Information Technology and Management (ITM)*, 10(2-3):135–149, September 2009.
- [45] Iris Braun and Matthias Wauer and Sandro Reichert and Josef Spillner and Anja Strunk and Alexander Schill. A Monitoring and Adaptation Architecture for the Internet of Services. In *Proceedings of WWW/Internet*, November 2009. Rome, Italy.
- [46] S. Jagannathan and J. Vitek. Optimistic Concurrency Semantics for Transactions in Coordination Languages. In *International Conference on Coordination Models and Languages (Coordination)*, pages 183–198, February 2004. Pisa, Italy.
- [47] Jane Greenberg and Kristina Spurgin and Abe Crystal. Final Report for the AMeGA (Automatic Metadata Generation Applications) Project. Technical report, University of North Carolina at Chapel Hill, 2005.
- [48] C. Janiesch, M. Niemann, and N. Repp. Towards a Service Governance Framework for the Internet of Services. In *Proceedings of the 17th European Conference on Information Systems (ECIS)*, June 2009.

- [49] J. M. Jähnert, S. Wesner, and V. A. Villagrà. The Akogrimo Mobile Grid Reference Architecture - Overview, 2002. Akogrimo Deliverable D3.1.3.
- [50] A. Keller and H. Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Technical Report RC22456, IBM Research, May 2002.
- [51] L. Kemper. *Systemansatz eines kollaborativen und wissensintensiven Dienstleistungsmanagements: Knowledge Intensive Service Management System (KISMS)*. PhD thesis, Universität Paderborn, August 2007.
- [52] H. Kett, G. Scheithauer, N. Weiner, and A. Weisbecker. Integrated Service Engineering (ISE) for Service Ecosystems: An Interdisciplinary Methodology for the Internet of Services. In *eChallenges e-2009 Conference Proceedings*, October 2009. Istanbul, Turkey.
- [53] M. Kirsch-Pinheiro, Y. Vanrompay, and Y. Berbers. Context-Aware Service Selection Using Graph Matching. In *Proceedings of the 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop*, November 2008. Dublin, Ireland.
- [54] A. Klein. *Datenqualität in Sensordatenströmen*. PhD thesis, Technische Universität Dresden, Dresden, Germany, 2009.
- [55] M. M. Kong and D. Truong. DCE Directory Services. *Hewlett-Packard Journal*, December 1995.
- [56] J. Krogstie, O. I. Lindland, and G. Sindre. Defining Quality Aspects for Conceptual Models. In *Proceedings of IFIP8.1 working conference on Information Systems Concepts: Towards a Consolidation of Views*, pages 216–231, 1995. Marburg, Germany.
- [57] D. Kuroпка and M. Weske. Implementing a Semantic Service Provision Platform — Concepts and Experiences. *Special Issue on Service Oriented Architectures and Web Services of Journal Wirtschaftsinformatik*, 1(1):16–24, 2008.
- [58] S. Köhn. Mass Production of Web Services. Master’s thesis, TU Dresden, March 2010.
- [59] S. König, S. Hudert, and T. Eymann. Socio-Economic Mechanisms to Coordinate the Internet of Services: The Simulation Environment SimIS. *Journal of Artificial Societies and Social Simulation*, 13(2):6, 2010.
- [60] M. Lacage, M. Ferrari, M. Hansen, and T. Turetletti. NEPI: Using Independent Simulators, Emulators, and Testbeds for Easy Experimentation. In *4th Workshop on Real Overlays and Distributed Systems (ROADS)*, October 2009. Big Sky, Montana, USA.
- [61] D. Lamanna, J. Skeene, and W. Emmerich. Specification Language for Service Level Agreements (SLAng), 2003. TAPAS Deliverable D2.

- [62] S. Lehmann and P. Buxmann. Preisstrategien für Software- und Serviceanbieter. In *Schriften zur Quantitativen Betriebswirtschaftslehre*, number 4 in Working Paper, 2008.
- [63] A. Lööf. Internet usage in 2008 – Households and Individuals. European Statistical Report KS-QA-08-046-EN-N, European Community, November 2008.
- [64] V. Marques, R. Aguiar, F. Fontes, J. Jähnert, and H. Einsiedler. Enabling IP QoS in Mobile Environments. In *Mobile Summit*, September 2001. Barcelona, Spain.
- [65] Matthias Winkler and Alexander Schill. Towards Dependency Management in Service Compositions. In *Proceedings of the International Conference on e-Business (ISE-B)*, pages 79–84, 2009. Milan, Italy.
- [66] M. Meyerhöfer and K. Meyer-Wegener. Estimating Non-functional Properties of Component-based Software Based on Resource Consumption. *Electronic Notes in Theoretical Computer Science*, 114:25–45, 2005.
- [67] G. A. Mihaila, L. Raschid, and M.-E. Vidal. Using Quality of Data Metadata for Source Selection and Ranking. In *ACM SAC, Special Track on WebDB*, pages 93–98, 2000.
- [68] G. D. Modica, V. Regalbuto, O. Tomarchio, and L. Vita. Enabling renegotiations of SLA by extending the WS-Agreement specification. In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, pages 248–251, July 2007. Salt Lake City, Utah, USA.
- [69] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li. QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments. *IEEE Communications Magazine*, 39(11):140–148, November 2001.
- [70] Y. Nakagome and H. Mori. Flexible Routing in the Global Communication Network. In *Proceedings of the 7th International Teletraffic Congress (ITC-7)*, pages 426.1–426.8, June 1973. Stockholm, Sweden.
- [71] M. Niemann, C. Janiesch, N. Repp, and R. Steinmetz. Challenges of Governance Approaches for Service-Oriented Architectures. In *Proceedings of the 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, June 2009.
- [72] D. Oberle, N. Bhatti, S. Brockmans, M. Niemann, and C. Janiesch. Effektive Handhabung von Service-Informationen im Internet der Dienste. *Business & Information Systems Engineering (BISE) Journal*, 5, 2009.
- [73] OMG. Handling Non-Functional Properties in a Service-Oriented Architecture: Request for Information, December 2008. OMG Request for Information.
- [74] D. Oppenheimer, A. Ganapathi, and D. A. Patterson. Why do Internet services fail, and what can be done about it? In *Proceedings of the 4th Usenix Symposium on Internet Technologies and Systems (USITS)*, March 2003. Seattle, Washington - USA.

- [75] J. O’Sullivan, D. Edmond, and A. H. M. ter Hofstede. Formal description of non-functional service properties. Technical report, Queensland University of Technology, Brisbane, Australia, 2005.
- [76] M. Ouzzani and A. Bouguettaya. Efficient Access to Web Services. *IEEE Internet Computing*, 8(2):34–44, March-April 2004.
- [77] F. D. Paoli, M. Palmonari, M. Comerio, and A. Maurino. A Meta-Model for Non-Functional Property Descriptions of Web Services. In *IEEE International Conference on Web Services*, September 2008. Beijing, China.
- [78] K. Pichanaharee and T. Senivongse. QoS-Based Service Provision Schemes and Plan Durability in Service Composition. In *Distributed Applications and Interoperable Systems*, volume 5053 of *LNCS*, pages 58–71, May 2008. Oslo, Norway.
- [79] S. Pietschmann, M. Voigt, A. Rümpel, and K. Meißner. CRUISe: Composition of Rich User Interface Services. In *Proceedings of the 9th International Conference on Web Engineering (ICWE)*, pages 473–476, June 2009. San Sebastian, Spain.
- [80] L. L. Pipino, Y. W. Lee, and R. Y. Yang. Data Quality Assessment. *Communications of the ACM*, 45(4):211–218, April 2002.
- [81] S. Preißler, H. Voigt, D. Habich, and W. Lehner. Stream-Based Web Service Invocation. In *13. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW)*, March 2009. Münster, Germany.
- [82] D. Pym, R. Taylor, C. Tofts, M. Yearworth, B. Monahan, and F. Gittler. Systems and services sciences: a rationale and a research agenda. Technical report, HP Labs, 2006.
- [83] A. Raffio and C. Tziviskou. Service Level Agreement Languages, 2005. Unpublished, Politecnico di Milano.
- [84] J. S. Rellermeyer, G. Alonso, and T. Roscoe. R-OSGi: Distributed Applications through Software Modularization. In *Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference Conference (MIDDLEWARE)*, November 2007. Newport Beach, California, USA.
- [85] N. Repp. *Überwachung und Steuerung dienstbasierter Architekturen - Verteilungsstrategien und deren Umsetzung*. PhD thesis, TU Darmstadt, Fachgebiet Multimedia-Kommunikation, July 2009.
- [86] C. Riedl, J. M. Leimeister, and H. Krcmar. New Service Development for Electronic Services – A Literature Review. In *Proceedings of the Fifteenth Americas Conference on Information Systems (AMCIS)*, pages 1–9, August 2009. San Francisco, California, USA.
- [87] C. Riedl, N. May, J. Finzen, S. Stathel, T. Leidig, and R. Belecheau. Managing Service Innovations with an Idea Ontology. In *Proceedings of XIX. International Conference of European Association for Research on Services (RESER)*, September 2009. Budapest, Hungary.

- [88] E. S. Rosenbloom and W. Notz. Statistical Tests of Real-Money versus Play-Money Prediction Markets. *Electronic Markets*, 1(16):63–69, 2006.
- [89] S. Röttger and S. Zschaler. CQML+: Enhancements to CQML. In *Proceedings of 1st Intl. Workshop on QoS in CBSE*, 2003. Toulouse, France.
- [90] S. Röttger and S. Zschaler. Tool Support for Refinement of Non-Functional Specifications. *Software and Systems Modeling*, 6(2):185–204, June 2007.
- [91] G. Scheithauer, K. Voigt, V. Bicer, M. Heinrich, A. Strunk, and M. Winkler. ISE Workbench: Integrated Service Engineering. 7th Business Process Management Conference (BPM) Demo Track, September 2009. Ulm, Germany.
- [92] C. Schroth and T. Janner. Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IEEE IT Professional*, 9(3):36–41, May/June 2007.
- [93] M. Senger, P. Rice, A. Bleasby, and M. Uludag. Soaplab: Open Source Web Services Framework for Bioinformatics Programs. In *The 10th Annual Bioinformatics Open Source Conference*, 2009.
- [94] M. V. Sivaselvan. A Unified View of Hybrid Simulation Algorithms. In *NEES Hybrid Simulation Workshop*, April 2006. Berkeley, California, USA.
- [95] J. Spillner. Privacy-enhanced Service Execution. In *Вестник ДУІКТ: Proceedings of COMINFO - 4th International Conference for Modern Information and Telecommunication Technologies*, September 2008. Livadia, Crimea - Ukraine.
- [96] J. Spillner. Live Media Guide for Demonstrators. Technical Report 000, TU Dresden, October 2009.
- [97] J. Spillner. Service Orientation in Middleware Components for Scalable Service Marketplaces. In *19th International Crimean Conference on Microwave and Telecommunication Technology (КрымТелКо)*, volume 1, pages 360–361, September 2009. Sevastopol, Crimea - Ukraine.
- [98] J. Spillner, I. Braun, and A. Schill. Service Adaptivity through Cross-Domain Reconfiguration of Non-Functional Properties. In *Proceedings of MAI - International DisCoTec Workshop on Middleware-Application Interaction*, Electronic Communications of the EASST, pages 17–22, June 2009. Lisbon, Portugal.
- [99] J. Spillner, I. Braun, and A. Schill. Towards Unified Service Hosting. In *INSTICC-Tagungsband: 4th International Conference on Software and Data Technologies*, volume 2, pages 31–36, July 2009. Sofia, Bulgaria.
- [100] J. Spillner, B. Buder, T. Schiefer, and A. Schill. Contract Services for Post-Discovery Guarantee Management. In *INSTICC-Tagungsband: 4th International Conference on Software and Data Technologies/Special Session on ACT4SOC and e-Health Services*, volume 2, pages 369–375, July 2009. Sofia, Bulgaria.

- [101] J. Spillner, A. Caceres, B. Buder, R. Kursawe, L. S. Globa, and A. Schill. Extending Social Networks with Service Delivery Capabilities for User-Centric Service Trading. In *Proceedings of the 10th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, February 2010. Lviv-Slavske, Ukraine.
- [102] J. Spillner, M. Feldmann, I. Braun, T. Springer, and A. Schill. Ad-hoc Usage of Web Services with Dynvoker. In *Towards a Service-Based Internet: Proceedings of ServiceWave*, volume 5377 of *LNCS*, pages 208–219, December 2008. Madrid, Spain.
- [103] J. Spillner and J. Hoyer. SLA-driven Service Marketplace Monitoring with Grand SLAM. In *INSTICC-Tagungsband: 4th International Conference on Software and Data Technologies*, volume 2, pages 71–74, July 2009. Sofia, Bulgaria.
- [104] J. Spillner, A. Kümpel, I. Braun, and A. Schill. Infrastruktur zur experimentellen Evaluierung von Konzepten im Internet der Dienste. In *Workshop Internet der Dienste auf der GI-Jahrestagung*, September 2010. Leipzig, Germany.
- [105] J. Spillner and A. Schill. Dynamic SLA Template Adjustments based on Service Property Monitoring. In *2009 IEEE International Conference on Cloud Computing (CLOUD)*, pages 183–189, September 2009. Bangalore, India.
- [106] J. Spillner and A. Schill. The TECJIA Service Platform: Web Service Sharing based on Modular Platform Services. In *Proceedings of FIS - 2nd Future Internet Symposium*, September 2009. Berlin, Germany.
- [107] J. Spillner, M. Winkler, S. Reichert, J. Cardoso, and A. Schill. Distributed Contracting and Monitoring in the Internet of Services. In *Proceedings of DAIS: IFIP International Conference on Distributed Applications and Interoperable Systems*, volume 5523 of *LNCS*, pages 129–142, June 2009. Lisbon, Portugal.
- [108] J. Spohrer, P. P. Maglio, J. Bailey, and D. Gruhl. Steps Toward a Science of Service Systems. *Computer*, 40(1):71–77, 2007.
- [109] H. Stanek, N. Smith, and A. Giordano. Modellierung und Normierung von Datenqualität im GIS. In *Symposium für Angewandte Geographische Informationsverarbeitung (AGIT)*, 1995. Salzburg, Österreich.
- [110] V. Stantchev and C. Schröpfer. Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing. In *Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing (GPC)*, volume 5529 of *LNCS*, pages 25–35, May 2009. Geneva, Switzerland.
- [111] N. Steinmetz, H. Lausen, and M. Kammerlander. Crawling Research Report - Version 1. Deliverable D2.1, Service-Finder Project, October 2008.
- [112] G. Stoyanova, B. Buder, A. Strunk, and I. Braun. ConQo – A Context- and QoS-Aware Service Discovery. In *Proceedings of IADIS Intl. Conference WWW/Internet*, October 2008. Freiburg, Germany.

- [113] D. M. Strong, Y. W. Lee, and R. Y. Wang. Data Quality in Context. *Communications of the ACM*, 40(5):103–110, May 1997.
- [114] A. Strunk, I. Braun, S. Reichert, and A. Schill. Supporting Rebinding in BPEL Processes . In *In Proceedings of the 7th International Conference on Web Services (ICWS)*, 8 2009. Los Angeles, California, USA.
- [115] D. Sánchez, D. Isern, and M. Millan. Content annotation for the semantic web: an automatic web-based approach. *Knowledge and Information Systems*, May 2010.
- [116] K. Tserpes, D. Kyriazis, A. Menychtas, T. Varvarigou, F. Silvestri, and D. Laforenza. An Open Architecture for QoS Information in Business Grids. In *Towards Next Generation Grids: Proceedings of the CoreGRID Symposium*, pages 37–49, August 2007. Rennes, Frankreich.
- [117] W. van der Aalst. Pi calculus versus Petri nets: Let us eat “humble pie” rather than further inflate the “Pi hype”. *Business Process Trends*, 3(5), May 2005.
- [118] A. van Moorsel. Metrics for the Internet Age: Quality of Experience and Quality of Business. Technical Report HPL-2001-179, HP Labs, 2001.
- [119] K. Vidackovic, I. Kellner, and J. Donald. Business-Oriented Development Methodology for Complex Event Processing - Demonstration of an Integrated Approach for Process Monitoring. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems (DEBS)*, July 2010. Cambridge, United Kingdom.
- [120] L.-H. Vu and K. Aberer. Towards Probabilistic Estimating Quality of Online Services. Technical report, Swiss Federal Institute of Technology Lausanne (EPFL), 2009.
- [121] R. Wacker, C. Baumann, and C. Funk. Begutachtung des Contract Wizard - Deliverable D.TEXO.2.3.3: Report on Legal and Regulatory Aspects, January 2010.
- [122] F. Wilson. What is the meta-quality of your library? *SCONUL Focus*, 1(38):85–88, 2006.
- [123] M. Winkler and J. Spillner. SLA Management and contract-based Service Execution. In *Proceedings of ICSOC/ServiceWave Demo Session*, December 2009. Stockholm, Sweden.
- [124] M. Winkler and T. Springer. SLA Management for the Internet of Services. In *INSTICC-Tagungsband: 4th International Conference on Software and Data Technologies*, volume 2, July 2009. Sofia, Bulgaria.
- [125] T. Xu, K. Qian, and X. He. Service Oriented Dynamic Decoupling Metrics, 2006. SWWS.
- [126] Y. Yao, F. Yang, and S. Su. Flexible Decision Making in Web Services Negotiation, September 2006. Varna, Bulgaria.

- [127] H. Q. Yu and S. Reiff-Marganiec. Non-functional Property based service selection: A survey and classification of approaches. In *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC)*, November 2008. Dublin, Ireland.
- [128] S. Zschaler. Formal Specification of Non-functional Properties of Component-Based Software Systems. *Software and Systems Modeling*, onlinefirst, February 2009.
- [129] M. zur Muehlen, M. Indulska, and K. Kittel. Towards Integrated Modeling of Business Processes and Business Rules. In *19th Australasian Conference on Information Systems*, December 2008. Christchurch, New Zealand.
- [130] M. zur Muehlen, J. V. Nickerson, and K. D. Swenson. Developing Web Services Choreography Standards – The Case of REST vs. SOAP. *Decision Support Systems*, 40(1):9–29, July 2005.
- [131] J. Øyvind Aagedal. *Quality of Service Support in Development of Distributed Systems*. PhD thesis, University of Oslo, Faculty of Mathematics and Natural Sciences, Department of Informatics, March 2001.

Anhang A

Ausgewählte Publikationen

Die in dieser Dissertation enthaltenen Forschungsergebnisse decken sich punktuell mit publizierten Ergebnissen, die in der Folge durch Diskussionen und die weitere Entwicklung und Evaluierung verfeinert worden sind. Eine Auswahl dieser eigenen Publikationen soll zum besseren Überblick nach Kapiteln und Themenblöcken geordnet an dieser Stelle aufgeführt werden.

Die erarbeiteten Konzepte und Verfahren zur Metaqualität in Kapitel 3 sind bisher nicht publiziert worden. Die Dissertation stellt somit die Erstveröffentlichungsform dar. Eine baldige Einreichung zur vertiefenden Darstellung zentraler Thesen wird angestrebt.

Die Architekturkonzepte, Komponenten und Methoden der Dienstplattform SPACE in Kapitel 4 sind bereits durch Publikationen in Fachzeitschriften und auf Konferenzen belegt worden. Diese sollen an dieser Stelle als thematisch geordnete Auswahl noch einmal aufgeführt werden.

- Suche nach Diensten und vertragliche Bindung
 - *Contract Services for Post-Discovery Guarantee Management*. J. Spillner, B. Buder, T. Schiefer, A. Schill. 4th International Conference on Software and Data Technologies/Special Session on ACT4SOC. Sofia, Bulgarien, Juli 2009. INSTICC-Tagungsband Bd. 2, S. 369-375.
- Überwachung und Anpassung der vertragsgebundenen Dienstauführung
 - *Distributed Contracting and Monitoring in the Internet of Services*. J. Spillner, M. Winkler, S. Reichert, J. Cardoso, A. Schill. 9th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS) / 4th International Federated Conferences on Distributed Computing Techniques (DisCoTec). Lissabon, Portugal, Juni 2009. Springer LNCS 5523, S. 129-142.
 - *SLA-driven Service Marketplace Monitoring with Grand SLAM*. J. Spillner, J. Hoyer. 4th International Conference on Software and Data Technologies (ICSOFT). Sofia, Bulgarien, Juli 2009. INSTICC-Tagungsband Bd. 2, S. 71-74.
 - *Service Adaptivity through Cross-Domain Reconfiguration of Non-Functional Properties*. J. Spillner, I. Braun, A. Schill. 3rd International DisCoTec Workshop on Middleware-Application Interaction

(MAI). Lissabon, Portugal, Juni 2009. Electronic Communications of the EASST, S. 17-22 / ACM International Conference Proceeding Series, S. 13-18.

- *Zusicherung nichtfunktionaler Eigenschaften und Dienstgüte im Future Internet of Services*. I. Braun, S. Reichert, J. Spillner, A. Strunk, A. Schill. PIK - Praxis der Informationsverarbeitung und Kommunikation 31 (4), 2008, S. 225-231.

- Dienstbeschreibungen und Vertragsvorlagen als dynamisierte Artefakte

- *Dynamic SLA Template Adjustments based on Service Property Monitoring*. J. Spillner, A. Schill. 1st IEEE International Conference on Cloud Computing (CLOUD). Bangalore, Indien, September 2009. CLOUD-Tagungsband, S. 183-189.

Die auf der SPACE-Plattform aufbauende Experimentierumgebung für das Internet der Dienste in Kapitel 5 ist durch weitere Einreichungen thematisiert worden.

- *Infrastruktur zur Experimentellen Evaluierung von Konzepten im Internet der Dienste*. J. Spillner, A. Kümpel, I. Braun, A. Schill. Angenommen zum GI-Workshop Internet der Dienste (IDD). Leipzig, September 2010.

Anhang B

Experimentierumgebung

Aufgrund der Natur verteilter, nur teilweise kontrollierbarer Systeme unterliegen die im Rahmen dieser Arbeit entwickelten Software-Prototypen und durchgeführten Experimente mit resultierenden Primärdaten einer besonderen Sorgfalt bezüglich ihrer langfristigen externen Validierbarkeit und Nachvollziehbarkeit. In diesem Abschnitt werden dazu grundlegende Eigenschaften der physischen Experimentierumgebung sowie Versionierungs- und Archivierungsinformationen bereitgestellt.

Die Software der Basisdienstplattform SPACE (vgl. Abschnitte 4.2 und 4.3) steht unter Open-Source-Konditionen (Lizenz GPLv3 und kompatibel) zur Verfügung. Die langfristige Sicherung des Zugriffs und der Weiterentwicklung der Software wird durch die Nutzung eines verteilten Versionskontrollsystems (Git) realisiert.

- Primär-Repository: <http://serviceplatform.org/code/>
- Sekundär-Repository: <http://gitorious.org/spaceserviceplatform>
- Versionsstand: 03. Juni 2010, Branch: `stable-spaceflight-1.0beta3`
- SHA-1-Commit-ID: `b71ac445a20d7a02d930914fb89e303ce8851394`

In ähnlicher Form wird die in Abschnitt 4.4 vorgestellte Software zur dienstlebenszyklusgebundenen Verarbeitung der Metaqualitäts-Metriken als SPACE-Erweiterung unter dem Namen *Cross-Format Non-Functional Properties (Xnfp) Tools* zur Verfügung gestellt. Darin sind der Metadaten-Korrelationsdienst (MD-CS) sowie die von diesem Plattformdienst genutzte Bibliothek Service Abstraction Layer (SAL) enthalten.

- Repository: <http://gitorious.org/xnfp>
- Versionsstand: 18. Juni 2010, Branch: `master`
- SHA-1-Commit-ID: `5b044dc7baf114a61370f2fa00b6440355fa3d42`

Die Experimentierumgebung inklusive der SPACE-Erweiterungen zur Durchführung der Experimente und der clientseitigen Simulationsagenten (vgl. Abschnitt 5.2) stehen über das Projekt *Internet of Services Simulators* zur Verfügung.

- Repository: <http://gitorious.org/ios-simulators>
- Versionsstand: 30. Juni 2010, Branch: `master`
- SHA-1-Commit-ID: 840c223f17e74102220eaf4f6491d75bbe100a05

Schließlich ist ein Open-Access-Zugriff auf die während der Experimente und Simulationen (vgl. Abschnitt 5.3) gewonnenen und generierten Datenbestände über die Protokolle Git und HTTP möglich.

- Primär-Repository: <http://data.serviceplatform.org/>

Die eingesetzten Systeme setzen sich aus physischen und virtuellen Servern zusammen.

- Das Crowdserving-Portal läuft als virtueller Linux-Server mit SPACE-Komponenten auf einem Xeon-E5405-Prozessor und 2 GB RAM. Als Datenbank kommt MySQL 5.1 für SPACE und SQLite 3 für Noosfero zum Einsatz.
- Die für die Experimente verwendete Replika ist ein Core2-Dualprozessorsystem mit 2200 MHz Taktfrequenz unter SPACEflight 1.0 beta3 mit 2 GB RAM.
- Angeschlossene Ausführungsumgebungen ohne Replikation: Das MPWS-System zum Hosting der Dienste über SOAPlab läuft als virtueller Linux-Server mit Opteron-2224-Dualprozessor und 2 GB RAM. Der Betrieb der Flugbuchungs-Web-Services erfolgt über einen PentiumIII-Prozessor mit 800 MHz Takt und 512 MB RAM unter Linux.