

Eine dienstorientierte Methodik zur Koppelung von Netz-QoS-Architekturen

Dissertation

an der

**Fakultät für Mathematik, Informatik und Statistik
der
Ludwig-Maximilians-Universität München**

vorgelegt von

Harald Rölle

Tag der Einreichung: 1. Juli 2005

Tag der mündlichen Prüfung: 28. Juli 2005

1. Berichterstatter: **Professor Dr. Heinz-Gerd Hegering**, Universität München
2. Berichterstatter: **Professor Dr. Gabrijela Dreo Rodosek**, Universität der
Bundeswehr München

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

ISBN 3-89963-215-X

© Verlag Dr. Hut, München 2005.
Sternstr. 18, 80538 München
Tel.: 089/66060798
www.dr.hut-verlag.de

Die Informationen in diesem Buch wurden mit großer Sorgfalt erarbeitet. Dennoch können Fehler, insbesondere bei der Beschreibung des Gefahrenpotentials von Versuchen, nicht vollständig ausgeschlossen werden. Verlag, Autoren und ggf. Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene fehlerhafte Angaben und deren Folgen.

Alle Rechte, auch die des auszugsweisen Nachdrucks, der Vervielfältigung und Verbreitung in besonderen Verfahren wie fotomechanischer Nachdruck, Fotokopie, Mikrokopie, elektronische Datenaufzeichnung einschließlich Speicherung und Übertragung auf weitere Datenträger sowie Übersetzung in andere Sprachen, behält sich der Verlag vor.

1. Auflage 2005

Druck und Bindung: printy, München (www.printy.de)

*Für alle, deren Weg sich nicht ausschließlich
durch den geringsten Widerstand definiert.*

Danksagung

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl von Prof. Dr. Heinz-Gerd Hegering an der Lehr- und Forschungseinheit „Kommunikationssysteme und Systemprogrammierung“ des Instituts für Informatik der Ludwig-Maximilians-Universität in München.

Mein ganz besonderer Dank gilt meinem Doktorvater und Chef, Prof. Dr. Heinz-Gerd Hegering. Für seine Rolle als Doktorvater bedanke ich mich für die hervorragende und stets vorbildliche Betreuung während der Erstellung der Arbeit. Nicht nur seine Anleitung zum wissenschaftlichen Arbeiten, besonders auch seine immer konstruktive Kritik waren maßgeblich am Gelingen der Arbeit beteiligt. In seiner Rolle als Chef hat mir einen zeitlichen und organisatorischen Rahmen geschaffen, der es mir überhaupt erst ermöglicht hat, diese Arbeit zu erstellen. Ganz herzlich bedanken möchte ich mich auch bei meiner Zweitgutachterin, Prof. Dr. Gabrijela Dreo Rodosek, für ihre äußerst hilfreichen Anmerkungen, die die Qualität der vorliegenden Arbeit gesichert und erhöht haben. Bei beiden Gutachtern möchte ich mich für ihren persönlichen Einsatz und besonders für die zügige Bearbeitung von Vorversionen der Arbeit bedanken.

Danken möchte ich ferner allen früheren und aktuellen Kolleginnen und Kollegen des MNM-Teams. Das Wissen, die Diskussionen, die Streitkultur und die Ablenkungen, die ich durch sie erfahren durfte, waren der Boden, auf dem diese Arbeit gedeihen konnte.

Ein spezieller Dank gebührt meinem Freund Niv. Wäre er nicht zur genau richtigen Zeit für mich da gewesen, hätte ich wahrscheinlich nie Gelegenheit bekommen, diese Arbeit zu verfassen.

Nicht zuletzt bin ich meinen Eltern zu großem Dank verpflichtet. Ihre bedingungslose Unterstützung ebnete mir Zeit meines Lebens alle Wege, die ich beschritten habe.

Last but not least geht mein tiefempfundener Dank an Manuela. Wie so häufig, steht auch hinter dieser Arbeit eine starke Frau. Ohne ihren Rückhalt und ihre Leidenschaft wäre diese Arbeit nicht möglich gewesen.

Zusammenfassend möchte ich es mit den Worten einer meiner Vorgänger halten und mich bei allen bedanken, die mir weit mehr beigebracht haben, als ich zur Erstellung dieser Arbeit benötigt hätte.

München, im Juni 2005

ZUSAMMENFASSUNG

Seit einiger Zeit werden verteilte IT-Anwendungen entwickelt, die dezidierte Anforderungen an die Qualität der Ende-zu-Ende-Verbindung zwischen ihren Teilnehmern stellen (z.B. Voice-over-IP, Videokonferenzsysteme etc.). Damit wird die Funktionsfähigkeit solcher Applikationen maßgeblich durch die Beschaffenheit von Netz-Infrastrukturen bestimmt, auf denen sie aufbauen. Die von einem Netz bereitgestellten Dienste müssen also gewissen Qualitätsansprüchen genügen, sie müssen eine Dienstgüte bzw. Quality-of-Service (QoS) anbieten und durchsetzen. Hierzu wurden und werden unterschiedlichste Architekturen entworfen, die QoS auf Ebene eines Netzes realisieren. Während diese Architekturen in homogenen Netzen einzelner Betreiber ihre Funktionsfähigkeit prinzipiell gezeigt haben, stellt sich auf Ebene der Netzverbände ein vollständig anderes Bild dar. Betreiberübergreifend werden heute keine Netzdienste mit QoS angeboten, nicht zuletzt aufgrund des Drucks von Seiten der Anwendungen wird dies aber für die Zukunft erwartet. Da weiterhin nicht abzusehen ist, dass sich global eine gemeinsame Netz-QoS-Architektur durchsetzen wird, müssen Möglichkeiten gefunden werden, wie die Heterogenität im Netzverbund gehandhabt werden kann.

Betrachtet man bestehende Arbeiten, so fällt auf, dass diese größtenteils auf die Betrachtung spezifischer Architekturen oder bestimmter Dienste begrenzt sind. Eine universelle Koppelungsarchitektur im Sinne einer für jeden Fall unmittelbar anwendbaren Lösung wurde bisher nicht entwickelt. Die Komplexität der Fragestellung und der Breite des Spektrums denkbarer Netz-QoS-Architekturen lassen eine Existenz einer solchen allgemeinen Koppelungsarchitektur auch fraglich erscheinen.

Die Dissertation verfolgt einen strikt dienstorientierten Ansatz, der sich in einem Satz als „Generischer Weg zur Gewinnung einer spezifischen Lösung“ zusammenfassen lässt. Dabei werden im Wesentlichen zwei Hauptschritte durchgeführt. Ausgangspunkt des ersten Schritts sind die zu koppelnden Netz-QoS-Dienste. Sie werden durch Modellierung in eine normalisierte Darstellung überführt. Dabei werden unter Betrachtung statischer und dynamischer Aspekte zuerst die den Diensten zugrunde liegenden Netz-QoS-Architekturen modelliert. Hierauf basierend werden Modelle für Netz-QoS-Dienste erstellt. Durch die Normalisierung ist die Grundlage gelegt, um im zweiten Schritt mittels der dann entworfenen generischen Methodik die fallspezifische Lösung gewinnen zu können. Dabei werden, neben der Identifizierung der zur Koppelung notwendigen architekturellen Bausteine, auch die für die Koppelung benötigten Arbeitsabläufe bestimmt. Sowohl Modellierung als auch die Methodik sind an den Gesichtspunkten der Dienstorientierung ausgerichtet, speziell wird in beiden Teilen der gesamte Lebenszyklus eines Dienstes betrachtet.

Illustriert werden Modellierung und Methodik an verschiedenen Beispielen, die ihren Ursprung in realen Szenarios haben. Die Betrachtung unterschiedlicher geschäftlicher Anwendungsfälle (wie z.B. Ausschreibung, Aufwandsabschätzung etc.) rundet die Arbeit ab.

ABSTRACT

In recent years, distributed IT applications were developed which pose explicit requirements on the quality of end-to-end connections between the participating parties. Examples include voice-over-IP, video conferencing etc. Hence, those applications' functionality is significantly influenced by the underlying network infrastructures on which applications build upon. Services offered by networks must therefore meet certain criteria of quality – they have to provide and enforce quality-of-service (QoS). For this reason, different architectures have been and still are being developed to realize QoS on the network level. While such architectures have proven to be functional in principle for homogeneous networks of single providers, the situation is completely different when it comes to interconnecting autonomous or/and heterogeneous networks. Today, no QoS-enabled network services are offered which span multiple provider domains. But this is expected and necessary for the future, last but not least because of the pressure imposed by the application level. As there is no universal QoS architecture on the horizon to become accepted globally, ways have to be found to handle heterogeneity in a network of networks.

When reviewing existing work in the field, it becomes evident that most approaches are limited to specific architectures or certain services. What is still missing is a universal internetworking architecture which could be applied in any situation and to any QoS architecture. The existence of such a generic architecture seems to be uncertain regarding the complexity of the task and the broad spectrum of QoS architectures which have to be covered.

This thesis strictly follows a service-oriented approach which can be summarized as “a generic way for gaining a specific solution”. The approach consists of two major steps. The starting point of the first step are the QoS network services which are to be coupled as well as the QoS architectures these services are built upon. Initially, a normalized representation of the architectures is gained by modeling. In doing so, static and dynamic aspects of QoS architectures are considered. Afterwards, the modeling of QoS network services itself takes place and is based on the models for architectures. The whole modeling process creates a normalized starting point for the second major step. It consists of a generic methodology for gaining specific solutions to solve the coupling problem for individual cases. In this step the necessary architectural building blocks and the required workflows for the coupling are identified. The modeling as well as the methodology are strictly service-oriented, especially in considering the whole service life cycle for both parts.

The generic model and the methodology are illustrated by a series of examples which are derived from real world scenarios. The thesis' results are supported by an analysis of different business use cases, e.g. bid invitation, estimation of effort etc.

INHALT

1	Einführung	1
1.1	Fokus und Fragestellung der Arbeit	2
1.2	Vorgehensmodell und Ergebnisse der Arbeit	6
2	Problembeschreibung, Ansatz und Anforderungsanalyse	9
2.1	Szenarios für die Koppelung von Netz-QoS-Architekturen	12
2.1.1	Voice-over-IP-Koppelung von Standorten	12
2.1.2	Koppelung zwischen IP Zugangs- und Kernnetzprovider	14
2.1.3	Providerhierarchien in Netzverbänden	16
2.2	Einordnung von Netz-QoS-Architekturen	19
2.2.1	Netz-QoS-Architektur	19
2.2.2	Basistechnologien	24
2.2.3	Einordnung in das Dienstumfeld	24
2.2.4	Netz-QoS-Dienst	27
2.2.5	Netz-QoS-Dienstinstanz	27
2.2.6	Zusammenfassung der Einordnung	27
2.3	Dimensionen der Koppelungsproblematik	29
2.3.1	Dimension Architekturteilbereiche	29
2.3.2	Dimension Anwendungsfall	30
2.3.3	Dimension Providerhierarchie	31
2.3.4	Dimension Garantietyp	32
2.3.5	Dimension Lebenszyklus	32
2.3.6	Dimension Dynamik	32
2.3.7	Dimension Freiheitsgrade	33
2.4	Anforderungen und Lösungsansätze	34
2.4.1	Allgemeine Anforderungen an einen Lösungsansatz	34
2.4.2	Auswahl des prinzipiellen Lösungsansatzes	36
2.4.2.1	Individuallösungen	36
2.4.2.2	Umbrella-Koppelung	37
2.4.2.3	Lösungsansatz dieser Arbeit	39
2.4.3	Anforderungen an dienstorientiertes Modell	40
2.4.4	Anforderungen an Methodik	42
2.5	Zusammenfassung von Problembeschreibung und Anforderungsanalyse	44

3	Status quo der Koppelung von Netz-QoS-Architekturen	45
3.1	Koppelungsverfahren bestehender Netz-QoS-Architekturen	46
3.1.1	Asynchronous Transfer Mode (ATM)	46
3.1.2	Integrated Services (IntServ)	47
3.1.3	Differentiated Services (DiffServ)	49
3.1.4	Universal Mobile Telecommunications System (UMTS)	50
3.2	Forschungsansätze	52
3.2.1	Schmitt – Heterogenous Network Quality of Service Systems	52
3.2.2	Spezifische Ansätze	55
3.3	Großprojekte	57
3.3.1	Projekt TEQUILLA	57
3.3.2	Projekt QUASAR	58
3.4	Zusammenfassung und benachbarte Gebiete	60
3.4.1	Basistechnologie MPLS	61
3.4.2	Spezifikation von QoS	64
4	Modellierung von Netz-QoS-Diensten	67
4.1	Lösungsidee und Vorgehen	68
4.2	Folgen der Dienstorientierung	72
4.2.1	Dienstorientierung anhand des MNM-Dienstmodells	72
4.2.1.1	Seitenunabhängiger Teil	72
4.2.1.2	Dienstnehmerdomäne	75
4.2.1.3	Dienstleisterdomäne	75
4.2.1.4	Kernaussagen der Dienstsicht	76
4.2.2	Reduktion auf Domänensicht	76
4.2.3	Zusammenfassung und Fokussierung der Modellierung	80
4.3	Modell für Netz-QoS-Architekturen	82
4.3.1	Statisches Grundmodell für Netz-QoS-Architekturen	82
4.3.1.1	Modellierungstechnik mit UML	84
4.3.1.2	Daten-Kategorie	85
4.3.1.3	QoS-Parameter	90
4.3.1.4	QoS-Kategorie	97
4.3.1.5	Session	102
4.3.1.6	Vollständiges statisches Grundmodell	102
4.3.2	Lebenszyklus bei Netz-QoS-Architekturen	106
4.3.2.1	Lebenszyklusphase Standardisierung	107
4.3.2.2	Lebenszyklusphase Planung	107
4.3.2.3	Lebenszyklusphase Vereinbarung	107
4.3.2.4	Lebenszyklusphase Bereitstellung	108
4.3.2.5	Lebenszyklusphase Nutzung	108
4.3.2.6	Lebenszyklusphase Änderung	108
4.3.2.7	Lebenszyklusphase Beendigung	109
4.3.2.8	Zeitpunkte im Lebenszyklus	109
4.3.3	Kardinalitätsanalyse im statischen Grundmodell	110
4.3.3.1	Auffaltungstransformation	113

4.3.3.2	Kardinalitäten der Session-Assoziation	116
4.3.4	Dynamisches Modell durch Lebenszyklusanalyse	117
4.3.4.1	Variabilitätsanalyse und Grundvorgänge	117
4.3.4.2	Lebensabschnittskorridore der Grundvorgänge	119
4.3.4.3	Ereignisse in den Korridoren	120
4.3.5	Zusammenfassung des Modells für Netz-QoS-Architekturen	123
4.4	Modell für Netz-QoS-Dienste	124
4.4.1	Individualisierung des Architektur-Modells	124
4.4.1.1	Anpassungen am statischen Grundmodell	124
4.4.1.2	Anpassungen der dynamischen Modellteile	125
4.4.2	Einbettung in das MNM-Dienstmodell	126
4.4.2.1	Nutzungsspezifische Teile	126
4.4.2.2	Managementspezifische Teile	130
4.4.2.3	Das Service Agreement	130
4.5	Zusammenfassung der Modelle	131
4.5.1	Zwischenabgleich mit Anforderungen	131
5	Generische Koppelungsmethodik für Netz-QoS-Dienste	135
5.1	Methodik zur Modellierung	137
5.1.1	Erstellung eines Netz-QoS-Architektur-Modells	138
5.1.2	Erstellung eines Netz-QoS-Dienst-Modells	139
5.2	Basisvergleich und Grundlagen	141
5.2.1	Grundeigenschaften des Anwendungsfalls	141
5.2.2	Namensschema und Notationsstruktur	144
5.2.3	Generische Aktivitätsmuster der Koppelung	146
5.2.3.1	Aktivitätsmuster zur Ermittlung optimaler Paarungen	146
5.2.3.2	Aktivitätsmuster für Koppelungssymmetrie	147
5.2.4	Grundproblem zustandsloser und zustandsbehafteter Entitäten	149
5.2.4.1	Zustandslos / -behaftet bei Transitiv-Koppelung	149
5.2.4.2	Zustandslos / -behaftet bei Tunnel-Koppelung	151
5.3	Vergleich der statischen Modellteile	153
5.3.1	Gegenüberstellung statischer Teile von Daten-Kategorien	153
5.3.1.1	Attribut Verbindungsorientiert	155
5.3.1.2	Klasse Kommunikationsrelation	159
5.3.1.3	Attribut Paketgrenze_erhaltend	167
5.3.2	Gegenüberstellung statischer Teile von QoS-Kategorien	172
5.3.2.1	Klasse QoS-Parameter	173
5.3.2.2	Klasse Semantik	175
5.3.2.3	Klasse Garantietyp	176
5.3.2.4	Klasse Wertemenge	178
5.3.2.5	Behandlung ungepaarter QoS-Parameter	185
5.4	Vergleich der dynamischen Modellteile	188
5.4.1	Anpassung der Kardinalitäten	188
5.4.2	Abstimmung von Ereignissen	195
5.5	Abschließende Schritte der Koppelung	198
5.5.1	Vervollständigung der Dienstkoppelung	199

INHALT

5.5.1.1	Koppelung der Service Access Points	199
5.5.1.2	Koppelung der Managementfunktionalität	199
5.5.2	Realisierungsentscheidung	200
5.5.3	Projektplanung und Dienstvereinbarung	201
5.5.3.1	Abarbeitung relevanter Teillösungen	201
5.5.3.2	Aufteilung der Umsetzung	202
5.5.3.3	Entwurf einer Dienstvereinbarung	203
5.5.4	Implementierung der Koppelung	203
5.6	Zusammenfassung und Überblick über die Koppelungsmethodik	204
5.6.1	Abgleich mit den Anforderungen	206
6	Anwendungsbereiche von Modellierung und Methodik	209
6.1	Ergebnisabschätzung einer Koppelung	210
6.2	Taxonomien für Architekturen, Dienste und Koppelungen	212
6.2.1	Modellierung als Taxonomie	212
6.2.2	Methodik als Taxonomie	213
6.3	Erschließung bestehender Arbeiten	215
6.4	Geschäftliche Anwendungsfälle	218
6.4.1	Anwendungsfall Ausschreibung	218
6.4.2	Anwendungsfall Angebotserstellung	218
6.4.3	Anwendungsfall Angebotsvergleich	219
6.5	Problemfall zentrale Austauschpunkte	221
7	Zusammenfassung und Ausblick	223
A	Statisches Modell der UMTS-QoS-Architektur Revision 6	227
	Abkürzungen	233
	Abbildungsverzeichnis	237
	Tabellenverzeichnis	241
	Literatur	243
	Teilfallindex	257
	Teillösungsindex	259
	Schlagwortindex	261

KAPITEL 1

EINFÜHRUNG

Seit einiger Zeit werden verteilte IT-Anwendungen entwickelt, die dezidierte Anforderungen an die Qualität der Ende-zu-Ende-Verbindung zwischen ihren Teilnehmern stellen. Damit wird die Funktionsfähigkeit solcher Applikationen maßgeblich von der Beschaffenheit von Netz-Infrastrukturen bestimmt, auf denen sie aufbauen.

Ende-zu-Ende
Qualität

Betrachtet man beispielsweise reine Hypertext-Anwendungen wie das klassische HTTP / HTML im World Wide Web, so stellt man fest, dass sie geringe Ansprüche an ein Netz stellen. So ist seine prinzipielle Funktionalität in Netzen unterschiedlichster Bandbreiten gewährleistet. Verzögerungen in der Übertragung bringen zwar Komforteinbußen für den Nutzer mit sich, die Anwendung bleibt aber benutzbar. Multimedia-Applikationen, wie z. B. Videokonferenzsysteme, benötigen für eine korrekte Funktion dagegen ganz bestimmte Qualitätsmerkmale von Seiten des Netzes. Im Fall einer Videokonferenz muss für die gesamte Dauer der Sitzung unter anderem eine hinreichend große Bandbreite für die zeitnahe Übertragung der Videodaten bereitgestellt werden. Weiterhin darf auch die Ende-zu-Ende Verzögerung zwischen allen Teilnehmern ein gewisses Limit nicht überschreiten, da ansonsten z.B. Lippensynchronizität, ruckfreie Filmqualität oder auch interaktive Gespräche nicht mehr möglich sind. Können bei solchen Anwendungen also Voraussetzungen durch ein Netz nicht erfüllt werden, so kann die Applikation ihre Funktionalität nicht erbringen; sie ist nicht oder nur eingeschränkt nutzbar.

Multimedia-
Applikationen als
Beispiel für explizite
Qualitätsanforderungen

Die für Anwendungen von einem Netz bereitgestellten Dienste müssen also gewissen Qualitätsansprüchen genügen, sie müssen eine **Quality-of-Service (QoS)**¹ anbieten und durchsetzen. Hierzu wurden und werden unterschiedlichste Architekturen entworfen, die eine QoS auf Ebene eines Netzes realisieren. Während diese Architekturen in homogenen Netzen einzelner Betreiber ihre Funktionsfähigkeit prinzipiell gezeigt haben, stellt sich auf Ebene der Netzverbände ein vollständig anderes Bild dar. Betreiberübergreifend werden heute keine Netzdienste mit QoS angeboten, nicht zuletzt aufgrund des Drucks von Seiten der Anwendungen wird dies aber für die Zukunft erwartet [[BHPB+ 00](#), [GHSB 03](#)].

Betreiberübergreifend heute kein
QoS

¹Die englische Bezeichnung ist auch in deutschsprachiger Literatur sehr häufig vorzufinden und wird in dieser Arbeit dem Begriff „Dienstgüte“ vorgezogen.

1.1 FOKUS UND FRAGESTELLUNG DER ARBEIT

Fokussierung auf Netzebene

Um die von verteilten IT-Anwendungen gestellten Anforderungen an die Ende-zu-Ende Qualität befriedigen zu können, müssen mehrere Komponenten zusammenspielen und betrachtet werden. Abbildung 1.1 zeigt ein grobes Schichtenmodell. Die vorliegende Arbeit konzentriert sich dabei auf die Schicht der Netze. Entsprechend bleiben Fragestellungen der Ende-zu-Ende-QoS im Betriebssystem und den Anwendungen selbst hier unbetrachtet.

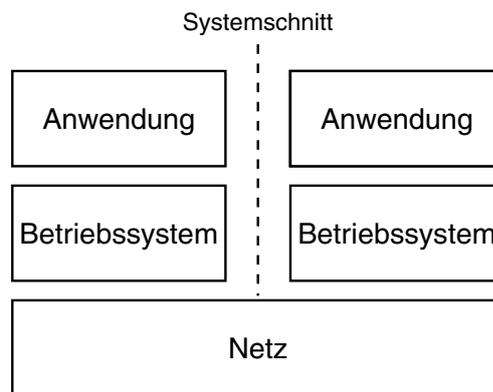


Abbildung 1.1: Grobschichtenmodell verteilter IT-Anwendungen

Koppelproblematik Die Netzebene ist nicht als monolithisches Netz realisiert, sie stellt sich vielmehr i.a. als ein Verbund von Teilnetzen dar, die durch verschiedene Dienstleister betrieben werden. Entsprechend können in den autonomen Teilnetzen unterschiedliche Netz-QoS-Architekturen zum Einsatz kommen. Soll nun auf Netzebene eine Ende-zu-Ende-QoS realisiert werden, stellt sich unmittelbar die Frage, wie die unterschiedlichen Netz-QoS-Architekturen miteinander zu koppeln sind.

Bestehende Arbeiten fokussieren spezifische Netz-QoS-Architekturen

Keine generische Lösung der Koppelungsproblematik

Betrachtet man bestehende Arbeiten auf dem Gebiet der Koppelung von Netz-QoS-Architekturen, so fällt neben anderen Einschränkungen auf, dass diese größtenteils auf die Betrachtung spezifischer Architekturen begrenzt sind (siehe Kap. 3). Eine generische Koppelungsarchitektur im Sinne einer für jeden Fall direkt anwendbaren Lösung wurde bisher nicht entwickelt. Die Komplexität der Fragestellung und die Breite des Spektrums denkbarer Netz-QoS-Architekturen lassen eine Existenz einer solchen allgemeinen Koppelungsarchitektur auch fraglich erscheinen. Entweder würde nämlich diese vereinheitlichte Koppelungsarchitektur selbst zu komplex sein, um realisiert zu werden, oder eine prinzipiell handhabbare Lösung wäre zu generisch, um für reale Koppelungsvorgänge eine sinnvolle Hilfestellung zu geben.

Die vorliegende Arbeit geht im Weiteren von folgenden Grundannahmen aus:

Netz-QoS-Architekturen als solche sind notwendig. Häufig wird das Argument genannt, dass QoS-Architekturen auf Netzebene nicht notwendig seien, wenn Netzkapazitäten weit über das im Mittel benötigte Maß vorhanden sind, um damit auch Spitzenlasten (und damit jegliche Lastsituation) ohne Einschränkungen bewältigen zu können (das sog. **Overprovisioning**). Dabei wird meist nur der Parameter Bandbreite betrachtet. Allerdings lassen sich Parameter wie Verzögerung und Jitter durch Overprovisioning erheblich schwerer in den Griff bekommen. Weiterhin, unabhängig von der Frage, inwiefern Overprovisioning ökonomisch zu verantworten ist, kann es den Anwendungen aber keine Auskunft darüber geben, ob die geforderten Eigenschaften des Netzes, selbst im günstigsten Fall, überhaupt erreichbar sind. Einen anwendungsorientierten Vergleich von Overprovisioning und reservierenden QoS-Ansätzen finden sich beispielsweise in [FHP 00, BBS 98, BrSh 98].

Overprovisioning
nicht ausreichend

Global wird sich keine einheitliche Netz-QoS-Architektur durchsetzen. In Anlehnung an die Durchsetzung des IP-Protokolls als globale Konvergenz von Protokollen der Vermittlungsschicht könnte argumentiert werden, dass für den Bereich der Netz-QoS-Architekturen eine analoge Entwicklung zu erwarten sei. Dementsprechend wäre eine Behandlung der Koppelungsproblematik nicht nachhaltig notwendig. Einer solchen Entwicklung stehen jedoch einige schwerwiegende Argumente entgegen. Neben der reinen Tatsache, dass trotz des Drucks von Seiten der Anwendungen ein Konvergieren der Architekturen nicht stattgefunden hat, haben sich bestehende Architekturen für verschiedene Einsatzschwerpunkte als unterschiedlich geeignet erwiesen (z.B. [RFC 2998, FLNN+ 00]). Auch neu entwickelte Architekturen werden häufig vor dem Hintergrund bestimmter Einsatzszenarios entworfen. Daraus leitet sich auch für die absehbare Zukunft eine zu erwartende parallele Existenz mehrerer Architekturen ab. Bereits bestehende QoS-fähige Installationen (z.B. existierende ATM-Netze) würden aus Gründen des Investitionsschutzes vergleichsweise langsam ersetzt werden [BHPB+ 00, GHSB 03]. Hierfür ist also ein integrativer Ansatz erforderlich. Zuletzt gibt der Fakt, dass weiterhin neue Architekturen entworfen werden, ohne dass diese breiten Einsatz im Markt finden, keinen Anlass zu der Annahme, dass sich eine dieser Architekturen global durchsetzen wird.

Eignung von
Architekturen
szenarioabhängig

Investitionsschutz

Zur Lösung der Koppelungsproblematik ist eine statisch-architekturelle Betrachtung der zum Einsatz kommenden Netz-QoS-Architekturen nicht ausreichend. Für eine Anleitung zur Koppelung müssen auch dynamische Aspekte berücksichtigt werden. Dies bezieht sich vor allem auf die Vorgänge zur Installation der Koppelung als auch auf jene, die während der Nutzung der Koppelung notwendig ablaufen. Wichtige Einflussgrößen hierfür sind z.B. die Beziehungen der beteiligten Entitäten zueinander und die mit der Koppelung verfolgten Ziele und Gesamteigenschaften. Neben der Lösung von eher statischen technisch / architekturellen Problemen ist also auch ei-

Statisch-
architekturelle
Betrachtung nicht
ausreichend

KAPITEL 1: EINFÜHRUNG

Dienstorientierung Die Betrachtung der zur Koppelung notwendigen Vorgänge im Sinne einer Prozessorientierung notwendig. Dies weist somit den Weg zu einer dienstorientierten Sichtweise (z.B. [Dreo 02, GHHK+ 01]) der Koppelungsproblematik.

Generische, dienstorientierte Methodik **Ziel dieser Arbeit** ist es, einen nachhaltigen Beitrag zur Lösung der Koppelungsproblematik von Netz-QoS-Architekturen zu leisten. Lösungsansatz hierzu ist die Spezifikation einer dienstorientierten, generischen Methodik. Ihre Anwendung in konkreten Fällen soll dann unter Berücksichtigung der individuellen Umstände strukturiert durch den Koppelungsprozess führen und bei der Spezifikation der Koppelungsarchitektur behilflich sein.

Teilfragestellungen Zur Umsetzung dieses Ziels ergeben sich die folgenden Teilfragestellungen:

Klassifizierung ■ Wie können Netz-QoS-Architekturen kategorisiert und klassifiziert werden?

Bei der gegebenen Spannbreite von existierenden und möglichen Architekturen muss ein Klassifizierungsschema existieren, anhand dessen Architekturen zueinander in Relation gesetzt und verglichen werden können.

Gesamtfunktionalität ■ Basierend auf der Klassifikation, welche Architekturen können sinnvoll gekoppelt werden und welchen Einschränkungen unterliegt eine Koppelung?

Erfahrungen mit bereits bestehenden Architekturen haben gezeigt, dass der, aus einer Koppelung resultierende, Netzverbund hinsichtlich der QoS-Gesamtfunktionalität deutlich hinter denen der Einzelarchitekturen zurückbleiben kann. Um für einen gegebenen Anwendungsfall entscheiden zu können, ob eine Koppelung sinnvoll ist, gilt es zu klären, inwieweit die geforderten Eigenschaften erhalten werden.

Dienst-Lebenszyklus ■ Welche Rolle spielt der Dienst-Lebenszyklus?

Betrachtet man unterschiedliche Paare von zu koppelnden Architekturen, so haben bisherige Erfahrungen gezeigt, dass hierbei gleiche oder ähnliche Teilschritte zu erledigen sind. Allerdings ist der Zeitpunkt für deren Ausführung bei unterschiedlichen Koppelungspaaren höchst unterschiedlich. Für Netz-QoS-Dienste stellt sich damit die Frage, welche Rolle der Dienstlebenszyklus bezüglich einzelner Teilschritte spielt.

Teilvorgänge und architekturelle Komponenten ■ Welche Schritte und Teilvorgänge sind für eine Koppelung notwendig und welche architekturellen Komponenten werden dabei benötigt?

Die Klärung dieser Frage kennzeichnet einen der Kernteile der vorliegenden Arbeit. Durch die Antwort auf diese Frage ergibt sich die Grundlage zur Entwicklung einer generischen Methodik.

1.1 FOKUS UND FRAGESTELLUNG DER ARBEIT

- **Welchen Einfluss haben Diensttypen der zu koppelnden Architekturen und der Einsatzzweck des Gesamtnetzes?** Einfluss von Diensttypen und Einsatzzweck

Wie bereits festgestellt ist die Eignung konkreter QoS-Architekturen einsatzspezifisch zu bewerten. Entsprechend stellt sich die Frage, welchen Einfluss der Diensttyp spielt, der unter Verwendung einer QoS-Architektur realisiert wird. Weiterhin steht zu vermuten, dass der geplante Einsatzzweck Einflüsse auf die Koppelung ausübt.
- **Wie können eventuell auftretende Mehrdeutigkeiten in der Lösung der Kopplungsproblematik beherrscht werden?** Mehrere Lösungsmöglichkeiten

Die bisherigen Forschungsansätze haben selbst für identische Paare konkreter Architekturen unterschiedliche Lösungen der Kopplungsproblematik hervorgebracht. Demnach muss eine allgemeine Methodik mit der nicht existierenden Eindeutigkeit einer Lösung umgehen können.
- **Wie können bisherige Arbeiten zur Kopplungsproblematik eingeordnet und nutzbar gemacht werden?** Nutzung bestehender Arbeiten

Es wurden bereits eine Reihe von Arbeiten und Projekten zur Netz-QoS-Kopplungsproblematik erstellt. Der Großteil teilt dabei die Einschränkung, dass sie sich auf konkrete Architekturen beschränken und keinen generischen Ansatz liefern. Die dort erworbenen Erkenntnisse können aber auch einen Beitrag zu einer generischen Methodik liefern; entsprechend ist zu klären, wie diese Arbeiten nutzbar gemacht werden können.

1.2 VORGEHENSMODELL UND ERGEBNISSE DER ARBEIT

Die Bearbeitung der Aufgabenstellung und die Lösung der gestellten Teilfragestellungen wird durch das im Folgenden vorgestellte und in Abbildung 1.2 dargestellte Vorgehensmodell erreicht. Die Pfeile in der Abbildung symbolisieren den Informationsfluss zwischen den einzelnen Bereichen. Beginnt bzw. endet ein Pfeil am Rand eines übergeordneten Bereichs, dann bedeutet das, dass die Information aus allen Unterbereichen stammt bzw. in allen Unterbereichen benötigt wird.

Pfeilsemantik	In Kapitel 2 werden zur Präzisierung der Aufgabenstellung zunächst einige konkrete Szenarios und das grundlegende abstrakte Szenario eingeführt. Anschließend daran werden einige grundlegende Begriffe definiert. Im weiteren werden dann die verschiedenen Dimensionen der Problemstellung konkretisiert. Neben weiteren Quellen werden dazu aus den Szenarios und den Anwendungsfällen allgemeine Anforderungen abgeleitet. Zur Verfeinerung der Anforderungen werden zunächst prinzipielle Lösungsextreme für die Koppelungsproblematik diskutiert und dabei der in dieser Arbeit verfolgte Ansatz vorgestellt und eingeordnet. Abschließend werden die Anforderungen für den verfolgten Ansatz konkretisiert.
Szenarios grundlegende Begriffe Dimensionen der Problemstellung allg. Anforderungen Lösungsextreme konkretisierte Anforderungen	Anhand der ermittelten Anforderungen wird in Kapitel 3 der Status quo bewertet. Dabei werden zunächst bestehende Netz-QoS-Architekturen dahingehend untersucht wie diese mit der Koppelungsproblematik umgehen. Weiterhin werden aktuelle und relevante Forschungsarbeiten untersucht, wie auch Großprojekte im Koppelungsumfeld. Die anschließende Bewertung zeigt dann, dass der Status quo nicht den gestellten Anforderungen genügt. Ergänzend werden einige zur Arbeit benachbarte Gebiete angerissen.
Status quo zur Lösung der Koppe- lungsproblematik	Die folgenden zwei Kapitel befassen sich dann mit der Entwicklung einer generischen Methodik zur Lösung der Koppelungsproblematik.
Aus Dienstorient. folgen Rahmenbe- dingungen Dienstorient. Modell für Architekturen Statische und dynamische Aspekte Modell für Dienste	In Kapitel 4 werden zunächst die Folgen der Dienstorientierung für das untersuchte Umfeld dargestellt und damit die Rahmenbedingungen für die Modellierung geschaffen. Im nächsten Schritt wird dann ein dienstorientiertes Modell für Netz-QoS-Architekturen entwickelt, das den gestellten Anforderungen entspricht. Dabei ist die Unterscheidung zwischen statischen Aspekten und dynamischen Aspekten im Lebenszyklus ein entscheidender Schlüssel, um Gemeinsamkeiten und Unterschiede zwischen Netz-QoS-Architekturen aufzeigen zu können. Aus dem Modell für Architekturen wird im darauffolgenden Schritt ein Modell für Netz-QoS-Dienste erstellt.
Methodiken für Architektur- und Dienstmodelle	Nachdem mit den Modellen für Netz-QoS-Architekturen und Netz-QoS-Dienste eine normalisierende Darstellung realisiert ist, wird in Kapitel 5 die generische Methodik zur Koppelung von Netz-QoS-Diensten vorgestellt. In der ersten Sektion wird eine Methodik zur Gewinnung eines Netz-QoS-Archi-

1.2 VORGEHENSMODELL UND ERGEBNISSE DER ARBEIT

tektur-Modells bzw. eines Dienst-Modells angegeben. Die anschließenden Schritte befassen sich mit der Koppelung an sich. Im Ergebnis können dann mit der gesamten Methodik sowohl konkrete Teilkomponenten als auch konkrete Teilprozesse gefunden werden, die für eine Koppelung bestimmter Netz-QoS-Dienste notwendig sind.

Methodik zur Koppelung

Abschließend werden in Kapitel 6 unterschiedliche Anwendungsgebiete untersucht. Dies gilt sowohl für die Modellierung von Netz-QoS-Architekturen als auch für die entwickelte Koppelungsmethodik. Dabei werden u.a. Möglichkeiten aufgezeigt, wie auf Basis der Methodik bestehende Arbeiten neu erschlossen werden können. Weiterhin werden typische geschäftliche Anwendungsfälle für Modellierung und Methodik diskutiert.

Anwendungsgebiete für Modellierung und Methodiken

Kapitel 7 schließlich fasst die Arbeit zusammen und gibt einen Ausblick auf mögliche weiterführende Arbeiten.

Hinweise zum Layout

In dieser Arbeit werden folgende Kennzeichnungen benutzt:

Englische Begriffe sind *kursiv* gesetzt.

Teile eines Modells sind in *Optima kursiv* gesetzt.

Absätze, die längere Beispiele enthalten, sind beidseitig eingerückt abgedruckt.

„Längere wörtliche Zitate sind beidseitig eingerückt und kursiv gedruckt.“

Am Ende der Arbeit finden sich Verzeichnisse der im Text enthaltenen Abbildungen und Tabellen, ein Abkürzungsverzeichnis, sowie Literaturreferenzen und ein Index.

KAPITEL 1: EINFÜHRUNG

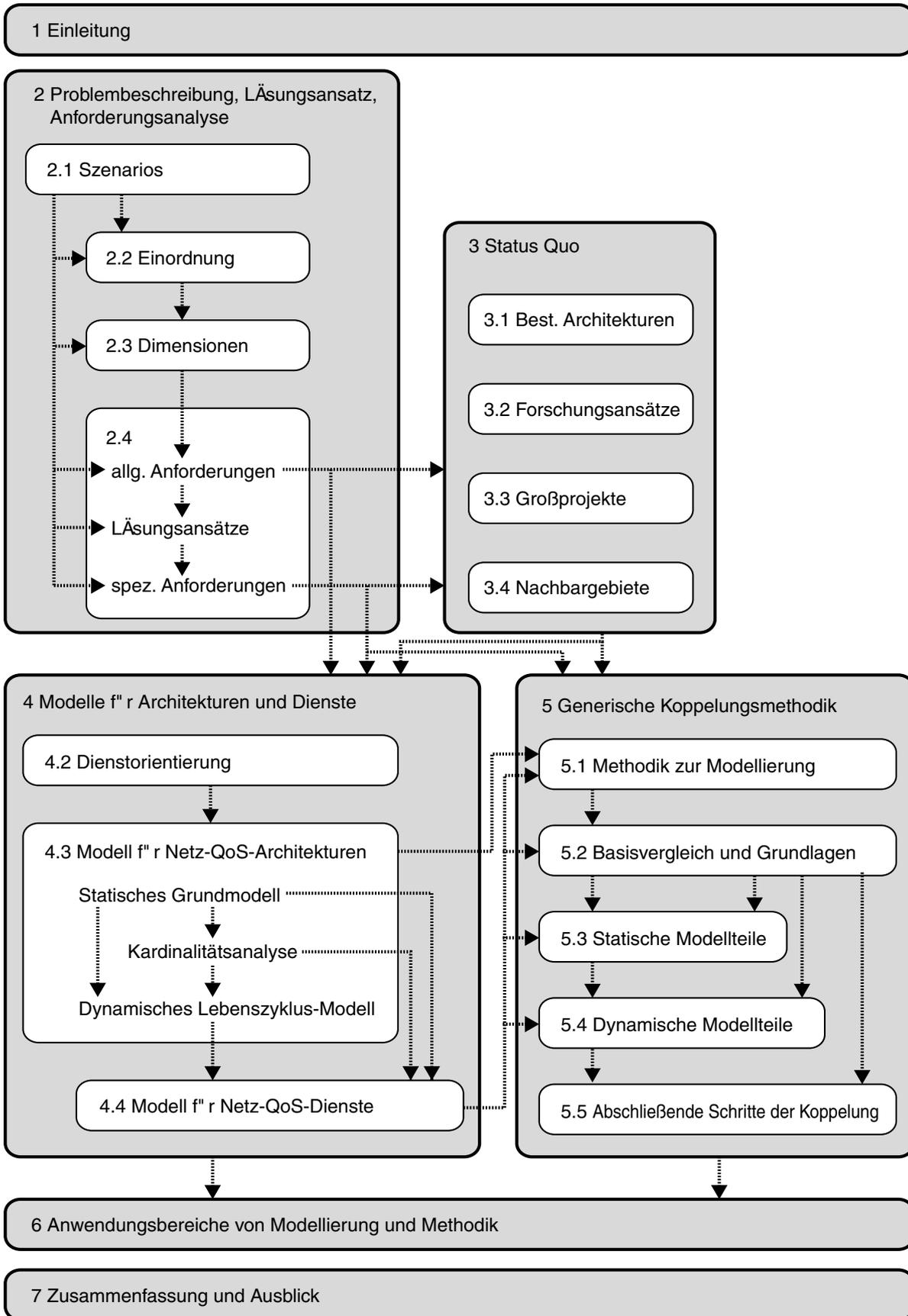


Abbildung 1.2: Vorgehensmodell der Arbeit im Überblick

KAPITEL 2

PROBLEMBESCHREIBUNG, LÖSUNGSANSATZ UND ANFORDERUNGSANALYSE

In diesem Kapitel werden die für diese Arbeit grundlegenden Begriffe definiert und gegen benachbarte Bereiche abgegrenzt und somit der Fokus dieser Arbeit aufgezeigt. Hierzu werden zunächst reale und abstrakte Szenarios vorgestellt, die im Weiteren dann als durchgängiges Beispiel in der Begriffsklärung dienen. Die Koppelungsproblematik wird anschließend anhand ihrer typischen Dimension weiter charakterisiert. Abgeschlossen wird dieses Kapitel mit einer Anforderungsanalyse an die Koppelung von Netz-QoS-Architekturen, die anhand der Szenarios und vor dem Hintergrund des Dienstumfeldes vorgenommen wird. Um dabei die Anforderungen präzise formulieren zu können, werden auch prinzipielle Lösungsmöglichkeiten für die Koppelungsproblematik diskutiert und es wird die für diese Arbeit gewählte Lösungsvariante erstmalig kurz vorgestellt.

Überblick über das Kapitel

Zur Illustration dieses Kapitels werden im Schwerpunkt zwei bekannte Netz-QoS-Architekturen herangezogen. Zum einen ist dies die **Asynchronous Transfer Mode (ATM)**-Technologie, zum anderen die **Differentiated Services (DiffServ)**-Architektur. Eine tiefer gehende Betrachtung der beiden Netz-QoS-Architekturen wird über die Arbeit hinweg erfolgen, zu ihrer groben Charakterisierung werden sie nachfolgend kurz beschrieben.

ATM und DiffServ als Standard-Beispiele

Standardbeispiel 1: Asynchronous Transfer Mode (ATM)

ATM [[ATM-ARCH 02](#)]¹ definiert umfassend eine medienunabhängige Netztechnologie. Die Pflege und Weiterentwicklung von ATM wird von dem Industriekonsortium ATM-Forum [[ATMForum-HP](#)] übernommen, das in enger Kooperation mit internationalen Standardisierungsgremien wie der International Telecommunication Union – Telecom Standardization (ITU-T) [[ITUT-HP](#)] zusammenarbeitet. In dieser Arbeit wird soweit als möglich auf Spezifikationen des ATM-Forums Bezug genommen, da diese eine Vorreiterrolle für die

Asynchronous Transfer Mode (ATM)

¹Das Dokument [[ATM-ARCH 02](#)] ist selbst nur eine Zusammenfassung und ein Wegweiser durch die Vielzahl von Einzeldokumenten in denen ATM spezifiziert ist.

Spezifikationen der ITU-T spielen und somit aktueller sind. Außerdem sind die Spezifikationsdokumente des ATM-Forums zum Zeitpunkt der Drucklegung für den Leser kostenfrei zugänglich, im Gegensatz zu den Standards der ITU-T.

Umfangreiche Spezifikation

ATM setzt Garantien „hart“ durch

Die Spezifikation von ATM erstreckt sich von der Abbildung auf unterschiedlichste Technologien der Bitübertragungsschicht, über ein eigenes Adressierungsschema und eigene Management-Protokolle (z.B. Routing, Verbindungsmanagement etc.), bis hin zur Definition unterschiedlicher Datendienste. Interessant ist ATM für diese Arbeit deshalb, weil für die angebotenen Datendienste explizit und präzise deren Dienstgütemerkmale definiert werden. Fordert ein Anwender solche Dienstgütemerkmale an, bietet die gesamte ATM-Technologie auch alle Möglichkeiten der absoluten Durchsetzung der gewählten Einstellungen. Salopp gesagt ist ATM eine umfangreiche Netztechnologie, die es ermöglicht Dienstgüteanforderungen „hart“ durchzusetzen.

Standardbeispiel 2: Differentiated Services (DiffServ)

Differentiated Services (DiffServ)

DiffServ [[RFC 2474](#), [RFC 2475](#), [RFC 3260](#)] ist ursprünglich eine Erweiterung des IPv4-Protokolls² [[RFC 791](#)] und ist durch eine mittlerweile abgeschlossene IETF³ Working Group⁴ [[DiffServ-WG](#)] im Rahmen des Internet-Standardisierungsprozesses [[RFC 2026](#)] entstanden.

DiffServ bietet nur relative Garantien

In Leistungsumfang und Ausführlichkeit der Spezifikation steht DiffServ im direkten Kontrast zu ATM. Dienstgütemerkmale werden durch DiffServ nicht konkret benannt, vielmehr wird eine relative Unterscheidung zwischen verschiedenen Verkehrsklassen vorgenommen. Verkehrsklassen lassen sich in DiffServ hinsichtlich ihrer zugrunde liegenden Dienstgütemerkmale frei definieren. Allerdings ist die Durchsetzbarkeit der Dienstgüteeigenschaften durch die Basis der IPv4-Architektur so weit eingeschränkt, dass eine absolute Umsetzung von geforderten Parametern prinzipbedingt nicht gewährleistet werden kann. Wiederum salopp ausgedrückt, kann DiffServ keine „harten“, sondern nur relative Garantien bieten.

ATM und DiffServ im Kontrast illustrieren Koppelungsproblematik

Selbstverständlich sind ATM und DiffServ weder die einzig existierenden Netz-QoS-Architekturen, noch werden in dieser Arbeit alleinig sie betrachtet. Allerdings haben beide Architekturen eine gewisse Marktbedeutung erlangt, wodurch ihnen Beachtung geschenkt werden muss. Weiterhin illustrieren die Beispiele ATM und DiffServ aufgrund ihrer Verschiedenheit sehr gut, wie

²Hier wird nur die DiffServ-Variante für das IPv4-Protokoll fokussiert, die IPv6-Variante bleibt unbetrachtet. Dies geschieht ohne Beschränkung der Allgemeinheit, denn für die in diesem Abschnitt gemachten Aussagen kann die IPv6-Variante als strukturell identische gesehen werden.

³Internet Engineering Task Force (IETF)

⁴Die Tatsache, dass die Working Group abgeschlossen ist, besagt nur, dass die ursprünglich gesteckten Ziele erreicht worden sind. Dies hat keinen Einfluss auf die Gültigkeit der erarbeiteten Standards. Diese sind weiterhin gültig.

stark sich Netz-QoS-Architekturen in ihren einzelnen Aspekten unterscheiden können. Deshalb wurden sie für dieses Kapitel als Standard-Beispiele ausgewählt.

2.1 SZENARIOS FÜR DIE KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Szenarios zeigen praktische Relevanz der Koppelproblematik

Beginnend mit der Vorstellung einiger Szenarios wird in diesem Teil der Arbeit die praktische Relevanz der Koppelproblematik aufgezeigt. Später werden die vorgestellten Szenarios auch zur Illustration der präzisierten Problemstellung dienen sowie zur Ableitung von Anforderungen, wie sie an die in dieser Arbeit entwickelten Lösung zu stellen sind.

Reale und abstrakte Szenarios

Die ersten beiden Szenarios werden dabei als real bezeichnet. Sie schildern exemplarische Problemstellungen, wie sie heute in realen Umgebungen vorzufinden sind und für die Aufgabenstellung dieser Arbeit typisch sind. Das dritte, abstrakte Szenario beschreibt anhand des realen Beispiels Internet die typischen Gegebenheiten in Netzverbänden autonomer Dienstanbieter, die eine Grundursache für die Notwendigkeit dieser Arbeit darstellen.

2.1.1 Voice-over-IP-Koppelung von Standorten

In diesem Abschnitt wird ein Szenario vorgestellt, das die Koppelung einer Telefonielösung einer Firma namens ALV AG beschreibt.

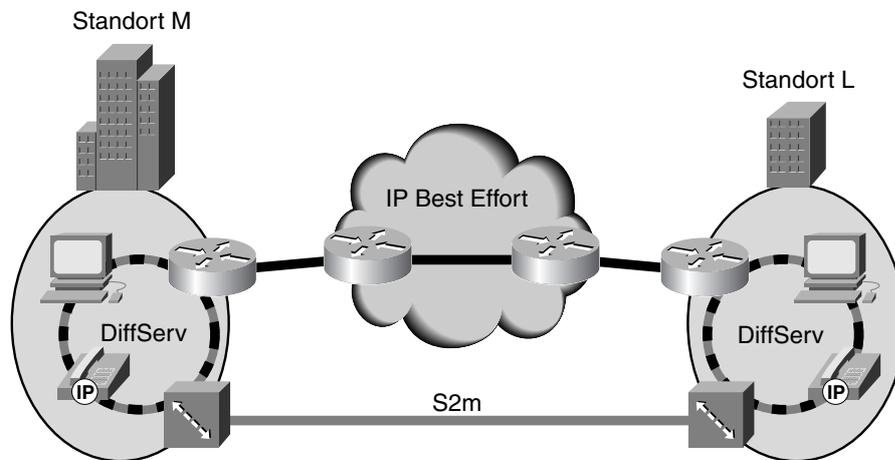


Abbildung 2.1: Standortvernetzung mit separaten Verbindungen

VoIP Koppelung zweier Standorte

Die ALV AG ist ein auf die zwei Standorte M und L verteiltes Unternehmen mit mehreren tausend Mitarbeitern und setzt an jedem Standort **Voice-over-IP (VoIP)** (z.B. [Blac 02, DPG 00]) als Telefonielösung ein. Innerhalb eines jeden Standortes kommt dabei zur Qualitätsabgrenzung zu den Datendiensten die DiffServ-Architektur zum Einsatz. Die beiden Standorte sind bisher über eine leitungsvermittelte Festverbindung in Form eines S2M Primär-multiplexanschlusses [ITU-T I.431] verbunden, der 30 dedizierte ISDN⁵ B-Kanäle bietet. Abbildung 2.1 stellt dieses Szenario dar. Weiterhin sind beide

Zunächst separate Verbindungen

⁵Integrated Services Digital Network (ISDN)

2.1 SZENARIOS FÜR DIE KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Standorte zur Datenkommunikation auf IP-Basis über ein **Virtual-Private-Network (VPN)** vernetzt, das durch die Firma Data-Systems realisiert wird (für Betrachtungen zu VPNs in ähnlichem Umfeld siehe [Sail 02]). Charakteristika des bestehenden VPN sind dabei:

- Vertraulichkeit der Datenkommunikation

Vertraulichkeit

Der Dienstanbieter garantiert dafür, dass die zwischen den Standorten ausgetauschten Daten von keinem Dritten eingesehen werden können.

- Best-Effort IP

Best-Effort IP

Die Kommunikation erfolgt rein über das IP-Protokoll. Es werden keine Garantien über die verfügbare Bandbreite, Verzögerung etc. gegeben.

Aus Kapazitäts- und Kostengründen soll nun die leitungsvermittelte Festverbindung durch eine IP-Strecke ersetzt werden. Während die bisher genutzten Datendienste keine expliziten QoS-Anforderungen stellen, ändert sich dies im Fall von VoIP. Damit die Qualität des Anwendungsdienstes VoIP von seinen Nutzern als wenigstens zufrieden stellend beurteilt wird, müssen bestimmte Voraussetzungen hinsichtlich z.B. Verbindungsaufbauzeit, Verzögerung und Bandbreite erfüllt werden [Thae 01]. Bei einer Zusammenlegung des Verkehrs von Datenkommunikation und Sprachkommunikation müssen also QoS-fähige Dienste eingesetzt werden, um die Qualitätsanforderungen von VoIP gegenüber den Datendiensten durchsetzen zu können.

Jetzt gemeinsame genutzte Verbindung

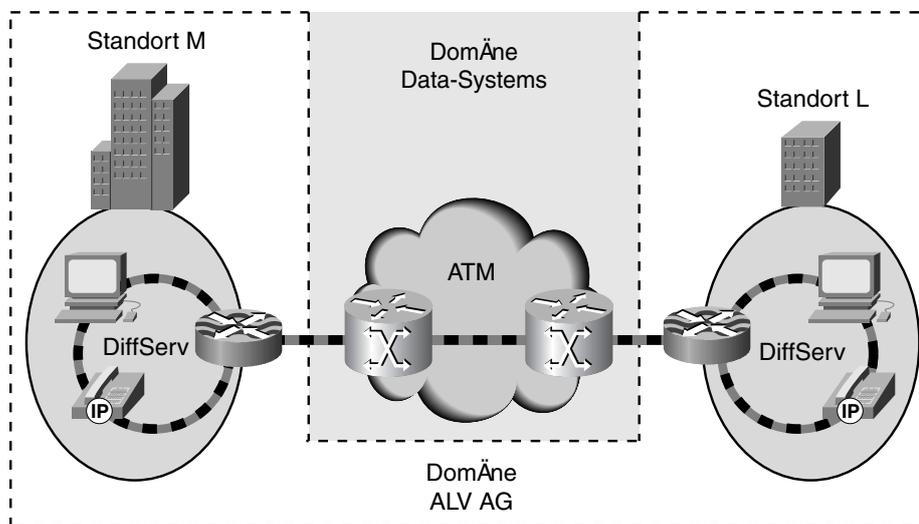


Abbildung 2.2: Standortvernetzung mit gemeinsamen Verbindungen

Der bisherige Dienstanbieter des reinen IP-Datendienstes, die Firma Data-Systems, hat den ATM-basierten IP-Dienst in seinem Produktportfolio. Dieser Dienst würde es ermöglichen harte Garantien bezüglich der für VoIP benötigten Qualitätsmerkmale zu vereinbaren und auch durchzusetzen. Damit könnten Daten nichtprivilegierter Anwendungen und jene von VoIP über den gleichen Dienst zwischen den Standorten ausgetauscht werden. In Abbildung 2.2 ist das geplante Szenario dargestellt. Wollte man diesen Dienst

ATM als Integrationsplattform

nutzen, stellen sich nun einige typische Fragen bezüglich der Koppelung der DiffServ-Architektur an den ATM-basierten Dienst:

Typische Fragen

- | | |
|---|---|
| Qualitätseigenschaften des Gesamtsystems | ■ Kann die DiffServ-Architektur so an den ATM-basierten Dienst gekoppelt werden, dass das Gesamtsystem die erforderlichen Qualitätsmerkmale erfüllt? |
| Übertragung der DiffServ-Markierungen | ■ Kann die Qualitätszuordnung von Daten nach der Übertragung am anderen Standort wieder nachvollzogen werden?
Um ein unternehmensweites IT-Management sinnvoll betreiben zu können, müssen beide Standorte die gleichen DiffServ-Markierungen verwenden. Nach dem Transit durch die Domäne der Data-Systems muss dann am empfangenen Standort die gleiche DiffServ-Markierung wieder hergestellt werden, wie sie am versendenden Standort verwendet wurde. |
| Umsetzung von Qualitätsmerkmalen an Domänengrenze | ■ Wie werden DiffServ-bezogene Qualitätsmerkmale an den Grenzen zwischen der ALV AG und der Data-Systems nach ATM und umgekehrt übersetzt? |
| Zusätzliche technische Komponenten | ■ Welche technischen Komponenten müssen ersetzt bzw. zusätzlich installiert werden? Wo werden sie installiert, wer betreibt sie und wie werden sie betrieben? |
| Managementinteraktionen | ■ Welche (technischen) Managementaktionen sind vor der Betriebsphase notwendig? Welche während des Betriebs? |

Fragestellungen nicht spezifisch für dieses Szenario

Diese und ähnlich Fragen stellen sich allerdings nicht nur im Falle der Koppelung mit ATM, sondern analog auch bei der Koppelung mit einer anderen Architektur. Wiederum sind dann diese Fragen von neuem zu beantworten, da die Lösungen jeweils spezifisch für den jeweiligen Koppelungsfall sind.

Dieser Abschnitt macht deutlich, dass spezifische Probleme in der Koppelung zwischen unterschiedlichen QoS-fähigen Diensten und Architekturen zu lösen sind. Aber selbst bei der Koppelung zwischen Diensten, die auf derselben QoS-Architektur beruhen, werden Fragen aufgeworfen, die nicht trivial zu beantworten sind. Das Szenario im folgenden Abschnitt wird dies illustrieren.

2.1.2 Koppelung zwischen IP Zugangs- und Kernnetzprovider

Videokonferenzdienst in Providerverbund

Ein deutschlandweit operierender Kernnetzprovider bietet einen Videokonferenz-Dienst an. Eine Reihe von Zugangs Providern (Kunden des Kernnetzproviders) haben bereits oder möchten zukünftig diesen Dienst einkaufen, um ihn einerseits selbst nutzen zu können und andererseits ihn an die eigenen Kunden wieder verkaufen zu können.

2.1 SZENARIOS FÜR DIE KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Im Netz des Kernnetzproviders teilen sich Datendienste und der Videokonferenzdienst die gleiche Kommunikationsinfrastruktur. Zur Abgrenzung der Qualitätsanforderungen des Videokonferenzdienstes von den restlichen Datendiensten wird dabei ein einfaches „Gold, Silber, Bronze“-Modell basierend auf Internet DiffServ eingesetzt, wobei der Videokonferenzdienst der „Gold“-Klasse zugeordnet ist.

DiffServ beim Kernnetzprovider

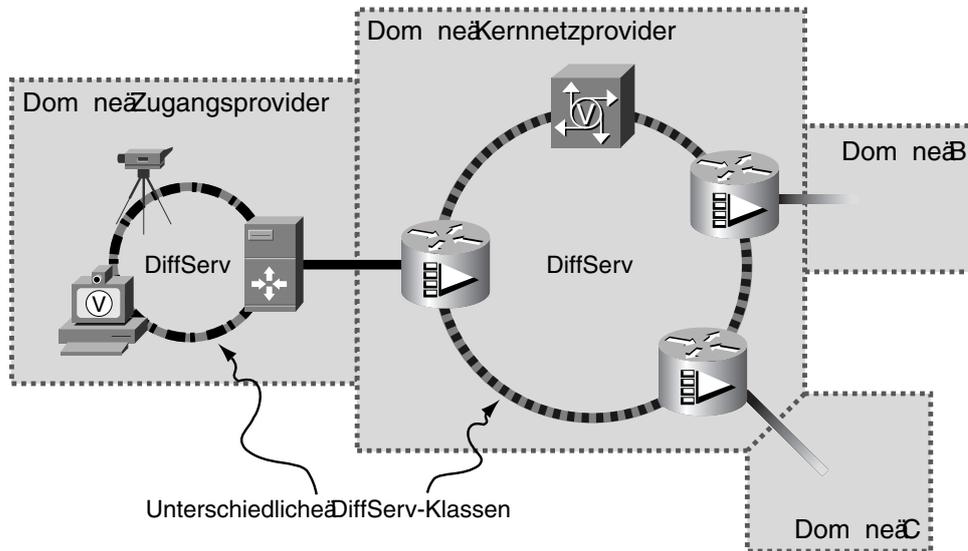


Abbildung 2.3: Ankopplung eines neuen Zugangsproviders

Ein Zugangsprovider möchte nun den Videokonferenzdienst einkaufen. Er setzt in seiner eigenen Domäne ebenfalls bereits DiffServ ein. Allerdings nutzt dieser eine feinere Klassifizierung mit fünf unterschiedlichen Dienstklassen (Abb. 2.3), außerdem verwendet er Router eines anderen Herstellers. Bevor die Koppelung an den Kernnetzprovider vollzogen werden kann, stellen sich hierbei beispielsweise folgende typische Fragen:

Zugangsprovider mit anderen DiffServ-Dienstklassen

Typische Fragen

- Gibt es im Netz des Zugangsproviders bereits eine passende DiffServ-Klasse für den Videokonferenzdienst? Da DiffServ nur die Unterscheidung unterschiedlicher Qualitätsklassen spezifiziert, selbst aber keine solchen Klassen konkret definiert oder vorschreibt, muss zuerst geklärt werden, ob im Netz des Zugangsproviders bereits eine Entsprechung zu den Qualitätsklassen des Videokonferenzdienstes beim Kernnetzprovider existiert.
- Um im Fall von DiffServ äquivalente Qualitätsklassen überhaupt identifizieren zu können, muss zunächst geklärt werden, wie die Klassen definiert sind. Häufig geschieht dies bei DiffServ nämlich nicht explizit über eindeutig definierte Parameter (wie dies bei ATM der Fall ist) sondern nur implizit. Beispielsweise werden Qualitätsklassen häufig definiert, indem Warteschlangen- und Verwerfungseinstellungen verwendet werden, die in den Routern bereits durch ihre Hersteller vordefiniert sind.

Existenz einer geeigneten Qualitätsklasse

Vergleichbarkeit der Qualitätsklassen

Gemeinsame Nutzung einer Qualitätsklasse durch mehrere Dienste	<ul style="list-style-type: none"> ■ Falls eine der DiffServ-Klassen beim Zugangs- oder Kernnetzprovider nicht exklusiv dem Videokonferenzdienst zugeordnet ist, wie soll der Verkehr dieser Dienste im jeweils anderen Netz behandelt werden? Sollen hier ebenfalls die Qualitätseigenschaften gelten oder sollen diese beim Domänenübergang geändert werden? Wenn ja, wie unterscheidet man den Verkehr des Videokonferenzdienstes und worauf bildet man den anderen Verkehr ab? Von wem wird diese Abbildung vorgenommen?
Anpassung der Klassenmarkierungen	<ul style="list-style-type: none"> ■ Wenn zwar äquivalente DiffServ-Klassen in beiden Netzen bestehen, ist allerdings deren (Bit-) Markierung in den IP-Datagrammen nicht zwangsweise identisch. Falls notwendig, wer nimmt die Anpassung der Markierungen beim Domänenübergang vor? Geschieht das für beide Datenrichtungen in der gleichen Domäne oder behandelt jede Domäne eingehende (oder ausgehende) Datenströme selbst?
Klassenänderungen oder neuer Klassen	<ul style="list-style-type: none"> ■ Konnte keine sinnvolle Übereinstimmung bei den QoS-Klassendefinitionen gefunden werden, in welcher Domäne werden dann Änderungen vorgenommen? Wird eine bestehende Klasse modifiziert oder wird eine neue DiffServ-Klasse eingeführt?
Änderungen in Klassendefinition	<ul style="list-style-type: none"> ■ Für den Fall, dass in einer Domäne die Definition einer DiffServ-Klasse verändert wird, wie muss die andere Domäne darauf reagieren?
Koppelungsproblematik selbst bei gleicher Netz-QoS-Architektur	<p>Das vorliegende Beispiel zeigt, dass selbst bei gleicher zugrunde liegender Netz-QoS-Architektur eine Koppelung zweier Netze problematisch sein kann und keineswegs trivial zu realisieren ist.</p> <p>Weiterhin deutet das vorliegende Szenario bereits an, dass unterschiedliche Hierarchiestufen (hier zwischen Zugangs- und Kernnetzprovider) einen Beitrag zur Koppelungsproblematik liefern. Der folgende Abschnitt zeigt anhand des Beispiels Internet-Eigenschaften von Providerhierarchien auf.</p>

2.1.3 Providerhierarchien in Netzverbänden

Internet als typ. Beispiel für Netzverbände	<p>Betrachtet man Netzverbände allgemeiner, so zeigen sich hierarchische Strukturen bei den beteiligten Diensteanbietern. Die im Folgenden anhand des Internets dargestellten Strukturen (beispielsweise nach [CDZ 97]) finden sich dabei analog auch in anderen Netzverbänden. Unterschiede in Benennung und Hierarchietiefe spielen für die in dieser Arbeit entscheidenden strukturellen Aussagen keine Rolle. In Abbildung 2.4 ist eine solche Hierarchie exemplarisch für typische Strukturen im Internet dargestellt.</p>
Benennung der Hierarchiestufen	<p>Ausgehend von der Domäne der Endsysteme sind diese über eine sog. Point-of-Presence (PoP) an das Internet angekoppelt. Ein PoP zeichnet sich durch räumliche Nähe zu seinen Kunden aus, und ist auf ein kleines Gebiet in der Größenordnung von Stadtteilen bis maximal eines Ballungszentrums beschränkt. PoPs wiederum beziehen ihre Konnektivität von einem Zugangsprovider (access provider), der mehrere PoPs bedient. Das Ein-</p>

2.1 SZENARIOS FÜR DIE KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

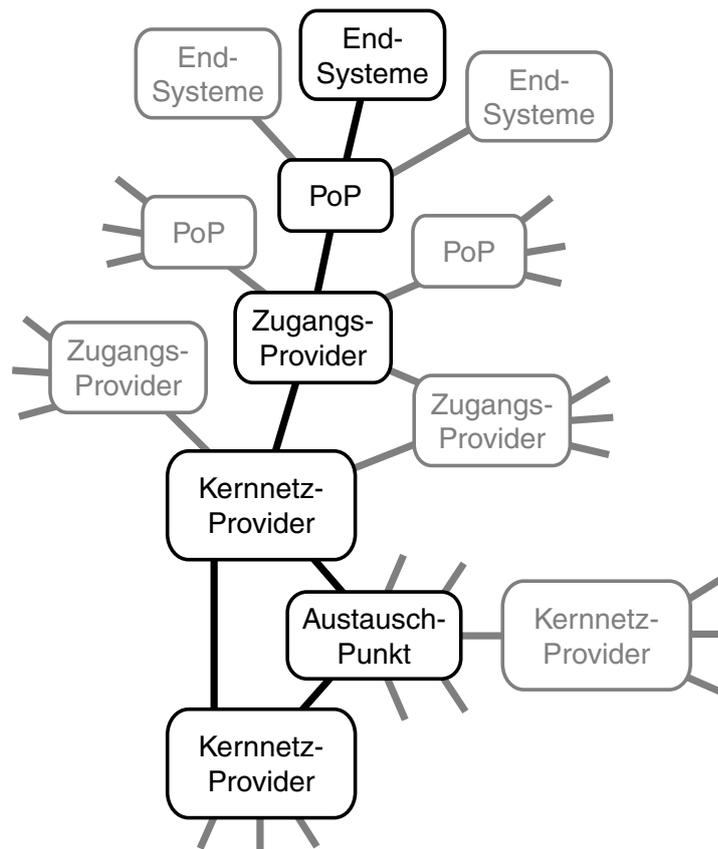


Abbildung 2.4: Exemplarisches Schema eines hierarchischen Netzverbundes

zugsgebiet eines Zugangsproviders reicht typischerweise von Ballungszentren bis zur Ausdehnung auf ein Staatsterritorium. Kontinental bis weltweit operieren **Kernnetzprovider (core provider)**, deren Kunden wiederum die Zugangsprovider sind. Die Kernnetzprovider stellen die Wurzel der Hierarchie dar, sie bilden das Basisnetz (*backbone*) des Netzverbundes.

In der Realität ist diese Hierarchie aber nicht als Baumstruktur im strikten Sinn ausgeprägt. Betrachtet man die Hierarchieschichten von den Endkunden zu den Kernnetz Providern, stellt man fest, dass Vermaschungen auf sowohl gleicher Hierarchiestufe als auch zur unterliegenden zunehmen. In der Regel existieren zwischen Endsystemdomänen keine direkten Verbindungen und meist sind diese jeweils nur an einen PoP angebunden. Gleiches gilt zwischen PoPs und ihren Beziehungen zu den Zugangs Providern. Zwischen Zugangs Providern findet man allerdings auch direkte Verbindungen und ebenso, dass diese an mehrere Kernnetzprovider angebunden sind. Auf der untersten Hierarchiestufe der Kernnetzprovider finden sich neben direkten Verbindungen auch **zentrale Austauschpunkte**.

Zunehmende
Vermaschung
Richtung
Kernnetzprovider

Dienstorientiert betrachtet besteht eine Wertschöpfungskette ausgehend von den Kernnetzen bis hin zu den Endsystemen. Dabei ist jede einzelne der Domänen frei in der technischen Realisierung der Dienste.

Zum Beispiel im Fall des Internets hat sich zwar bzgl. des ISO⁶-OSI⁷ Referenzmodells [ITU-T X.200] auf der Netzebene ein durchgängiges Kommunikationsprotokoll durchgesetzt, nämlich IPv4 [RFC 791]. Andere Schichten aber sind völlig heterogen bei den unterschiedlichen Betreibern realisiert.

Architekturen unterschiedlich gut geeignet
Keine universell geeignete Architektur

Nicht anders stellt sich die Situation für die sinnvoll einsetzbaren QoS-Architekturen dar. Hier haben sich für die unterschiedlichen Ebenen der Providerhierarchie jeweils bestimmte Architekturen als geeignet erwiesen [XiNi 99, VPMK 04]. Keine der heute verfügbaren QoS-Architekturen ist dabei für alle Ebenen und Anforderungen gleich gut geeignet und auch innerhalb einer Ebene gibt es durchaus konkurrierende Ansätze.

Am Beispiel der Internet-Welt zeigt sich dies bereits in deren beiden Hauptansätzen: Integrated Services (IntServ) [RFC 2210] ist demnach wegen seiner Skalierungsproblematik eher für Endsystem- und PoP-Domänen geeignet. DiffServ dagegen ist in unterschiedlicher Eignung auch in anderen Bereichen in der Providerhierarchie anwendbar und konkurriert in den Endsystem- und PoP-Domänen mit IntServ.

Es sei hier nochmals darauf verwiesen, dass selbst für den Fall, dass gleiche Architekturen verwendet würden, die Koppelungsproblematik trotzdem existieren kann. In Abschnitt 2.1.2 wurde dies am Beispiel von DiffServ bereits erläutert.

Heterogenität unausweichlich

Der vorausgegangene einfache Blick auf Providerhierarchien in Netzverbänden macht bereits deutlich, dass man im Bereich der QoS-Architekturen aktuell mit einer heterogenen Landschaft zurechtkommen muss. Vor dem Hintergrund, dass nicht nur aus der Wissenschaft heraus, sondern auch durch Produktentwicklungen neue QoS-Architekturen an Bedeutung gewinnen werden (Mobilfunknetze der dritten Generation sind ein Beispiel hierfür), zeigt sich, dass die Heterogenität auch in Zukunft ein bestimmendes Merkmal sein wird.

⁶International Organization for Standardization (ISO)

⁷Open Systems Interconnection (OSI)

2.2 EINORDNUNG VON NETZ-QoS-ARCHITEKTUREN

Nachdem im vorangegangenen Abschnitt einige konkrete Beispielszenarios die Problemstellung exemplarisch verdeutlicht haben, werden in diesem Abschnitt die zentralen Begriffe der Arbeit eingeordnet und voneinander abgegrenzt.

Vor allem ist hier der Begriff der Netz-QoS-Architektur zu nennen. Er wird in Unterscheidung zu Basistechnologien einerseits und zu Netz-QoS-Diensten andererseits definiert. Dabei wird eine erste, sehr grobe Modellierung vorgenommen, die im Weiteren zu einer ersten Strukturierung dieser Arbeit dient. Im Rahmen der Abgrenzung zum Begriff der Netz-QoS-Dienste wird außerdem auch eine Einordnung in das Dienstumfeld vorgenommen.

Abgrenzung des Architektur-Begriffs

2.2.1 Netz-QoS-Architektur

Betrachtet man die Szenarios aus den Abschnitten 2.1.1 und 2.1.2, so stellt man zunächst fest, dass sowohl auf Seiten der Kunden als auch auf Seiten der Dienstanbieter standardisierte Grundlagen zur Realisierung von Dienstgüte in den eigenen Netzen genutzt werden. Eine solche prinzipiell standardisierte Basis soll hier als **Netz-QoS-Architektur** bezeichnet werden. Im Sinne des Groschichtenmodells von Abbildung 2.5 (vgl. Sekt. 1.1) beschränkt sich dabei eine solche Architektur auf die untersten Schichten der Netze, anwendungs- und betriebssystemspezifische Belange bleiben außen vor.

Keine anwendungs- und betriebssystemspezifischen Belange

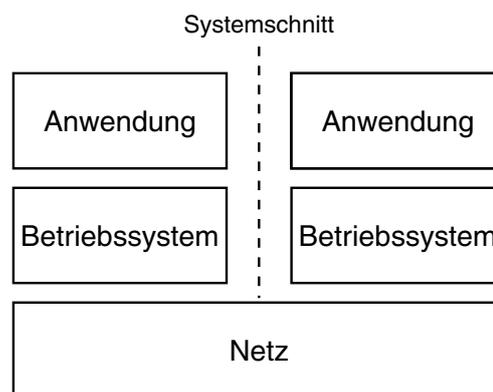


Abbildung 2.5: Groschichtenmodell verteilter IT-Anwendungen

Bekannte Beispiele für Netz-QoS-Architekturen sind die bereits beschriebenen ATM und DiffServ, aber auch das in Abschnitt 2.1.3 erwähnte IntServ. Mobilkommunikationssysteme der dritten Generation, wie z.B. Universal Mobile Telecommunications System (UMTS) [TS 123 107, TS 123 207], bringen ebenfalls eigene QoS-Architekturen mit. In der Aufzählung der prominentesten Beispiele

mag der Leser hier Multi Protocol Label Switching (MPLS) vermissen. Für die Einordnung von MPLS sei auf den nachfolgenden Abschnitt [2.2.2](#) verwiesen.

Im Folgenden werden die für Nutzer einer Netz-QoS-Architektur wesentlichen Merkmale beschrieben. Diese Merkmale dienen auch als Basis für die im Anschluss vorgenommene Grobmodellierung:

Universell
anwendbar

- Die Architektur ist im Sinne der sie nutzenden Anwendungen universell einsetzbar, mindestens ist sie auf eine ganze Klasse von Applikationen anwendbar. Architekturen, die auf genau eine Anwendung hin zugeschnitten sind und dabei auch andere Schichten aus [Abbildung 2.5](#), Seite [19](#) umfassen, fallen nicht darunter.

Datenprotokolle

- Es werden konkrete Protokolle zum Versand und Empfang von Daten spezifiziert, sog. **Datenprotokolle**. Dabei wird auch festgelegt, wie die Zuordnung von QoS-Eigenschaften zu den Daten erfolgt.

DiffServ beispielsweise stützt sich für sein Datenprotokoll vollständig auf IPv4 ab. Für die Zuordnung der QoS-Eigenschaften zu den Daten wird in DiffServ ein Bit-Feld im IPv4 Protokoll-Header verwendet.

QoS-Parameter
unterschiedlich
präzise festgelegt

- Die prinzipiell unterstützten **QoS-Parameter** werden spezifiziert. Dabei variiert der Grad der Spezifikation zwischen unterschiedlichen Architekturen erheblich.

ATM beispielsweise definiert alle verfügbaren QoS-Parameter exakt (z.B. *peak cell rate*, *sustained cell rate*, *maximum cell transfer delay*, *peak to peak cell delay variation*, *cell loss rate* etc., vgl. [\[ATM-TM 99\]](#)).

DiffServ dagegen spezifiziert selbst letztlich überhaupt keine konkreten Parameter [\[RFC 2475, RFC 3140\]](#). Hier übernehmen Zusatzspezifikationen wie z.B. [\[RFC 2597, RFC 3246\]](#) die nähere Bestimmung und Beschreibung der Dienstgütemerkmale. Allerdings ist man nicht verpflichtet, sich auf solche Zusatzspezifikationen abzustützen, man kann alternativ eigene Definitionen verwenden.

QoS-Kategorien
gruppieren QoS-
Parameter

- Weiterhin spezifizieren einige Architekturen sog. **QoS-Kategorien**. Einer solchen Kategorie werden allgemeine Eigenschaften zugeordnet, die nicht explizit über QoS-Parameter ausgedrückt werden. Einer QoS-Kategorie wird dann eine Untermenge der in der gesamten Architektur verfügbaren QoS-Parameter zugeordnet. Hauptsächlich werden dabei solche Parameter ausgenommen, die im Zusammenhang mit den allgemeinen Eigenschaften einer Kategorie keinen Sinn ergeben.

Ein gutes Beispiel für QoS-Kategorien findet sich wiederum in ATM. Hier sind 6 Kategorien definiert (vgl. [Tab. 2.1](#)). Die Kategorie Unspecified Bit Rate (UBR) beispielsweise stellt die Best-

2.2 EINORDNUNG VON NETZ-QoS-ARCHITEKTUREN

QoS-Parameter	ATM QoS-Kategorien					
	constant bit rate	real-time var. bit rate	non-real-time var. bit rate	unspecified bit rate	available bit rate	guaranteed frame rate
peak cell rate	✓	✓	✓	✓	✓	✓
sustained cell rate		✓	✓			
minimum cell rate					✓	✓
maximum burst size		✓	✓			✓
maximum frame size						✓
maximum cell transfer delay	✓	✓				
peak-to-peak cell delay variation	✓	✓				
cell loss rate	✓	✓	✓			

Tabelle 2.1: Spezifizierbare QoS-Parameter der ATM QoS-Kategorien (nach [ATM-TM 99])

Effort Kategorie von ATM dar, entsprechend machen QoS-Parameter wie *cell delay variation* oder *cell loss rate* unter Umständen keinen Sinn und können für UBR auch nicht vereinbart werden.

In der DiffServ-Architektur beschreiben die Zusatzspezifikationen (z.B. [RFC 2597, RFC 3246]) solche QoS-Kategorien, dort **Class-of-Service (CoS)** genannt.

- Für Vorbereitungen der Datenkommunikation werden Steuerprotokolle angegeben. Hierunter fallen u.a. die Anforderung bestimmter QoS-Kategorien und die Vereinbarung von konkreten QoS-Parametern. Betroffen sind aber auch QoS-Aspekte wie Reservierung, Verfügbarkeitsprüfungen, Verkehrskontrakte, Verkehrsglättung, Forwarding-Strategien, Capability-Abgleich und dergleichen. Ebenfalls darunter zu verstehen sind Protokolle zur Signalisierung von Fehlerzuständen.

Die ATM-Protokolle zur Einrichtung eines Switched Virtual Circuit (SVC) an den Schnittstellen User-Network Interface (UNI) [ATM-UNI 02] bzw. Network-Network Interface (NNI) [ATM-PNNI 02] sind Beispiele für Steuerprotokolle. Diese Protokolle regeln den Verbindungsaufbau und die Verhandlung der gewünschten QoS-Parameter (sowohl mit dem Netz als auch mit dem Empfänger). Können die gewünschten Wertebelegungen der QoS-Parameter z.B. von Seiten des Netzes nicht erbracht werden oder lehnt der Empfänger den Verbindungsaufbau ab, geschieht auch hier eine entsprechende Fehlersignalisierung.

Workflows häufig nicht explizit spezifiziert

- Zusätzlich zu den Protokollen werden (manuelle) Prozeduren und Workflows angegeben, die durch Daten- und Steuerprotokolle nicht abgedeckt werden. Häufig geschieht dies allerdings nicht explizit.

Die Festlegung der Semantik der DiffServ-Klassen ist ein Beispiel für eine manuelle Prozedur. Hier wird von der Spezifikation nur implizit gefordert, dass Klassen definiert sein müssen. Konkrete Vorgaben, wie dies zu geschehen hat, werden nicht gemacht. Durch die bereits erwähnten Zusatzspezifikationen [[RFC 2597](#), [RFC 3246](#)] werden allerdings Vorschläge für eine Spezifikation unterbreitet.

Implementierungsrichtlinien nicht im Fokus dieser Arbeit

Neben den beschriebenen Teilen umfassen die Spezifikationen von Netz-QoS-Architekturen natürlich (mehr oder minder) exakte Spezifikationen zur Implementierung der jeweiligen Architektur. Diese spielen aber für diese Arbeit keine weitere Rolle, wie in Abschnitt [4.2.2](#) gezeigt wird, und werden deshalb an dieser Stelle nicht näher erläutert.

Schematische Grobmodellierung der Begriffe

Stellt man die bisher beschriebenen Teile schematisch in Blöcken dar, ergibt sich die Darstellung von [Abbildung 2.6](#). Die Datenprotokolle werden vom Segment Daten repräsentiert, die Steuerprotokolle vom Segment Steuerung. Im dritten Segment sind die Teile die Teile QoS-Kategorien und QoS-Parameter zusammengefasst. Jedes dieser drei Segmente hat dabei unmittelbare Anknüpfungspunkte zu den jeweils anderen beiden Segmenten. Zwischen dem Segment Daten und dem Segment QoS-Kategorien / QoS-Parameter repräsentiert dies die Zuordnung des Datenverkehrs zu Qualitätseigenschaften. Hinter der Verbindung zwischen dem Segment Steuerung und QoS-Kategorien / QoS-Parameter verbergen sich u.a. Protokolle, die für die Definition und Belegung von QoS-Kategorien / QoS-Parameter relevant sind. Der Anknüpfungspunkt zwischen Steuerung und Daten repräsentiert zum einen die klassischen Steueraufgaben wie z.B. Verbindungsaufbau, Flusssteuerung etc., zum anderen sind damit aber auch QoS-spezifische Steueraufgaben gemeint, wie z.B. Verkehrsglättung, Überlastkontrolle etc. Zentral in der Mitte, und mit Interaktionen zu allen der drei genannten Segmente, befinden sich die manuellen Prozeduren und Workflows. Für alle vier der genannten Bestandteile und den Interaktionen zwischen ihnen gilt aber, dass deren konkrete Ausprägung individuell zur dargestellten Netz-QoS-Architektur ist.

Prägt man [Abbildung 2.6](#) am Beispiel DiffServ aus, haben die Elemente folgende konkrete Bedeutung:

- Daten: Das IPv4-Protokoll
- Steuerung: Datensteuerungsprotokolle der IP-Welt, wie z.B. Internet Control Message Protocol (ICMP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP) etc. Keine QoS-relevanten Steuerprotokolle.
- QoS-Kategorien / QoS-Parameter: Keine verbindlich vorge-

2.2 EINORDNUNG VON NETZ-QoS-ARCHITEKTUREN

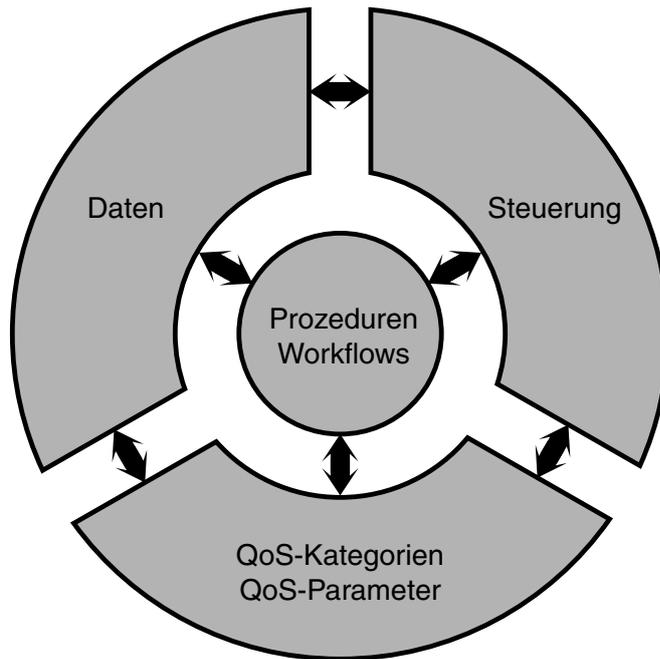


Abbildung 2.6: Grobmodellierung von Netz-QoS-Architekturen

schriebenen Teile, Vorschläge durch [[RFC 2597](#), [RFC 3246](#)].

- Interaktion zwischen Daten und Steuerung: Nur die klassischen, rein Datenspezifischen Interaktionen (z.B. Routing etc.), keine QoS-spezifischen Interaktionen.
- Interaktion zwischen Steuerung und QoS-Kategorien / QoS-Parameter: Keine
- Interaktion zwischen Daten und QoS-Kategorien / QoS-Parameter: Kodierung der Verkehrsklasse im IP-Header.
- Interaktion Prozeduren / Workflows und QoS-Kategorien / QoS-Parameter: Festlegung der CoS-Kategorien

Es bleibt festzuhalten, dass sich in bestehenden QoS-Architekturen die Ausprägung und Präzision der Festlegungen der oben genannten Merkmale stark unterscheiden. Dies machen alleine die gegebenen Beispiele aus der ATM-Architektur und der DiffServ-Architektur deutlich. Eben diese Spannweite unterbindet aber die Erstellung einer durchgängigen Koppelungsarchitektur und rechtfertigt den Ansatz wie er in dieser Arbeit verfolgt wird.

Ausprägung und Präzision der Festlegungen unterschiedlich

In diesem Abschnitt wurde eine sehr wichtige Abgrenzung dieser Arbeit vorgenommen. Der jetzt festgelegte Begriff der Netz-QoS-Architektur umfasst keine Arbeiten, die sich anwendungsspezifisch mit Dienstgüte beschäftigen, z.B. rein auf VoIP ausgelegt sind. Auch Beiträge, deren Fokus die Ende-zu-Ende Dienstgüte unter Einbeziehung der Endsysteme und den darauf ablaufenden Betriebssystemen und Applikationen ist, fallen nicht unter die Kategorie der Netz-QoS-Architekturen.

Ende-zu-Ende QoS und anwendungsspezifischer QoS nicht im Fokus dieser Arbeit

2.2.2 Basistechnologien

Die Spezifikation von Netz-QoS-Architekturen basiert in einigen Fällen auf sog. **Basistechnologien** oder **Basis-Rahmenwerken**. Diese charakterisieren sich dadurch, dass sie grundlegende Mechanismen zu Verfügung stellen, die sich zwar zur Realisierung von Netz-QoS-Architekturen eignen, selbst aber noch keine Netz-QoS-Architektur im Sinne von Abschnitt 2.2.1 darstellen. Entsprechend der Fokussierung dieser Arbeit auf Netz-QoS-Architekturen wird keine Basistechnologie und werden keine Basis-Rahmenwerke betrachtet.

Basistechnologien
nicht im Fokus
dieser Arbeit

Ein aktuelles und sehr prominentes Beispiel hierfür ist Multi Protocol Label Switching (MPLS) [RFC 3031]. Hierbei handelt es sich um einen Internet-Standard, der die ATM zugrunde liegenden Forwarding-Mechanismen auf die Internet-Welt überträgt. In der Literatur wird MPLS häufig als QoS-Architektur bezeichnet. Dies ist im Rahmen der hier getroffenen Definition so nicht korrekt, da die MPLS-Spezifikation in [RFC 3031] alleine nur den Forwarding-Mechanismus beschreibt, der es ermöglicht QoS-fähige Dienste zu realisieren.

MPLS alleine keine
QoS-Architektur

Erst durch zusätzliche Standards, wie [RFC 3035, RFC 3209, RFC 3270], werden QoS-Eigenschaften mit MPLS verfügbar. Zusammen mit jeweils einem der genannten Beispiele wird dann eine Netz-QoS-Architektur realisiert. Zu beachten ist dabei, dass mit jedem der genannten Standards eine jeweils andere auf MPLS basierende Netz-QoS-Architektur erstellt wird. So wird z.B. mit der Kombination [RFC 3031, RFC 3270] eine dem ursprünglichen DiffServ sehr nahe stehende Variante spezifiziert, während die Kombination [RFC 3031, RFC 3209] wichtige Bestandteile für IntServ auf Basis von MPLS spezifiziert. Im späteren Verlauf der Arbeit wird gezeigt werden, dass nur diese zusätzlichen Standards zu betrachten sind, die Basis-Technologie MPLS selbst aber keine Rolle spielt.

Zusätzliche
Standards für „QoS
over MPLS“

Der Charakter einer Basistechnologie wird dadurch unterstrichen, dass sich MPLS auch benutzen lässt, um z.B. VPN-Dienste zu realisieren, ohne dass diese QoS-fähig sind [DaRe oo]. Auch für Anwendungen im Bereich des reinen Traffic-Engineerings wird MPLS benutzt, wiederum ohne QoS-Eigenschaften zu nutzen.

MPLS bedeutet nicht
immer QoS

2.2.3 Einordnung in das Dienstumfeld

Bevor die Spezialisierung der Begriffe aus dem Netz-QoS-Umfeld in Richtung der Dienstorientierung begonnen wird, soll dieser Abschnitt zunächst einige grundlegende Begriffe aus dem Dienstumfeld klären. Zwar wurden in den Szenarios aus Sektion 2.1 bereits Begriffe aus dem Dienstumfeld, wie

2.2 EINORDNUNG VON NETZ-QoS-ARCHITEKTUREN

z.B. Provider verwendet, ohne diese aber eingeführt zu haben. Dies wird nun nachgeholt.

In [GHHK+ 01, GHKR+ 01] wurde das sog. **MNM-Dienstmodell** entwickelt. Es handelt sich um ein universelles Dienstmodell, das unabhängig vom konkreten Szenario angewendet werden kann. Eines der Hauptanliegen des MNM-Dienstmodells ist die Schaffung einer einheitlichen, universellen Terminologie im Bereich des Dienstmanagements. Durch diese Voraussetzungen ist das MNM-Dienstmodell gut geeignet, um in diesem Abschnitt grundlegende Begriffe im Dienstumfeld zu definieren.

Definitionen im Dienstumfeld anhand des szenario-unabhängigen MNM-Dienstmodells

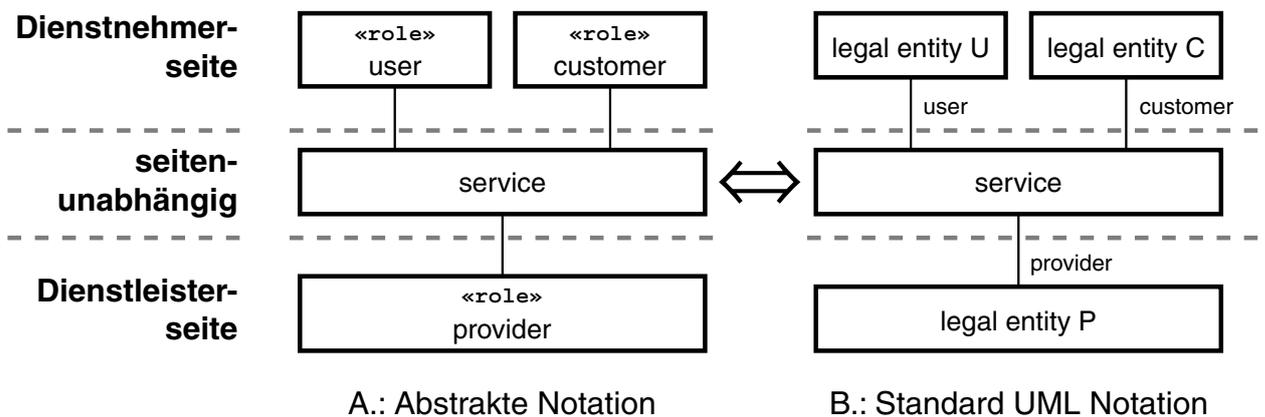


Abbildung 2.7: Basis-Modell (*Basic Model*) des MNM-Dienstmodells (nach [GHKR+ 01])

Im Rahmen des MNM-Dienstmodells wurden mehrere Sichten entwickelt. Das **Basis-Modell** (*Basic Model*) (Abb. 2.7) stellt dabei die einfachste Sicht dar. Es stellt die drei Fundamentalrollen **Nutzer** (*user*), **Kunde** (*customer*) und **Dienstleister** (*provider*) in Beziehung zum **Dienst** (*service*). Die drei Rollen sind dabei den prinzipiellen Seiten **Dienstnehmerseite** (*customer side*) und **Dienstleisterseite** (*provider side*) zugeordnet.

Fundamentalrollen Nutzer, Kunde und Provider

Auf der Dienstnehmerseite werden explizit die Rollen Nutzer und Kunde unterschieden. Eine natürliche oder juristische Person in der Rolle des Kunden ist mit zwei Hauptaspekten befasst. Zum einen ist es der Kunde, der eine vertragliche Beziehung über den Dienst mit dem Provider eingeht. Zum anderen ist der Kunde mit Managementinteraktionen des Dienstes befasst, die er über das **Customer Service Management (CSM)** [LaNe 00] zur Verfügung gestellt bekommt. Die Rolle des Dienstinutzers repräsentiert natürliche oder juristische Personen, die auf die Nutzfunktionalität eines Dienstes zugreifen. Selbstverständlich können die Rollen Nutzer und Kunde auch von derselben Person eingenommen werden. Auf der Dienstleisterseite übernimmt die Rolle des Providers einerseits den Gegenpart zum Kunden im Rahmen der vertraglichen Vereinbarung, andererseits ist er für die Realisierung des Dienstes zuständig. Im MNM-Dienstmodell wird der Dienst als unabhängig von seiner Realisierung angesehen. Entsprechend ist der Dienst selbst keiner der beiden Seiten zugeordnet.

Kundenrolle

Nutzerrolle

Providerrolle

Modelliert man das Szenario aus Abschnitt 2.1.1 mit dem Basis-Modell ergibt sich die Darstellung aus Abbildung 2.8. Die ALV AG ist Kunde des ATM-Dienstes. Da für die vorliegende Modellierung keine Unterscheidung bis zu einzelnen Mitarbeitern gemacht wurde, tritt hier die ALV AG auch als Nutzer des Dienstes auf. Würden einzelne Mitarbeiter der ALV AG, die den ATM-Dienst nutzen, sichtbar sein, dann würde man diese als eigene Klasse, die die Rolle der Nutzer einnehmen, modellieren. Angeboten und realisiert wird der ATM-Dienst von der Firma Data-Systems. Entsprechend ist diese als Dienstleister modelliert worden.

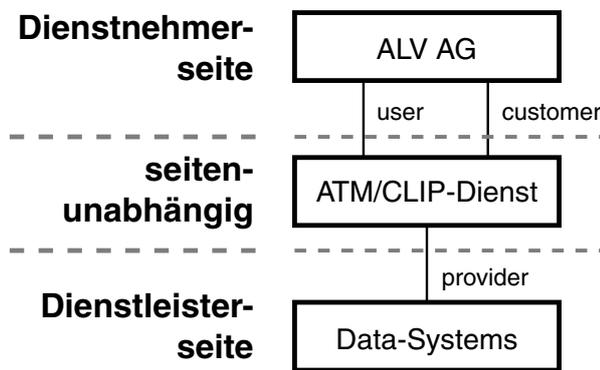


Abbildung 2.8: Basis-Modell für das Szenario der VoIP-Koppelung

Das Basis-Modell ist als Klassendiagramm der Unified Modeling Language (UML) [OMG 03-03-01] dargestellt. Die beteiligten Entitäten und der Dienst sind als Klassen modelliert. Da jede Rolle von einer natürlichen oder juristischen Person (*legal entity*) eingenommen wird, müssen diese für jede Rolle unterscheidbar gemacht werden, z.B. durch Anhängen eines Indexes. Um das Modell übersichtlicher darstellen zu können, wird deshalb der Stereotyp «**role**» eingeführt. Nutzt man diesen, so stellt sich das Basis-Modell in seiner nicht angewendeten, d.h. abstrakten Form wie in Abbildung 2.7-A dar. Wendet man das Modell in einem konkreten Fall an, so wird die Standardnotation von UML für Rollen benutzt (Abb. 2.7-B). Somit sind beide Abbildungen äquivalent.

Stereotypen im abstrakten Modell

Standardnotation im konkreten Modell

Verfeinerte Detailsichten

Weiterhin wird das MNM-Dienstmodell in dieser Arbeit eine wichtige Rolle spielen. Um nun die Beziehungen der Entitäten zum Dienst näher spezifizieren zu können, bietet das MNM-Dienstmodell zwei unterschiedliche Detailsichten: Die Dienst-Sicht (*Service View*) konzentriert sich auf den Dienst selbst, ohne auf Details der Umsetzung einzugehen. Die Realisierungs-Sicht (*Realization View*) dagegen betrachtet Realisierungsaspekte eines Dienstes in der Dienstleisterdomäne. Diese werden im späteren Verlauf der Arbeit eingeführt.

2.2.4 Netz-QoS-Dienst

Dienste, die von Providern ihren Kunden angeboten werden und auf Netz-QoS-Architekturen beruhen, werden als **Netz-QoS-Dienste** bezeichnet. Das Unternehmen ALV AG beispielsweise wird Kunde des Providers Data-Systems (Abschn. 2.1.1) für den ATM-Dienst. Dienstbegriff

Hierbei wird den Kunden nicht notwendigerweise der gesamte Umfang an Diensten und Funktionalitäten angeboten, wie er prinzipiell mittels einer Netz-QoS-Architektur realisiert werden könnte. Betreibt ein Provider beispielsweise ein ATM-basiertes Netz, könnte er trotzdem nur einen Teil der von ATM zur Verfügung gestellten QoS-Kategorien (vgl. Tab. 2.1, S. 21) als Dienst seinen Kunden anbieten. Gründe für solche Einschränkungen können zum einen technischer Natur sein. So befinden sich Infrastrukturkomponenten auf dem Markt, die nur einen Teil einer Spezifikation implementieren. Zum anderen sind auch wirtschaftliche und unternehmenspolitische Gründe ursächlich für eine Beschränkung, z.B. unpassende strategische Ausrichtung, zu geringe Ertragsaussichten etc. Teilfunktionalität einer Architektur als Dienst

Ebenfalls von Fall zu Fall unterscheidet sich die Abbildung von Teilen einer Netz-QoS-Architektur auf Dienste. Wiederum am Beispiel von ATM kann das bedeuten, dass der eine Provider einen Dienst anbietet, im Rahmen dessen der Nutzer zwischen unterschiedlichen QoS-Kategorien wählen kann. Ein anderer Provider könnte für jede von ATM gebotene QoS-Kategorie einen eigenen Dienst für die Kunden am Markt anbieten. Obwohl ATM bei beiden Anbietern die Grundlage für die Realisierung ist, entstehen unterschiedliche Dienst-Portfolios. Unterschiedliche Abbildungen auf Dienst

Ein Netz-QoS-Dienst ist also ein spezieller Fall eines IT-Dienstes, wie er durch das im vorangegangenen Abschnitt eingeführte MNM-Dienstmodell beschrieben wird. Entsprechend können die von einem Provider angebotenen Dienste im Sinne der objekt-orientierten Modellierung als Klasse aufgefasst werden. Schließt ein Kunde einen Vertrag, so wird eine Dienstinstanz erzeugt. Netz-QoS-Dienst Spezialfall eines allg. IT-Dienstes

2.2.5 Netz-QoS-Dienstinstanz

Haben Kunde und Provider einen Vertrag über einen Dienst abgeschlossen, wird der Dienst für diesen Kunden in Betrieb genommen. Folgt man dem objekt-orientierten Modellierungsansatz aus [Dreo 02], so spricht man nun von einer **Netz-QoS-Dienstinstanz**. Für das Szenario aus Abschnitt 2.1.1 unterhält der Provider eine Netz-QoS-Dienstinstanz für die ALV AG. Fortführung des OO-Ansatz

2.2.6 Zusammenfassung der Einordnung

Nachdem in den vorangegangenen Abschnitten die Einordnung in das Umfeld erfolgt ist, fasst Abbildung 2.9 die Beziehungen zwischen den Basisbe-

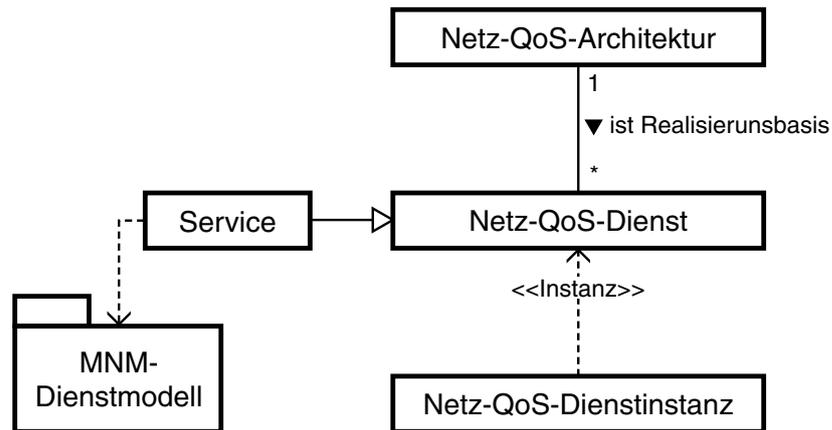


Abbildung 2.9: Beziehungen zwischen Basisbegriffen

Zusammenhang zwischen Basisbegriffen griffen nochmals zusammen. Dem Paradigma der Objektorientierung weiter folgend nutzt die Darstellung ein UML-Klassendiagramm [OMG 03-03-01] als Meta-Modell. Anzumerken ist, dass dies kein Klassendiagramm im Sinne einer programmiersprachlichen Implementierung ist, sondern der bereits in [GHKR+ 01] angewandten Meta-Modellierung entspricht. Die definierten Basisbegriffe sind hier als (Meta-) Klassen dargestellt, ihre Beziehungen untereinander durch entsprechende Assoziationen:

Architektur ist Realisierungsbasis

■ Eine Netz-QoS-Architektur (Abschn. 2.2.1) wird in mehreren Netz-QoS-Diensten (Abschn. 2.2.4) benutzt, um diese zu realisieren. Dabei basiert ein Netz-QoS-Dienst auf genau einer Netz-QoS-Architektur.

Netz-QoS-Dienst ist Spezialisierung von allg. IT-Dienst

■ Ein Netz-QoS-Dienst ist eine Spezialisierung eines allgemeinen IT-Dienstes und damit eine Spezialisierung des Dienstbegriffs, wie er im MNM-Dienstmodell [GHHK+ 01, GHKR+ 01] geprägt wurde und in Abschnitt 2.2.3 eingeführt wurde.

Konkrete Instanz

■ Eine einzelne, konkrete Kunden-Provider-Beziehung bezüglich eines Netz-QoS-Dienstes ist eine Netz-QoS-Dienstinstanz (Abschn. 2.2.5) dieses Dienstes.

Bevor Anforderungen an einen Lösungsansatz identifiziert werden, wird der folgende Abschnitt die Dimensionen der Koppelungsproblematik verdeutlichen.

2.3 DIMENSIONEN DER KOPPELUNGSPROBLEMATIK

Aus den Szenarios der Abschnitte 2.1.1, 2.1.2 und 2.1.3, sowie der im vorangegangenen Abschnitt getroffenen Einordnung, können in diesem Abschnitt nun die typischen Dimensionen der Koppelungsproblematik identifiziert werden. Abbildung 2.10 zeigt in einem Übersichtsdiagramm die Dimensionen auf. Sie leiten sich aus den typischen Fragestellungen der Szenarios, der bereits vorgenommenen Grobmodellierung und der Einordnung in das Dienstumfeld ab. In den nachfolgenden Abschnitten wird jeweils eine Dimension in Ursache und Bedeutung erläutert werden.

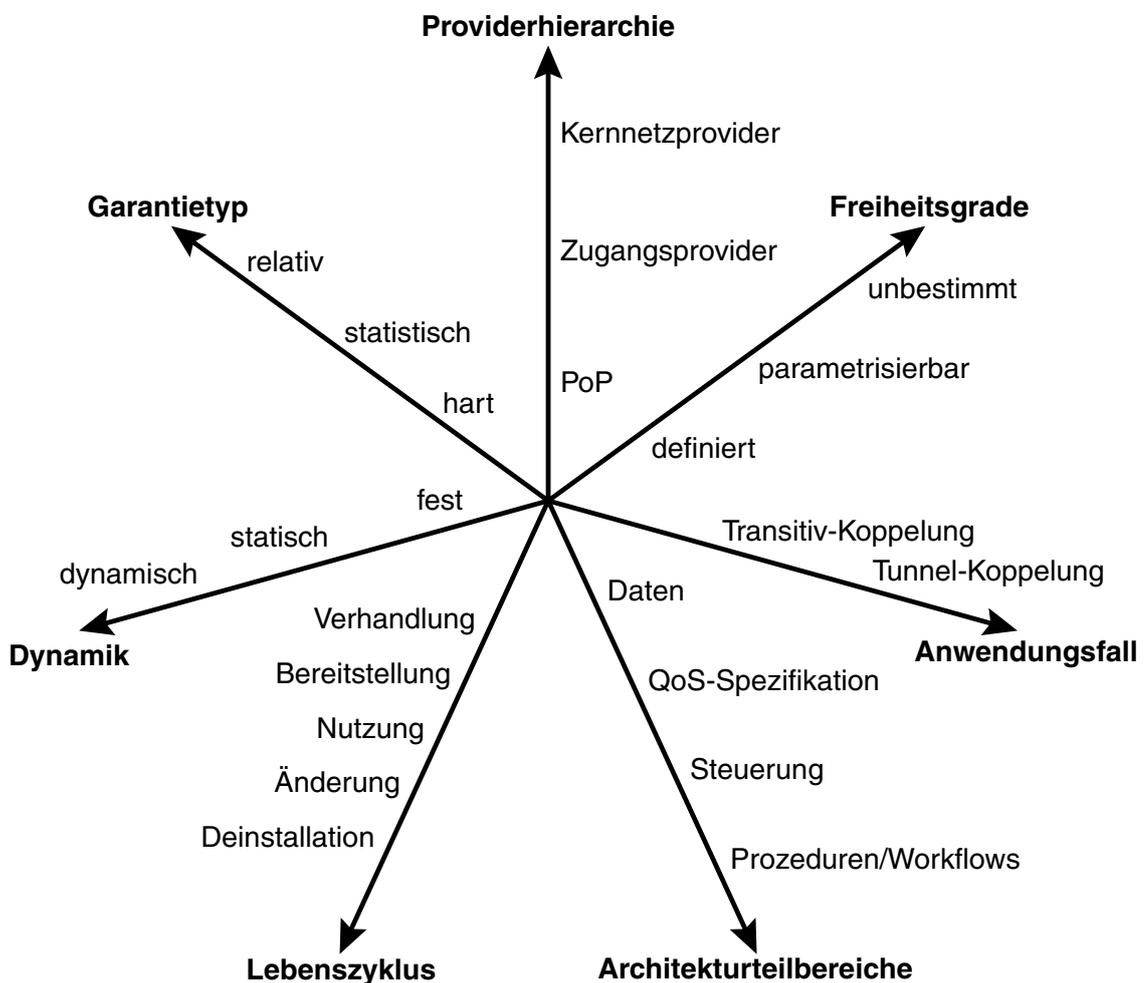


Abbildung 2.10: Dimensionen der Koppelungsproblematik

2.3.1 Dimension Architekturteilbereiche

Um eine Koppelung durchführen zu können, müssen die einzelnen Teile einer Architektur, wie sie in der Grobmodellierung in Abbildung 2.6, Seite 23

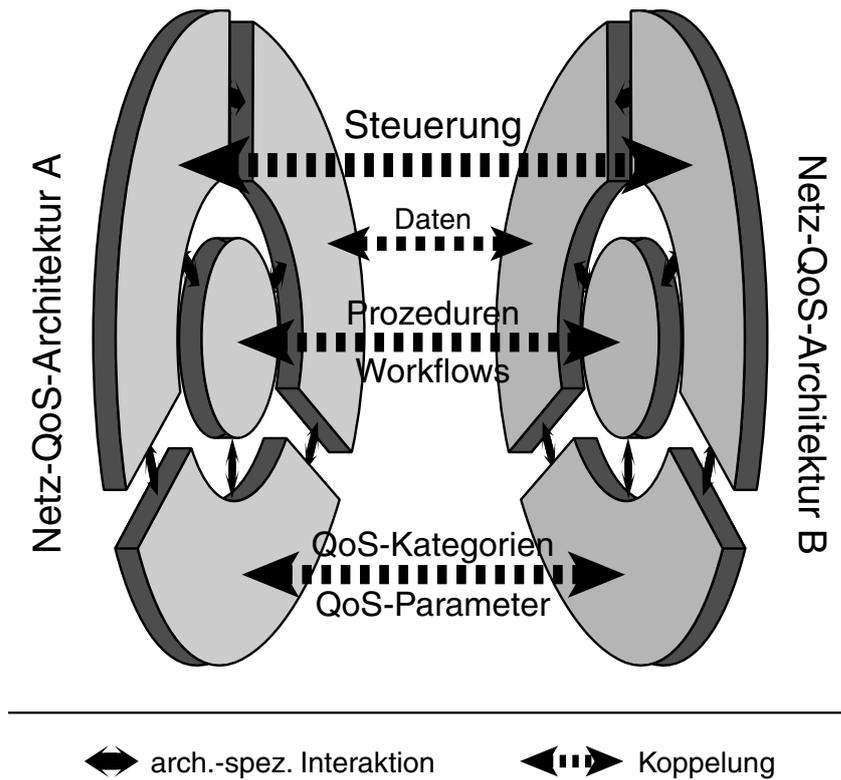


Abbildung 2.11: Koppelung wesentlicher Teilbereiche von Netz-QoS-Architekturen

Architekturspezifika
entscheidend für
Koppelproblematik

dargestellt sind, aufeinander abgebildet werden. Dabei stellen die architekturenspezifischen Eigenschaften der einzelnen Teilbereiche die entscheidenden Hürden dar. Somit sind die spezifischen Eigenschaften der Architekturteilbereiche einer Netz-QoS-Architektur einer der wichtigen Dimensionen der Koppelungsproblematik. Entsprechend Abschnitt 2.2.1 werden hier die vier Teilbereiche Daten, QoS-Parameter, Steuerung und Prozeduren / Workflows, sowie deren Interaktionen, zu unterscheiden sein. Für eine Koppelung sind Abbildungen der jeweiligen Teilbereiche festzulegen (Abb. 2.11) und architekturell zu unterstützen.

2.3.2 Dimension Anwendungsfall

Fundamentale
Anwendungsfälle

Die beiden Szenarios der Abschnitte 2.1.1 und 2.1.2 zeigen zwei völlig unterschiedliche Anwendungsfälle. Diese nehmen Einfluss auf die Koppelung, wie sich an den beiden Beispielen zeigen lässt.

Für die Koppelung der beiden Standorte aus Abschnitt 2.1.1 ist entscheidend, dass die DiffServ-Markierungen am jeweils anderen Standort korrekt nachvollzogen werden können. Während des Transits durch das Netz der Data-Systems ist nur die Erhaltung der Qualitätseigenschaften jeweils für VoIP- und Daten-Verkehr wichtig, eine semantische Identifizierung der Verkehrstypen in der Domäne der Data-Systems (Abb. 2.2, S. 13) ist nicht notwendig.

2.3 DIMENSIONEN DER KOPPELUNGSPROBLEMATIK

Speziell ist es nicht nötig, dass die kundenspezifischen DiffServ-Markierungen für den Provider erkennbar sind. Ein solcher Anwendungsfall wird im Weiteren als **Tunnel-Koppelung** bezeichnet.

Anwendungsfall
Tunnel-Koppelung

Für das Szenario der Ankoppelung des Zugangsproviders an den Kernnetzprovider zur Nutzung dessen Videokonferenzdienstes (Abschn. 2.1.2 und Abb. 2.3, S. 15) dagegen muss der Datenverkehr auch nach dem Übergang zwischen den Domänen semantisch identifiziert werden können. Der Grund hierfür ist, dass der Kernnetzprovider einerseits selbst mit den Videokonferenzdaten umgeht, und andererseits diese Daten beim Übergang in andere Domänen als zum Videokonferenzdienst zugehörig markieren muss. Für solche Anwendungsfälle wird von fortan der Begriff **Transitiv-Koppelungen** verwendet werden.

Anwendungsfall
Transitiv-Koppelung

Später wird gezeigt werden, dass die Unterscheidung dieser beiden Anwendungsfälle einen entscheidenden Einfluss auf die Umsetzung einer Koppelung hat.

2.3.3 Dimension Providerhierarchie

Die Dimension der Providerhierarchie beeinflusst die Koppelung insofern, dass selbst bei gleichen Paarungen von Netz-QoS-Architekturen unterschiedliche Realisierungen der Koppelung sinnvoll sind, wenn es sich um unterschiedliche Partner in der Providerhierarchie handelt. Dies begründet sich aus den sich von Fall zu Fall unterscheidenden „Machtpositionen“.

Providerhierarchie
bestimmt
Machtpositionen

Bezüglich der „Machtpositionen“ wird beispielsweise ein kleiner PoP-Betreiber alleine seinen Zugangsprovider kaum zu größeren Änderungen in dessen Infrastruktur bewegen können. Bei Koppelungen zwischen Kernnetz Providern dagegen kann davon ausgegangen werden, dass Infrastrukturänderungen bei allen beteiligten Seiten möglich sind.

Aber auch die Szenario-Charakteristika „Anzahl der beteiligten Partner“ und der „Wechselhäufigkeit der Partner“ werden durch die Stellung in der Providerhierarchie beeinflusst. Diese Charakteristika können wiederum die Gestaltung einer Koppelung verändern:

Beispielsweise die Wechselhäufigkeit der Partner. Bei einem PoP-Provider kann davon ausgegangen werden, dass Kunden deutlich häufiger wechseln als das mit Vereinbarungen auf Ebene der Kernnetzprovider der Fall ist, die häufig strategisch und langfristig angelegt sind.

Ist nun eine Koppelung langfristig angelegt, sind beispielsweise einmalige manuelle Vorgänge zu Beginn der Koppelung sicherlich akzeptabel. Ist die Wechselhäufigkeit der Kunden dagegen hoch, muss auf eine weitgehende Automatisierung für den Koppelungs-

vorgang Wert gelegt werden. Manuelle Aktionen wären wegen der Häufigkeit nur schwer zu akzeptieren.

2.3.4 Dimension Garantietyp

Unterschiedliche Garantielhärte Für Abbildungen des Teilbereichs der QoS-Parameter (Abb. 2.11) ist deren Garantietyp von großer Bedeutung. Selbst bei gleichlautenden Bezeichnungen eines QoS-Parameters kann sich dessen Umsetzung bezüglich der möglichen Zusicherungen in verschiedenen Architekturen unterscheiden.

Wie bereits die Beispiele von ATM und DiffServ zeigen, können QoS-Parameter hier nur mit unterschiedlicher „Härte“ durchgesetzt werden. Da in DiffServ Verkehrsklassen nur relativ zu einander unterschieden werden, kann DiffServ die Einhaltung diskreter Werte für bestimmte QoS-Parameter nicht garantieren. Will man nun DiffServ mit ATM koppeln, so unterscheidet sich die Güte und Exaktheit der Einhaltung von QoS-Parameter erheblich, selbst wenn bei der Festlegung der DiffServ-Verkehrsklassen die in ATM definierten QoS-Parameter zugrunde gelegt worden sind.

2.3.5 Dimension Lebenszyklus

Workflows implizieren Lebenszyklus Aus dem Teilaspekt der Prozeduren / Workflows motiviert sich die Dimension des Lebenszyklus. Selbst wenn zwei Netz-QoS-Architekturen gleiche Prozeduren / Workflows spezifizieren, kann sich der Zeitpunkt, zu denen diese auszuführen sind, unterscheiden.

Fasst man beispielsweise die Festlegung der Werte von QoS-Parametern als einen Workflow auf, dann muss in Implementierungen, die auf DiffServ basieren, dieser Workflow zur Verhandlungsphase ausgeführt werden. Dagegen kann bei ATM eine Änderung der Belegung von QoS-Parameter architekturbedingt jederzeit auch in der Nutzungsphase durchgeführt werden.

2.3.6 Dimension Dynamik

Unterschiedliche Dynamiken Abhängigkeiten vom Lebenszyklus, wie eben beschrieben, können aber nur dort existieren, wo eine Architektur eine entsprechende Dynamik zulässt. Diese Dimension betreffend findet sich in Netz-QoS-Architekturen die gesamte Spannbreite beginnend von in der Spezifikation festgelegten Gegebenheiten, über solche, die einmalig zu einem bestimmten Zeitpunkt festgelegt werden müssen, bis zu solchen, die dynamisch über den gesamten Lebenszyklus veränderlich sind.

Betrachtet man die Wertebelegung von QoS-Parametern in DiffServ, so sind diese für die gesamte Betriebsphase häufig unveränderlich. Das resultiert daraus, dass in den meisten Fällen für

die Festlegung der Verkehrsklassen Algorithmen und Parametrisierungen verwendet werden, die durch die Hersteller von Netzkomponenten (wie z.B. Router) bereits vordefiniert sind. In diesem Fall sind die Wertebelegungen der QoS-Parameter implizit statisch vorgegeben.

ATM dagegen definiert zwar einen festen Satz von QoS-Parametern, liefert aber die Möglichkeit deren Belegungen für jede einzelne Verbindung neu zu bestimmen und zu verhandeln. Hier ist die Belegung der QoS-Parameter also höchst dynamisch.

2.3.7 Dimension Freiheitsgrade

Schließlich wird die Koppelung von Netz-QoS-Architekturen durch unterschiedlich präzierte Freiheitsgrade erschwert. Im einen Extrem finden sich exakte Definitionen, im anderen Extrem unterbleiben jegliche Angaben. Freiheitsgrade durch Standards

Für die Dimension der Freiheitsgrade markieren wiederum die Beispiele ATM und DiffServ extreme Positionen. Bei ATM werden durch die Standardisierungsdokumente alle verfügbaren QoS-Parameter umfassend aufgelistet (siehe auch Tab. 2.1, S. 21) und die Semantik eines jeden Parameters exakt beschrieben. Für den Fall ATM liegt also der definierte Fall vor.

Bei DiffServ selbst dagegen werden nicht einmal die einzelnen Verkehrsklassen benannt, entsprechend werden keine QoS-Parameter spezifiziert, die eine Verkehrsklasse charakterisieren. Hier liegt also der unbestimmte Fall vor.

Durch die hier beschriebenen Dimensionen sind nun im Wesentlichen die Aspekte aufgezeigt, mit denen sich ein Lösungsansatz für die Koppelungsproblematik zu beschäftigen hat. Im folgenden Abschnitt werden nun konkrete Anforderungen entwickelt, die ein Lösungsansatz zu bewältigen hat. Dimensionen umreißen Aspekte für Lösungsansatz

2.4 ANFORDERUNGEN UND LÖSUNGSANSÄTZE

Nachdem im vorangegangenen Abschnitt die vielfältigen Dimensionen der Koppelungsproblematik veranschaulicht wurden, ist es die Aufgabe dieses Abschnitts Anforderungen zu entwickeln, die an eine Lösung der Koppelungsproblematik zu stellen sind. Zur zielgerichteten Identifizierung der Anforderungen wird dabei in drei Schritten vorgegangen:

- I) Es werden allgemeine Anforderungen abgeleitet, die an jeglichen Lösungsansatz zu stellen sind.
- II) Bevor eine Präzisierung der Anforderungen erfolgen kann, werden prinzipielle Lösungsansätze diskutiert. Diese werden an den vorher entwickelten allgemeinen Anforderungen gemessen. Speziell wird auch der in dieser Arbeit gewählte methodik-basierte Lösungsansatz erstmalig eingeführt.
- III) Unter Kenntnis des hier verfolgten Lösungsansatzes werden spezialisierte Anforderungen abgeleitet, die an einen solchen Ansatz zu stellen sind.

2.4.1 Allgemeine Anforderungen an einen Lösungsansatz

In diesem Abschnitt werden allgemeine Anforderungen betrachtet, wie sie an einen Lösungsansatz für die Problematik der Netz-QoS-Koppelung zu stellen sind.

Anforderung:
Nachhaltig und
allgemein
anwendbar

- Zunächst ist an einen Lösungsansatz die Anforderung der nachhaltigen und allgemeinen Anwendbarkeit zu stellen. Dies meint speziell, dass er sowohl für bestehende als auch für zukünftige Netz-QoS-Architekturen anwendbar sein muss. Diese Anforderung resultiert zum einen daraus, dass sich bis heute keine der bestehenden Netz-QoS-Architekturen global durchsetzen konnte. Aufgrund ihres unterschiedlichen Eignungsgrads für gegebene Anwendungen ist dies auch nicht zu erwarten (siehe hierzu auch Sektion 1.1). Zum anderen ist von einer Entwicklung neuer Netz-QoS-Architekturen auszugehen, beispielsweise aus den Bereichen der Mobilkommunikation.

Anforderung:
Produktorientierung

- Die Produktorientierung des Lösungsansatzes ist eine weitere Anforderung. Sie ergibt sich daraus, dass die an Kunden verkauften Produkte zwar auf Basis einer bestimmten Netz-QoS-Architektur implementiert sind, aber ein Anbieter häufig individuelle Modifikationen für sein Produkt vornimmt oder sogar vornehmen muss.

Am Beispiel ATM heißt das, dass für einen Dienst nicht alle in ATM definierten QoS-Kategorien angeboten werden. Im Falle

2.4 ANFORDERUNGEN UND LÖSUNGSANSÄTZE

eines DiffServ-basierten Dienstes ist die Definition und Spezifikation der Dienstklassen dem Dienstanbieter überlassen.

Diese beiden Beispiele zeigen, dass ein praxisrelevanter Lösungsansatz nicht nur Netz-QoS-Architekturen alleine, sondern auch die darauf aufbauenden Produkte, betrachten muss.

- Die dritte allgemeine Anforderung ist die szenariospezifische Ergebnisorientierung. Wie bereits die beiden Beispielszenarios aus den Abschnitten 2.1.1 und 2.1.2 gezeigt haben, gibt es grundlegend unterschiedliche Anwendungsfälle für eine Koppelung. Ein Lösungsansatz muss die Möglichkeit bieten solche szenariospezifischen Anforderungen einzubeziehen. Ein durch den Lösungsansatz erstelltes Ergebnis muss sich demnach auch an den für das konkrete Szenario gestellten Anforderungen messen lassen. Diese Anforderung bezieht sich aber nicht alleine auf die Dimension des Anwendungsfalls, sie wird auch durch die Dimension der Providerhierarchie unterstützt. Als szenariospezifisch ist nämlich ebenfalls die Anordnung in der Providerhierarchie anzusehen, entsprechend müssen hier alle Ebenen unterstützt werden. Anforderung:
szenariospezifische
Ergebnisorientierung
- Letztlich trivial ist die Anforderung, dass eine Lösung der Koppelungsproblematik auch realisierbar sein muss. Dass es sich jedoch um eine zu beachtende Anforderung handelt, wird der folgende Abschnitt zeigen. Dort werden mögliche Lösungsansätze diskutiert werden. Anforderung:
Realisierbarkeit

Der häufig bemühte Begriff der Unabhängigkeit von bestimmten Randbedingungen und Eigenschaften ist hier irreführend. Ein nachhaltiger und universeller Lösungsansatz muss sehr wohl die Möglichkeit bieten, bestimmte Randbedingungen in die konkrete Anwendung einzubeziehen. Der Lösungsansatz darf also nicht auf bestimmte Randbedingungen und Eigenschaften fixiert sein, sondern er muss vielmehr die Flexibilität bieten, in der konkreten Anwendung solche Eigenschaften und Randbedingungen einbeziehen zu können. Klärung:
Unabhängigkeit von
Randbedingungen

Am Beispiel der szenariospezifischen Ergebnisorientierung bedeutet dies, dass der Lösungsansatz selbstverständlich für alle Szenarios einsetzbar sein muss; wird er aber auf ein konkretes Szenario angewendet, dann müssen sich dessen konkrete Eigenschaften, soweit sie für die Fragestellung von Belang sind, einbeziehen lassen.

Mit den in diesem Abschnitt identifizierten grundlegenden Anforderungen ist nun noch kein bestimmter Lösungsansatz festgelegt. Bevor die Anforderungen konkretisiert werden, wird im folgenden Abschnitt deshalb die Spannweite der prinzipiell denkbaren Lösungsansätze beleuchtet. Lösungsansatz noch
offen

2.4.2 Auswahl des prinzipiellen Lösungsansatzes

Lösungsextreme zur Einordnung der Arbeit

Zur Lösung der Problematik der Netz-QoS-Koppelung bieten sich prinzipiell verschiedene Ansätze an. Um die Spannbreite der möglichen Ansätze aufzuzeigen, werden in diesem Abschnitt zunächst die zwei Extreme vorgestellt, zwischen denen sich Lösungen bewegen können. Anschließend werden diese Extreme vor den allgemeinen Anforderungen aus dem vorangegangenen Abschnitt bewertet. Im Anschluss wird dann der für diese Arbeit gewählte Lösungsansatz kurz vorgestellt, zwischen den Extremen eingeordnet und ebenfalls vor den allgemeinen Anforderungen erstmalig kurz bewertet.

2.4.2.1 Individuallösungen

Lösungsextrem 1: Individuelle Koppelung konkreter Architekturen

Das erste Extrem ist die Beschreibung der Koppelung individuell für zwei konkrete Architekturen oder sogar zwischen zwei konkreten Diensten. Diese Art von Lösungen ist individuell für die behandelte Architekturkombination bzw. für die konkreten Dienstausrüstungen.

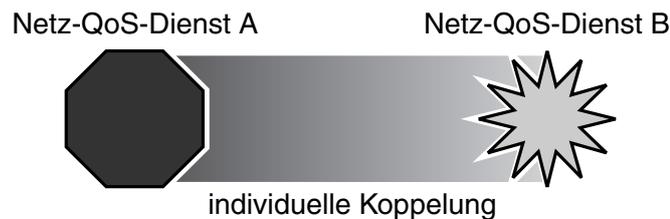


Abbildung 2.12: Individuallösung zwischen zwei Netz-QoS-Diensten

Misst man diesen Lösungsansatz an den allgemeinen Anforderungen ergibt sich folgendes Bild:

- Produktorientierung: potentiell ✓ ■ In Abhängigkeit davon, wie konkret das betrachtete Szenario ausgeprägt ist und ob die Betrachtungsebene nicht nur die von bestimmten Netz-QoS-Architekturen, sondern sogar die bestimmter Netz-QoS-Dienste ist, sind Produktorientierung und szenariospezifische Ergebnisorientierung beim Ansatz von Individuallösungen zumindest potentiell gut erfüllt.
- Szenariospezifische Ergebnisorientierung: potent. ✓ ■ Einer der wichtigsten Gründe, warum der Ansatz einer Individuallösung verfolgt wird, ist deren Realisierbarkeit. Häufig wird dieser Ansatz gerade deswegen gewählt, damit das gewonnene Ergebnis eben auch im konkreten Fall umgesetzt und eingesetzt werden kann.
- Realisierbarkeit: ✓ ■ In Bezug auf allgemeine und nachhaltige Anwendbarkeit versagt dieser Lösungsansatz jedoch. Dies ist offensichtlich und liegt in der Art des Ansatzes selbst begründet.
- Allgemeine und nachhaltige Anwendbarkeit: ✗

Heute häufig vorzufinden Obwohl es sich offensichtlich um eine sehr eingeschränkte Lösungsvariante bezüglich der allgemeinen Verwendbarkeit handelt, ist diese Art der Behandlung der Koppelungsproblematik die aktuell in der Literatur vorherr-

schende. Für eine ausführliche Behandlung sei hier vorausblickend auf Kapitel 3 verwiesen.

2.4.2.2 Umbrella-Koppelung

Das andere Extrem ist eine voll integrierende, universelle und diskrete Koppelungsarchitektur, die in Analogie zum Ansatz von [Kell 98] als Umbrella-Koppelung bezeichnet werden kann. Grundlegende Idee dieses Ansatzes ist es, dass alle existierenden und zukünftigen Netz-QoS-Architekturen unter einer allgemeinen Umbrella-QoS-Architektur U vereint werden können, wie dies in Abbildung 2.13 dargestellt ist. Dabei werden von der Umbrella-Architektur individuelle Abbildungen auf die zu integrierende Architektur angewandt.

Lösungsextrem 2:
Umbrella-Architektur

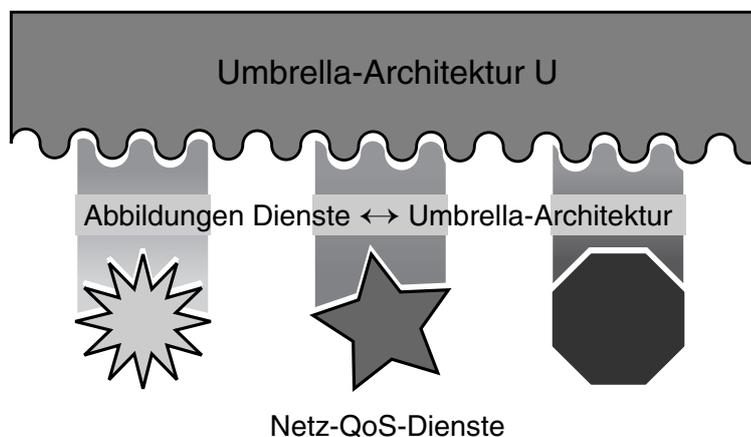


Abbildung 2.13: Umbrella-Koppelung umspannt beliebige Netz-QoS-Dienste

Grundvoraussetzung für diesen Lösungsansatz ist die Existenz der Umbrella-Architektur U. Um die geforderten Abbildungen zu den speziellen Architekturen vornehmen zu können, ohne dass Veränderungen an U selbst notwendig werden, muss U eine absolute Obermenge der zu integrierenden Architekturen darstellen, insbesondere in den folgenden Punkten:

Umbrella-Architektur
ist Obermenge

- U muß die Obermenge der unterschiedlichen Semantiken beherrschen. Speziell sind hier die enorm differierenden Semantiken der QoS-Parameter zu nennen [Gars 04].
- Die Schnittstellen von U müssen mächtig genug sein um die unterschiedlichen Semantiken abbilden zu können.

Obermenge der
Semantiken

Mächtige
Schnittstellen

Dabei wird bis jetzt nur die Ebene der Architekturen berührt. Soll eine Umbrella-Koppelung auf Dienst-Ebene erfolgen, sind weitere Charakteristika von U gefordert:

- Die Umbrella-Architektur muß auf allen Ebenen der Providerhierarchie anwendbar sein. Dabei steht einerseits für die Ebene der PoPs einfache Anwendbarkeit im Vordergrund und andererseits ist auf Ebene der Kernnetz-Provider Mächtigkeit und Leistungsfähigkeit gefragt.

Alle Ebenen der
Providerhierarchie

KAPITEL 2: PROBLEMBESCHREIBUNG, ANSATZ UND ANFORDERUNGSANALYSE

Gesamter Lebenszyklus ■ Um auf Dienstebene einsetzbar zu sein, muß die Umbrella-Koppelung die Dimension des Dienstlebenszyklus vollständig abdecken.

Alle Anwendungsfälle ■ Universalität der Umbrella-Architektur wird auch für die Dimension der Anwendungsfälle benötigt. Hier müssen alle Anwendungsfälle von einer Architektur abgedeckt werden.

Bewertet man den Umbrella-Lösungsansatz bezüglich der allgemeinen Anforderungen, dann ergibt sich folgendes Bild:

Allgemeine und nachhaltige Anwendbarkeit: ✓ ■ Bedingt durch seinen prinzipiellen Ansatz könnte ein Umbrella-Lösungsansatz alle existierenden und zukünftigen Netz-QoS-Architekturen behandeln und erfüllte damit die Anforderung der nachhaltigen und allgemeinen Anwendbarkeit.

Produktorient.: ✓
Szenariospezifische Ergebnisorient.: ✓ ■ Gleiches gilt für die Anforderungen nach Produktorientierung und szenariospezifischer Ergebnisorientierung. Wiederum werden diese Anforderungen immanent durch den integrierenden und universellen Ansatz abgedeckt.

Realisierbarkeit: ✗ ■ Das einzige, aber entscheidende Manko der Umbrella-Koppelung ist jedoch, dass sie sich nicht realisieren lässt oder dies zumindest extrem unwahrscheinlich ist. Dies begründet sich aus der notwendigen Mächtigkeit der Umbrella-Architektur U, die sich aus der Breite der unterschiedlichen Arten von Netz-QoS-Architekturen motiviert. Die beiden Standard-Beispiele dieses Kapitels, ATM und DiffServ, zeigen bereits, in welchem starkem Maß sich einzelne Architekturen unterscheiden können. Zusammen mit der Vielzahl an existierenden und zu erwartenden Netz-QoS-Architekturen ergibt sich ein breit gefächertes Spektrum an zu beherrschenden Varianten.

Hinderungsgrund Semantiken Ein gewichtiger Hinderungsgrund ist der notwendige umfassende Umgang mit den Semantiken aller QoS-Architekturen. Einerseits kann der aktuelle Stand der Forschung mit einer nahezu beliebigen Ausprägung der Semantiken von QoS-Parametern in generischer Weise auf Maschinenebene noch nicht so umgehen wie es hierfür notwendig wäre. Andererseits lassen Architekturen wie DiffServ aber bereits als Teil ihres Standards eine freie Definition der Parametersemantiken zu und für zukünftige Architekturen lassen sich deren QoS-Parameter kaum antizipieren.

Hinderungsgrund allg. Dienstmanagement Ein weiterer schwerwiegender Grund warum die Schaffung der Umbrella-Architektur U als äußerst schwierig anzusehen ist, liegt im Bereich des Dienstmanagements. Das Feld des allgemeinen Dienst-Management momentan noch Gegenstand der Forschung. Hier ist man noch nicht bei feingranularen und auf Maschinenebene automatisierbaren Konzepten angelangt. Für einen Umbrella-Ansatz auf Dienst-Ebene wäre ein umfassendes und tiefgreifendes Verständnis des allgemeinen Dienst-Managements aber erforderlich.

Zusätzlich kann man die Tatsache, dass bis heute kein solcher Ansatz

2.4 ANFORDERUNGEN UND LÖSUNGSANSÄTZE

Anforderungen	Extreme der Lösungsansätze	
	Individuallösungen	Umbrella-Koppelung
nachhaltig und allgemein anwendbar	✘	✓
Produktorientierung	potentiell ✓	✓
szenariospezifische Ergebnisorientierung	potentiell ✓	✓
Realisierbarkeit	potentiell ✓	✘
✓: erfüllt ✘: nicht erfüllt		

Tabelle 2.2: Erfüllung grundlegender Anforderungen durch prinzipielle Lösungsansätze

umgesetzt wurde, als einen weiteren Grund sehen, der die These der Unrealisierbarkeit stützt.

2.4.2.3 Lösungsansatz dieser Arbeit

Fasst man die Bewertung der beiden Extreme aus den vorangegangenen Abschnitten zusammen, erkennt man bereits anhand der allgemeinen Anforderungen, dass keine der beiden Lösungsansätze als befriedigend angesehen werden kann. Wie auch die zusammenfassende Tabelle 2.2 zeigt, sind dabei die Anforderungen nach nachhaltiger und universeller Anwendbarkeit auf der einen Seite und die Realisierbarkeit auf der anderen Seite die großen Problempunkte (Abb. 2.14).

Beide Extreme haben Schwächen

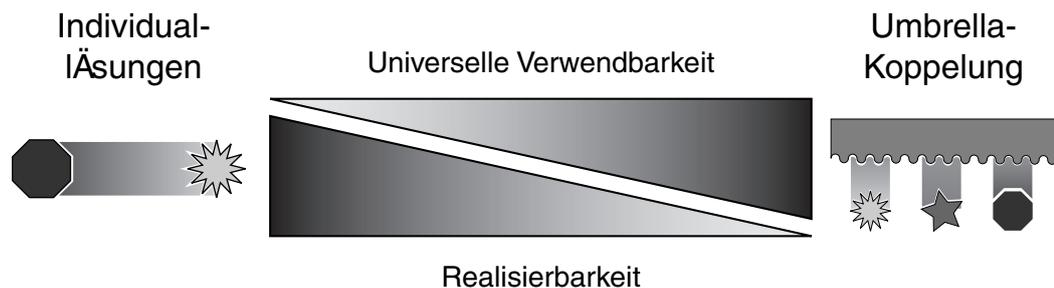


Abbildung 2.14: Extreme für einen Lösungsansatz

Der in dieser Arbeit gewählte Lösungsansatz versucht nun, beide Anforderungen zu erfüllen, ohne dabei die anderen Anforderungen außer Acht zu lassen. Aus den Überlegungen zur Unrealisierbarkeit des Umbrella-Ansatzes ist bereits klar geworden, dass die Anwendungen einer Koppelung immer einen beträchtlichen Individualanteil hinsichtlich der Szenarios und der Produkte (in Form der Dienste) aufweisen. Entsprechend werden die Ergebnisse einer Koppelung ebenso individuell gestaltet sein müssen.

Unerfüllte Anforderungen vereinen

Um dennoch der Anforderung nach allgemeiner Anwendbarkeit Rechnung tragen zu können, schlägt der in dieser Arbeit vorgeschlagene Lösungsansatz einen anderen Weg ein. In dieser Arbeit wird ein generisches Mapping-Konzept entwickelt, um daraus per „Verfeinerung“ schnell Abbildungslösungen gewinnen zu können. Es wird also bewusst darauf verzichtet, eine Lösung in Form einer Umbrella-Koppelung zu verfolgen. Stattdessen wird aber ein generischer und allgemein anwendbarer Weg beschrieben, wie man unter Berücksichtigung fallspezifischer Gegebenheiten zu einer konkreten Lösung gelangen kann. Grundidee des hier vorgestellten Lösungsansatzes ist also, eine generische Methodik zu verwenden, um zu individuellen Lösungen der Koppelungsproblematik zu gelangen.

Neuer Weg zur
allgemeinen
Anwendbarkeit

Grundlage
Modellierung Grundlage für eine generische Methodik ist dabei eine Modellierung von Netz-QoS-Diensten, die für eine „normalisierte“ Darstellung der zu koppeln- den Dienste sorgt. Diese Modellierung ermöglicht es schließlich die Methodik so generisch zu halten, dass diese universell einsetzbar wird.

Hauptschritte Abbildung 2.15 fasst die großen Hauptschritte des hiesigen Lösungsansatzes zusammen:

- Individuelle Netz-QoS-Dienste werden durch eine Modellierung normalisiert. Mit diesem Hauptschritt wird sich Kapitel 4 befassen.
- Basierend auf der normalisierten Darstellung wird durch eine generische Methodik eine individuelle Lösung erstellt. Kapitel 5 wird sich mit dieser Methodik befassen.

Nachdem jetzt die beiden Hauptbestandteile des gewählten Lösungsansatzes charakterisiert sind, kann mit der Verfeinerung von Anforderungen fortgefahren werden. Entsprechend der beiden Hauptbestandteile werden in den beiden folgenden Abschnitten spezifische Anforderungen an den jeweiligen Schritt gestellt werden.

2.4.3 Anforderungen an dienstorientiertes Modell

Spezielle
Anforderungen an
Modell In diesem Abschnitt werden die spezialisierten Anforderungen an den ersten der beiden Hauptschritte des gewählten Lösungsansatzes, nämlich an die Modellierung, beschrieben. Es sei betont, dass die in Abschnitt 2.4.1 gestellten allgemeinen Anforderungen selbstverständlich nach wie vor Bestand haben.

Anforderung:
Normalisierung ■ Um die Anwendbarkeit der generischen Methodik auch allgemein gewährleisten zu können, muss durch die Modellierung die Grundlage gelegt werden. Können nämlich beliebige Netz-QoS-Dienste durch gleichartige Modelle dargestellt werden, dann kann die generische Methodik die unterschiedlichen Netz-QoS-Dienste über ihre Modellierung auf struktureller Ebene in Beziehung zueinander setzen. Die Modellierung muss also eine „normalisierende“ Modellierung sein, die es der Methodik ermöglicht, die Netz-QoS-Dienste systematisch gegenüberzustellen.

2.4 ANFORDERUNGEN UND LÖSUNGSANSÄTZE

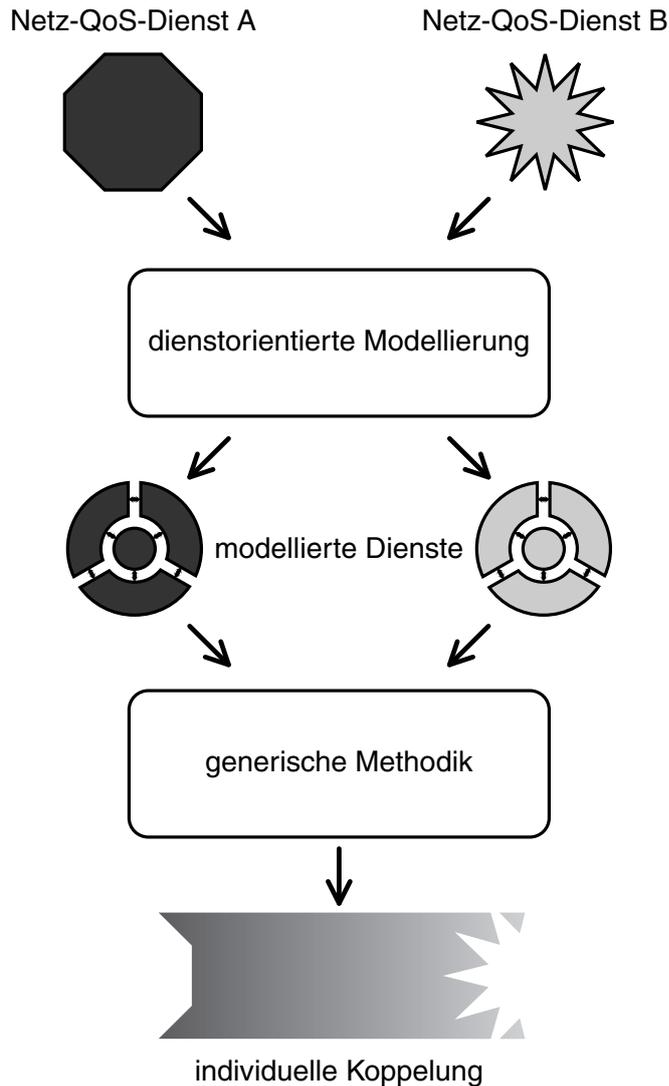


Abbildung 2.15: Übersicht über Methodik-basierten Lösungsansatz

len. Eine beliebige Darstellung des zu modellierenden Dienstes könnte nicht oder nur äußerst aufwendig gehandhabt werden.

Beispielsweise wäre eine Modellierung, die nur die Forderung einer beliebigen Darstellung mittels der UML macht, für eine Methodik ungeeignet. Zwar wären alle modellierten Dienst formal gleich dargestellt, eine inhaltlich unmittelbar vergleichbare Darstellung würde sich aber nur in den seltensten Fällen ergeben.

Normalisierend ist hier also in dem Sinne zu verstehen, dass die Ergebnisse der Modellierungen beliebiger Netz-QoS-Dienste auf struktureller Ebene miteinander in Beziehung gesetzt werden können.

- Die Modellierung muss dienstorientiert sein. Das bedeutet im vorliegenden Fall, dass die Modellierung nicht auf Ebene der Netz-QoS-Architekturen zu erfolgen hat, sondern auf Ebene der Netz-QoS-Dienste
- Anforderung: Dienstorientierung

zu erfolgen hat. Diese Anforderung leitet sich aus den beiden allgemeinen Anforderungen nach Produktorientierung und szenariospezifischer Ergebnisorientierung ab.

Anforderung:
Trennung von Dienst
und Realisierung

- Das Modell muss eine Trennung zwischen dem zu modellierenden Netz-QoS-Dienst an sich und seiner tatsächlichen Realisierung vornehmen. Erstens ist dies nach dem MNM-Dienstmodell [GHHK+ 01, GHKR+ 01] eine allgemein an dienstorientierte Modellierung zu stellende Anforderung. Zweitens befreit es die Lösungsfindung von unrelevanten Details. Beispielsweise ist völlig unerheblich, ob in der Realisierung eines DiffServ-Dienstes Edge-Router eingesetzt werden, die von Hersteller X oder Hersteller Y stammen, solange die gleichen Algorithmen verwendet werden.

Anforderung:
Management
integraler
Dienstbestandteil

- Ebenfalls wurde in [GHHK+ 01, GHKR+ 01] für dienstorientierte Modellierungen bereits gefordert, dass die Betrachtung von Management als integraler Bestandteil des Dienstes gesehen werden muss. Entsprechend müssen alle Management-Interaktionen, die zum ordnungsgemäßen Betrieb eines Netz-QoS-Dienstes benötigt werden, explizit mitbetrachtet werden.

Anforderung:
Abdeckung des
gesamten
Lebenszyklus

- Der gesamte Lebenszyklus eines Netz-QoS-Dienstes muss betrachtet werden. Dies ist wiederum bereits in [GHHK+ 01, GHKR+ 01] gefordert und wird weiterhin durch die gleichlautende Problemdimension aus Abschnitt 2.3.3 zusätzlich unterstrichen.

Für eine dienstorientierte Modellierung nennt [GHHK+ 01, GHKR+ 01] noch die generische und abstrakte Definition. Diese Anforderung ist für diese Arbeit bereits durch die allgemeine Anforderung nach allgemeiner und nachhaltiger Anwendbarkeit an einen Lösungsansatz abgedeckt.

2.4.4 Anforderungen an Methodik

Nachdem mit dem vorangegangenen Abschnitt die Anforderungen an den ersten Teil des Lösungsansatzes aufgestellt wurden, wird sich dieser Abschnitt mit den Anforderungen an die generische Methodik beschäftigen.

Anforderung: Inte-
grationsmöglichkeit
für Teil- und
Individuallösungen

- Bei Betrachtung des Lösungsansatzes der Individuallösungen in Abschnitt 2.4.2 wurde bereits festgestellt, dass eine große Zahl solcher Teil- oder Individuallösungen existieren. Eine generische Methodik darf die dort bereits entwickelten wertvollen Teillösungen nicht ignorieren, vielmehr muss sie die Möglichkeit bieten, die bereits erarbeiteten Erkenntnisse einzubeziehen und diese nutzbar zu machen.

Anforderung:
Gesamter
Lebenszyklus

- Wie für die Modellierung ist auch in der Methodik der gesamte Lebenszyklus eines Dienstes von großer Bedeutung. Wie das Beispiel DiffServ schon mehrfach gezeigt hat, genügt es nicht z.B. nur die Betriebsphase zu betrachten. Schon vor der Bereitstellungsphase müssen bei einer

2.4 ANFORDERUNGEN UND LÖSUNGSANSÄTZE

Beteiligung von DiffServ Entscheidungen für die Koppelung getroffen werden, die sich später nicht mehr ändern lassen.

- Die Grobmodellierung von Netz-QoS-Architekturen (Abb. 2.6, S. 23) zeigt bereits, dass neben Protokollen und Koppelkomponenten auch nichtautomatisierte Abläufe eine tragende Rolle spielen. Entsprechend muss eine Methodik neben Koppelementen und Workflows spezifizieren.

Anforderung:
Spezifikation von
Komponenten und
Workflows

Nochmals sei auch hier betont, dass die in Abschnitt 2.4.1 erarbeiteten allgemeinen Anforderungen ebenfalls von der Methodik zu beachten sind.

2.5 ZUSAMMENFASSUNG VON PROBLEMBESCHREIBUNG UND ANFORDERUNGSANALYSE

Bereits Szenarios zeigen Notwendigkeit der Arbeit

Aus den einführenden Betrachtungen aus Sektion 1.1 und den vorgestellten Szenarios (Sekt. 2.1) wurde klar, dass auch in Zukunft eine heterogene Landschaft von eingesetzten Netz-QoS-Architekturen zu erwarten ist. Ebenfalls wurde bereits durch die gegebenen Beispiele deutlich, dass zwischen unterschiedlichen Netz-QoS-Architekturen die Möglichkeit einer ad-hoc Koppelung im Allgemeinen höchst unwahrscheinlich ist und diese selbst zwischen gleichen Netz-QoS-Architekturen nicht selbstverständlich oder trivial ist.

Begriffsbildung

Durch Sektion 2.2 wurde dann der zentrale Begriff der Netz-QoS-Architektur definiert und mittels eines ersten Grobmodells dargestellt. Anhand des MNM-Dienstmodells wurde eine kurze Einbettung in das Dienstumfeld vorgenommen und mit dessen Hilfe wurden die Begriff Netz-QoS-Dienst und Netz-QoS-Dienstinstanz definiert. Ebenfalls in dieser Sektion wurde eine Abgrenzung zu Basistechnologien vorgenommen, die nicht im Fokus dieser Arbeit stehen.

Problemdimensionen

Die in Sektion 2.3 aufgezeigten Dimensionen der Koppelungsproblematik haben die unterschiedlichen Ursachen für die Notwendigkeit dieser Arbeit näher beleuchtet. Hierbei ist die Vielfältigkeit der Aspekte deutlich geworden, die eine universelle und einfache Lösung der Koppelungsproblematik verhindern.

Lösungsansätze und mehrteilige Anforderungen

Aus den Dimensionen der Koppelungsproblematik konnten dann in Sektion 2.4 zunächst erste allgemeine Anforderungen abgeleitet werden, die für alle Lösungen der Koppelungsproblematik erfüllt sein müssen. Durch die Betrachtung zweier Lösungsextreme konnte dann einerseits im Überblick gezeigt werden, dass die bisherigen Lösungsansätze Defizite aufweisen und andererseits aber eine diskrete allumfassende Lösung als utopisch angesehen werden muss. Daraus resultierend wurde der Lösungsansatz für diese Arbeit skizziert und anhand dessen spezifischere Anforderungen abgeleitet.

Im nachfolgenden Kapitel werden jetzt unter Kenntnis der Anforderungen der Stand der Forschung genauer betrachtet und bewertet.

KAPITEL 3

STATUS QUO DER KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Dieses Kapitel widmet sich der Untersuchung des aktuellen Standes in Fragen der Koppelung von Netz-QoS-Architekturen.

Am Anfang steht die Untersuchung der Koppelungsverfahren von bestehenden Standards. Es werden vier prominente Netz-QoS-Architekturen betrachtet. Anschließend werden die in der Forschung beschrittenen Wege eingeordnet. Der dritte Teil schildert dann zwei konkrete Großprojekte zur Koppelung von Netz-QoS-Architekturen.

Architekturstandards

Forschungsansätze

Großprojekte

Am Ende dieses Kapitels wird deutlich sein, dass die Koppelungsproblematik im Sinn der Anforderungen aus dem vorangegangenen Kapitel noch nicht zufriedenstellend gelöst ist. Hieraus ergibt sich vor dem Hintergrund des Status quo die Rechtfertigung dieser Arbeit.

Zusammenfassung

Abschließend werden noch Bezüge zu benachbarten Gebieten der Koppelungsproblematik hergestellt.

Nachbargebiete

3.1 KOPPELUNGSVERFAHREN BESTEHENDER NETZ-QoS-ARCHITEKTUREN

Nur die wichtigsten Architekturen In dieser Sektion werden einige wichtige Netz-QoS-Architekturen kurz vorgestellt. Die Liste der behandelten Architekturen kann und will nicht vollständig sein. Es werden hier nur jene Architekturen behandelt, die eine gewisse Marktbedeutung erlangt haben und selbst häufig Gegenstand von Forschungsarbeiten sind.

Koppelung aus den Standards Bezüglich der Koppelung einer Architektur an eine andere werden in dieser Sektion nur jene Beiträge betrachtet, die Teil der jeweiligen Architekturstandards sind. Forschungsarbeiten, die nicht Teil eines der Standards sind, werden später in diesem Kapitel untersucht.

3.1.1 Asynchronous Transfer Mode (ATM)

Asynchronous Transfer Mode (ATM) **Asynchronous Transfer Mode (ATM)** [ITU-T I.150, ATM-ARCH 02]¹ definiert umfassend eine medienunabhängige Netztechnologie. Die Pflege und Weiterentwicklung von ATM wird von dem Industriekonsortium ATM-Forum [ATMForum-HP] übernommen, das in enger Kooperation mit internationalen Standardisierungsgremien wie der International Telecommunication Union – Telecom Standardization (ITU-T) [ITUT-HP] zusammenarbeitet.

Umfangreiche Spezifikation Der Spezifikationsumfang von ATM ist weitreichend. Er erstreckt sich von der Abbildung auf unterschiedlichste Technologien der Bitübertragungsschicht, über ein eigenes Adressierungsschema und eigene Management-Protokolle (z.B. Routing, Verbindungsmanagement etc.) bis hin zur Definition unterschiedlicher Datendienste.

Leistungsfähige QoS-Funktionalität Im Bereich der Dienstgüte, die selbst nur ein Teil von ATM ist, glänzt ATM mit einer extrem detaillierten Spezifikation. Es wird ein ganzes Bündel verschiedener QoS-Kategorien samt zugehöriger Parameter (vgl. Tab. 2.1, S. 21) spezifiziert [ATM-TM 99]. Hinsichtlich der Leistungsfähigkeit im QoS-Bereich zeichnet sich ATM dadurch aus, dass es für alle Parameter aller Kategorien mindestens statistische Garantien bietet. Bei einigen Parametern werden sogar deterministische Zusagen gemacht. ATM arbeitet komplett verbindungsorientiert.

Eine Bewertung der Koppelungseigenschaften innerhalb der ATM-Welt anhand der Anforderungen aus Sektion 2.4 ergibt:

- Da es sich um einen spezifischen Standard handelt, ist eine generische Verwendbarkeit der von ATM vorgestellten Techniken und Methoden natürlich nicht gegeben. Das gleiche Argument erklärt, warum ATM kei-

¹Das Dokument [ATM-ARCH 02] ist selbst nur eine Zusammenfassung und ein Wegweiser durch die Vielzahl von Einzeldokumenten in denen ATM spezifiziert ist.

3.1 KOPPELUNGSVERFAHREN BESTEHENDER NETZ-QoS-ARCHITEKTUREN

ne normalisierende Modellierung von Netz-QoS-Architekturen mit sich bringen kann.

- ATM ist ein bis ins Detail ausspezifizierter Standard, der keinen Spielraum für Produktorientierung lässt. Auch auf spezifische Einsatzszenarios wird kein Bezug genommen.
- Koppelungen innerhalb der ATM-Welt sind voll realisierbar und auch realisiert worden.
- Von ATM wird die Dienstorientierung gut umgesetzt. In allen Dokumenten werden Realisierungsaspekte strikt von Nutzerschnittstellen getrennt. Die für eine Koppelung von ATM-Domänen notwendigen Protokolle und Verfahren werden in eigenen Dokumenten (z.B. [ATM-UNI 02, ATM-PNNI 02]) behandelt. Dabei wird auch der gesamte Lebenszyklus zumindest implizit betrachtet.
- Das Management von ATM ist eine der großen Säulen in der Spezifikation. Managementfragestellungen sind integraler Bestandteil des Standards, insbesondere wird die Managementschnittstelle zum Kunden explizit betrachtet.

indexLANE Zur Koppelung mit nicht-ATM-Netzen sieht der Standard im Prinzip zwei Varianten vor. Eine Variante nennt sich LAN² Emulation Over ATM (LANE), die andere Multi-Protocol Over ATM (MPOA).

Wie der Name bereits andeutet, ist der Ansatz von LANE [ATM-LUNI 97, ATM LANE ATM-LANE 98] die Nachbildung eines LANs nach der Art von z.B. Ethernet. ATM gestattet bei LANE die Vereinbarung von QoS-Parametern. Allerdings werden keine detaillierten Angaben oder Vorschläge gemacht, wie LANE zur Koppelung an andere Netz-QoS-Architekturen verwendet werden kann.

MPOA [ATM-MPOA 99] baut selbst auf LANE auf. Es kann benutzt werden um ATM MPOA z.B. IP direkt auf einer ATM-Infrastruktur zu betreiben. Allerdings ist in der MPOA-Spezifikation festgelegt, dass **keine** QoS-Vereinbarungen getroffen werden können. Damit ist MPOA zur Koppelung anderer Netz-QoS-Architekturen ungeeignet.

Insgesamt macht die ATM-Spezifikation also keine konkreten Angaben zur Ankopplung fremder Netz-QoS-Architekturen.

3.1.2 Integrated Services (IntServ)

Integrated Services (IntServ) ist eine Netz-QoS-Architektur, die im Rahmen des Internet-Standardisierungsprozesses [RFC 2026] entstanden ist. Der Standard ist zu Ende entwickelt und die zuständige Working Group [IntServ-WG] ist abgeschlossen.

Der Ansatz von IntServ [RFC 2210, RFC 2215, RFC 2216] verfolgt wie ATM die Reservierung von Ressourcen von Ende zu Ende um Qualitätsmerkma-

²Local Area Network (LAN)

KAPITEL 3: STATUS QUO DER KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Reservierung mit RSVP	le durchzusetzen. Da es sich um einen Standard der Internet-Welt handelt, setzt IntServ zum Datentransport auf das IP-Protokoll; der Datentransport ist also verbindungslos. Für die Reservierung wird auf das Resource Reservation Setup Protocol (RSVP) [RFC 2205, RFC 2208, RFC 2209] zurückgegriffen. Mit diesem Internet-Protokoll wird für jeden Datenstrom einzeln die Reservierung entlang des Pfades von Quelle zum Ziel durchgeführt.
Reservierbare Eigenschaften in eigenen Dokumenten	Die mittels RSVP reservierbaren Eigenschaften sind in eigenen Dokumenten festgelegt. IntServ spezifiziert hier zwei Varianten; den „Controlled-Load Network Element Service“ [RFC 2211] und den „Guaranteed Quality of Service“ [RFC 2212]. Durch den modularen Aufbau von IntServ können aber auch proprietäre Modelle spezifiziert werden.
IntServ skaliert schlecht	Die generelle Problematik von IntServ liegt in seiner schlechten Skalierbarkeit. Dadurch, dass für jeden Datenstrom ein eigener Pfad reserviert wird, müssen die Router, je näher sie sich an den Kernnetzen in der Providerhierarchie (Abb. 2.4, S. 17) befinden, immer mehr Reservierungen verwalten. In den Außenbereichen der Providerhierarchie ist dies gerade noch möglich. Spätestens im Bereich der Kernnetze können die Reservierungen durch ihre schiere Anzahl aber nicht mehr verwaltet werden.

Bezüglich der Anforderungen aus Sektion 2.4 stellt sich IntServ folgendermaßen dar:

- Durch die Fokussierung auf IP ist eine generische Verwendung der IntServ-Konzepte nicht möglich. Aus gleichem Grund ist das IntServ-Modell nicht für eine normalisierende Modellierung geeignet.
- IntServ ist in Teilen produktorientiert, da es zumindest möglich ist eigene Dienstvarianten neben „Controlled-Load Service“ und „Guaranteed Quality of Service“ einzusetzen.
- Der IntServ Standard bietet keine spezifischen Ausrichtungen für unterschiedliche Szenarios.
- Koppelungen innerhalb der IntServ-Welt sind möglich und wurden in der Vergangenheit durchgeführt.
- Die Art, wie IntServ spezifiziert ist, ist nicht dienstorientiert. Der Standard nimmt, mit wenigen Ausnahmen (siehe nächster Punkt), eine implementierungszentrierte Sicht ein.
- Zur Trennung zwischen Dienst und Realisierung sind in IntServ Ansätze vorhanden. Durch den Einsatz von RSVP und der modularen Spezifikation der damit reservierbaren Eigenschaften wird die Anwendersicht von Implementierungsfragen getrennt.
- Managementfragestellungen werden durch IntServ teilweise abgedeckt. Zwar findet keine explizite Trennung in internes und kundenspezifisches Management statt, aber immerhin sind MIBs für das Internet-Management spezifiziert.

3.1 KOPPELUNGSVERFAHREN BESTEHENDER NETZ-QoS-ARCHITEKTUREN

- Der Lebenszyklus wird nicht vollständig abgedeckt. IntServ konzentriert sich voll auf die Betriebsphase.

Zur Interoperabilität gibt IntServ konkrete Spezifikationen für wenige Fälle vor, beispielsweise zu ATM in [RFC 2381, RFC 2382] und zu DiffServ in [RFC 2998]. Generische Konzepte gibt der Standard keine an. Interoperabilität

3.1.3 Differentiated Services (DiffServ)

Bei **Differentiated Services (DiffServ)** handelt es sich ebenfalls um eine Technologie, die im Rahmen des Internet-Standardisierungsprozesses [RFC 2026] entstanden ist. Der Standard ist zu Ende entwickelt und die zuständige Working Group [DiffServ-WG] ist abgeschlossen.

Im Basisteil von DiffServ [RFC 2475, RFC 2474, RFC 3086, RFC 3140, RFC 3260] werden die Einbettung in das IPv4- und das IPv6-Protokoll sowie die grundsätzlichen Mechanismen der Architektur beschrieben. DiffServ unterscheidet sich darin von den beiden vorher vorgestellten Technologien ATM und IntServ. Die QoS-Mechanismen von DiffServ sind komponentenzentriert spezifiziert. Der Ausgangspunkt von DiffServ ist nicht das Gesamtverhalten eines Datenstroms im Durchgang durch das gesamte Netz. DiffServ konzentriert sich auf das Verhalten einzelner Router und wie sie die Datenströme bei deren Durchgang behandeln. Diese Verhaltensspezifikationen werden **Per Hop Behaviour (PHB)** genannt. Basis-Spezifikation ohne konkrete QoS-Eigenschaften

Mit genau diesen PHBs beschäftigen sich ergänzende Spezifikationen, wie z.B. Assured Forwarding (AF) [RFC 2597] und Expedited Forwarding (EF) [RFC 3246, RFC 3247, RFC 3248]. Prinzipiell ist es mit DiffServ aber auch möglich, dass jeder Anbieter seine eigenen PHBs und damit seine eigenen QoS-Eigenschaften festlegt. Standardisierte und freie PHBs

Durch die komponentenzentrierte Sicht von DiffServ ist es sehr schwierig, Aussagen über das Gesamtverhalten eines Datenstroms zu treffen. Dadurch, dass nur das Verhalten an einzelnen Knotenpunkten betrachtet wird, hängt das Gesamtverhalten nicht nur von der individuellen Netztopologie ab. Verschärft wird das Vorhersagbarkeitsproblem noch dadurch, dass bei IP im Prinzip für jedes Paket eine eigene Wegewahlentscheidung getroffen werden und sich damit der tatsächliche Weg eines Datenstroms jederzeit verändern kann. Zusammen mit der Tatsache, dass DiffServ keinen Pfadbegriff (wie bei IntServ) kennt, können in DiffServ nur relative Qualitätsgarantien gegeben werden. Deshalb wird DiffServ auch als eine **Class-of-Service (CoS)**-Architektur bezeichnet. Komponentenzentrierte Sicht erschwert Bestimmung des Gesamtverhaltens

Eine Überprüfung auf Erfüllung der Anforderungen aus Sektion 2.4 ergibt:

- Die hinsichtlich der Leistung deutlich eingeschränkte CoS-Architektur DiffServ ist nicht geeignet einen allgemeinen Beitrag zur Lösung der Koppelungsproblematik zu leisten.

- Durch die Freiheiten im Standard ist eine Ausrichtung auf unterschiedliche Dienstprodukte möglich. Allerdings lässt der Standard Lücken in Fragen, wie damit bei einer Koppelung umzugehen ist.
- Selbst eine Realisierung von Koppelungen innerhalb der DiffServ-Welt ist nicht ohne ergänzende (Forschungs-) Arbeiten möglich. Dies ist bedingt durch die Freiheiten / Ungenauigkeiten im Standard.
- Wie bei allen Architekturstandards kann das DiffServ-Modell nicht als normalisierendes, allgemeines Modell dienen.
- Die Dienstorientierung findet in der Spezifikation keinerlei Verwendung. Beispielsweise müssen selbst zur Realisierung einfacher Dienstvereinbarungen zusätzliche (Forschungs-) Arbeiten hinzugezogen werden.
- Durch die komponentenzentrierte Spezifikationsweise werden der Dienst und seine Realisierung vollständig vermischt.
- Managementfragestellungen werden durch DiffServ teilweise abgedeckt. Zwar findet keine explizite Trennung in internes und kundenspezifisches Management statt, aber immerhin sind MIBs für das Internet-Management spezifiziert.
- Von einer Abdeckung des gesamten Lebenszyklus ist DiffServ weit entfernt. Es werden nicht einmal alle Details der Betriebsphase geklärt.

Keine Spezifikationen zur Interoperabilität

Zur Interoperabilität von DiffServ mit anderen Netz-QoS-Architekturen macht der Standard selbst keinerlei Angaben. Durch die im Prinzip frei wählbare Semantik der PHBs ist nicht einmal zwischen zwei DiffServ-Netzen eine „out-of-the-box“ Interoperabilität gegeben.

3.1.4 Universal Mobile Telecommunications System (UMTS)

Universal Mobile Telecommunications System (UMTS) ist keine reine Netz-QoS-Architektur, sondern eine Mobilfunkarchitektur der dritten Generation. UMTS spezifiziert auch QoS-Eigenschaften – sowohl auf der Luftschnittstelle als auch im Festnetz zu anderen Anbietern. Hier wird die zum Zeitpunkt der Drucklegung der Arbeit aktuelle Fassung „Revision 6“ von UMTS betrachtet. Die aktuell implementierte Fassung von UMTS ist „UMTS Revision '99“.

UMTS ähnlich ausführlich wie ATM

UMTS ist ein sehr ausführlicher Standard. In Umfang und Detailgrad gleicht die UMTS-Spezifikation der von ATM. Im QoS-Bereich werden explizit und detailliert unterschiedliche QoS-Kategorien, deren Parameter sowie deren Bedeutung und Wertebereiche festgelegt. Die einzelnen Kategorien und Parameter erinnern auf den ersten Blick stark an jene von ATM, im Detail sind aber signifikante Unterschiede erkennbar.

3.1 KOPPELUNGSVERFAHREN BESTEHENDER NETZ-QoS-ARCHITEKTUREN

Die Spezifikation der Interoperabilität innerhalb der Architektur gleicht bezüglich den in dieser Arbeit gestellten Anforderungen denen von ATM.	Anforderungserfüllung wie bei ATM
Die Interoperabilität zu anderen Architekturen, insbesondere zu denen der Internet-Welt, sind bis jetzt nicht im Detail spezifiziert und sollen erst später ausgearbeitet werden.	Interoperabilität noch nicht standardisiert

3.2 FORSCHUNGSANSÄTZE

Neben der Arbeit in Standardisierungsgremien wurden und werden im Umfeld von Netz-QoS-Architekturen auch weitere Forschungsarbeiten betrieben. In dieser Sektion wird zuerst eine Arbeit vorgestellt, die zum Ansatz in dieser Arbeit einen engen Bezug aufweist. Im zweiten Abschnitt werden dann weitere Forschungsarbeiten charakterisiert und in den Kontext dieser Arbeit einsortiert.

3.2.1 Schmitt – Heterogeneous Network Quality of Service Systems

Mit [SchmJB 01] hat Schmitt ein Werk vorgelegt, das sich im Kern mit der gleichen Problematik auseinandersetzt, wie es diese Arbeit macht. Schmitt entwickelt hierzu ein konzeptuelles Modell von vernetzten QoS-Systemen.

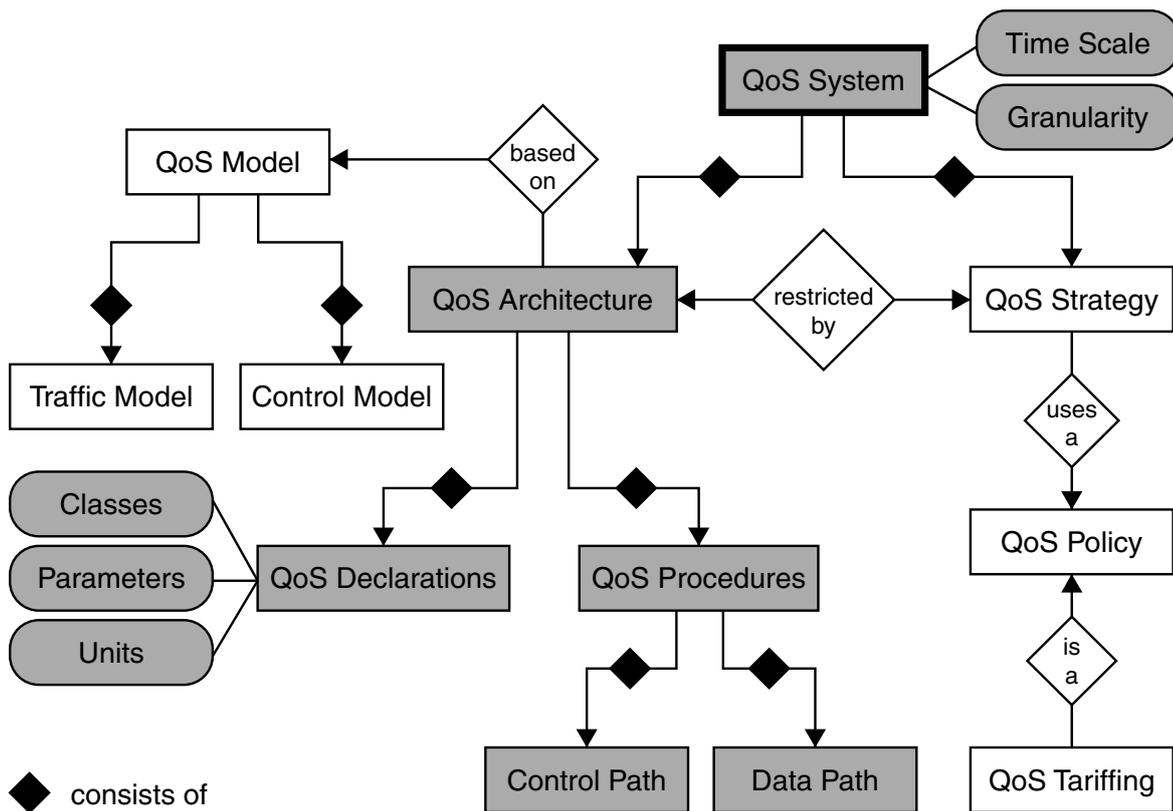


Abbildung 3.1: Konzeptuelles ER-Modell nach [SchmJB 01]

ER-Modell Schmitt entwickelt ein generisches Modell, dessen Entity Relationship (ER)-Diagramm in Abbildung 3.1 dargestellt ist. Im Zentrum des Modells steht die *QoS Architecture*. Diese untergliedert sich in *QoS Declarations* und *QoS Procedures*. Der eine Teil modelliert statische Eigenschaften, konzentriert sich

dabei aber ausschließlich auf die QoS-Parameter und QoS-Kategorien. Statische Eigenschaften des Datentransports bleiben unmodelliert. Der andere Teil modelliert dynamische Vorgänge. Hier werden neben den QoS-relevanten Teilen auch Belange der Datenübertragung betrachtet.

Eine Abstraktionsstufe über der *QoS Architecture* steht das *QoS Model*. Damit sind allgemeine Verkehrsmodelle und generelle Annahmen zur Steuerung und Verkehrskontrolle gemeint.

In der *QoS Strategy* ist die konkrete Umsetzung und Verwendung einer *QoS Architecture* durch einen Provider abgebildet. *QoS Strategy* und *QoS Architecture* bilden zusammen ein *QoS System*.

In seiner Arbeit konzentriert Schmitt sich auf die in Abbildung 3.1 grau hinterlegten Teile. Diese Fokussierung bringt allerdings einige Einschränkungen mit sich, die den Beitrag im Kontext dieser Arbeit problematisch erscheinen lassen. Strikt zu widersprechen ist Schmitt in folgender Aussage:

Fokussierung
schränkt Schmitt ein

[SchmJB 01][S.17]: „*[The shaded entities] ... are the factual components of a QoS system, whereas **all the other entities can be considered as context**. As such they are taken into account for the interworking, but **do not lead to interworking problems themselves**.*“

Für den Teil *QoS Model* ist die Aussage noch akzeptabel. Auch in dieser Arbeit werden allgemeine Verkehrsmodelle und ähnliches ebenfalls nicht explizit betrachtet werden. Das begründet sich durch den hohen Abstraktionsgrad von Verkehrsmodellen und ihrem nur mittelbaren Einfluss auf QoS-Architekturen selbst. Somit können sie keinen direkten Beitrag zur Problemlösung liefern.

Verkehrsmodelle
nicht konkret genug

Allerdings ist die Aussage, dass der Teil *QoS Strategy* selbst nicht zu Problemen bei der Vernetzung führen würde, als äußerst problematisch anzusehen. Im Modell von Schmitt verbergen sich nämlich hinter der *QoS Strategy* Teile, die bei dienstorientierter Betrachtung den Netz-QoS-Diensten zuzuschreiben sind:

- *QoS Policy*: „*The QoS strategy consists of a formal QoS policy how a provider exploits the features of the underlying QoS architecture.*“ Dahinter verbergen sich also Aspekte, wie ein Provider eine Architektur auf einen Netz-QoS-Dienst abbildet.
- *QoS Tariffing*: „*...a QoS policy which allows for gradual activation of features of the utilized QoS architecture is QoS tariffing.*“ Abgesehen davon, dass der Begriff *QoS Tariffing* für die beschriebene kundenspezifische Dienstanpassung sehr irreführend ist, wäre dies in einer dienstorientierten Betrachtung ein wichtiger Bestandteil.

Selbst wenn *QoS Tariffing* im (korrekten) Kontext des Abrechnungsmanagements stehen würde, dürfte es aus Sicht des Dienstmanagements nicht ignoriert werden. Die Bedeutung des Abrechnungsmanagements im Dienstumfeld wurde in [Radi 03] ausführlich dargestellt.

KAPITEL 3: STATUS QUO DER KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Allgemeine Anforderungen	
Anforderungen	Bewertung von [SchmJB 01]
nachhaltig und allgemein anwendbar	✓
Produktorientierung	✗
szenariospezifische Ergebnisorientierung	%
Realisierbarkeit	✓

Anforderungen an die Modellierung	
Anforderungen	Bewertung von [SchmJB 01]
Normalisierung	%
Dienstorientierung	✗
Trennung von Dienst und Realisierung	✗
Management integraler Dienstbestandteil	✗
Abdeckung des gesamten Lebenszyklus	✗

✓: erfüllt %: bedingt erfüllt ✗: nicht erfüllt

Tabelle 3.1: Erfüllung von Anforderungen durch [SchmJB 01]

Ansätze von Dienstorientierung werden nicht verfolgt Entsprechend den obigen Zitaten finden diese Teile in Schmitts Arbeit keine weitere Beachtung. Allerdings wurde in dieser Arbeit bereits gezeigt, dass die dienstorientierte Betrachtung der Koppelungsproblematik notwendig und sinnvoll ist. Beispielsweise hat das Szenario der Koppelung zwischen IP Zugangs- und Kernnetzprovider (Abschn. 2.1.2) gezeigt, dass unterschiedliche Ausprägungen derselben Netz-QoS-Architektur in Form unterschiedlicher Dienste bereits zu erheblichen Problemen bei der Koppelung führen können.

Keine dienstorientierte Betrachtung Normalisierenden Modellierung nur für Architekturen Schmitts Arbeit nimmt keine dienstorientierte Sichtweise auf die Koppelungsproblematik ein. Folglich lässt seine Arbeit Anforderungen wie die Trennung von Dienst und Realisierung, Management als integraler Bestandteil und die Abdeckung des gesamten Lebenszyklus, wie sie nach Abschnitt 2.4.3 an eine Modellierung zu stellen sind, außer Acht. Einzig die Forderung nach einer normalisierenden Modellierung wird erfüllt, allerdings nur auf Ebene der Netz-QoS-Architekturen und nicht auf Dienstebene.

Beitrag später wieder berücksichtigt Zusammengefasst ergibt sich, bezogen auf die Anforderungen aus Abschnitt 2.4.1 und Abschnitt 2.4.3, der Überblick aus Tabelle 3.1. Die Anforderungen aus Abschnitt 2.4.4 finden hier keine Anwendung, da Schmitts Arbeit nicht den gleichen Lösungsansatz wie diese Arbeit verfolgt. Abschließend bleibt festzustellen, dass Schmitts Arbeit keine Aspekte der Dienstorientierung berücksichtigt. Dennoch ist sein Beitrag wertvoll und wird auch im Fortgang dieser Arbeit immer wieder berücksichtigt werden.

3.2.2 Spezifische Ansätze

Die in der letzten Dekade veröffentlichten Forschungsaktivitäten aus dem Bereich der Koppelung von Netz-QoS-Architekturen konzentrieren sich auf drei Schwerpunkte:

■ Bestimmte Architekturen

Viele Forschungsarbeiten beschränken sich bei der Koppelung auf eine oder wenige konkrete Netz-QoS-Architekturen. Beispielsweise analysiert [Camp 96] die Koppelungsproblematik nur anfangs noch allgemein, beschränkt sich aber sehr schnell nur auf Spezialfragen der ATM-Architektur. [KZS 00] behandelt die Koppelung von IntServ mit ATM und liefert dabei ein gutes Beispiel, dass obwohl ein Standard vorhanden ist (siehe Abschn. 3.1.2), es aber auch Sinn machen kann andere Lösungsvarianten zu diskutieren.

Bestimmte Architekturen

Einen breiten Raum nehmen Arbeiten zur Koppelung von DiffServ ein – sowohl mit anderen Netz-QoS-Architekturen als auch mit DiffServ selbst. Da DiffServ viel Raum zur Forschung lässt (positiv gesehen) bzw. so schlecht spezifiziert ist (negativ gesehen), wurden viele verschiedene Varianten diskutiert. Beispiele liefern [TAPC+ 01, GBTB+ 02, Bles 04, HCMO+ 04].

Schwerpunkt DiffServ

Auch der Ansatz von [XiBo 05] fällt unter die Kategorie der Beschränkung auf bestimmte Architekturen. Zwar ist die dienstorientierte Sichtweise der Arbeit im Sinne einer Black-Box-Sicht konsequent umgesetzt, aber die vorgestellte Lösung geht wiederum stillschweigend von DiffServ-Netzen aus. Dadurch werden etliche Probleme sehr unterschiedlicher Architekturen (z.B. DiffServ versus ATM), wie sie später in Kapitel 5 behandelt werden, nicht in die Untersuchungen in [XiBo 05] miteinbezogen.

■ Spezifische Anwendungen

Einige Arbeiten beschränken sich nicht auf bestimmte Netz-QoS-Architekturen, sondern auf bestimmte Anwendungen. Dadurch sind ihre Lösungen wiederum szenariospezifisch. Jüngere Arbeiten sind z.B. [HeRa 05] für Multimedia-Ströme und [BaBa 04] für VoIP.

Spezifische Anwendungen

■ Isolierte Teilaspekte der Koppelungsproblematik

Die dritte generelle Problematik etlicher Forschungsarbeiten besteht darin, dass sie zwar teilweise generisch angelegt sind, dabei aber nur isolierte Teilaspekte der Koppelungsproblematik behandeln ohne diese in einen größeren Kontext einzuordnen. Einige neuere Beispiele sind: Managementintegration mit speziellen Implementierungstechniken [SGGG+ 01, VVWM+ 02, GrSe 05], Accounting [BSFR 05], Überlastkontrolle [LaOw 05], Stromaggregation [QGMT+ 05].

Isolierte Teilaspekte

Durch jede der drei Fokussierungen ist eine universelle und generische Betrachtung der Koppelungsproblematik aus dem Blickfeld geraten. Auch die

Generische Betrachtungen nicht im Fokus

KAPITEL 3: STATUS QUO DER KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

Arbeiten, die Teilaspekte der Koppelungsproblematik untersuchen, lassen sich nicht unmittelbar zu einer großen Gesamtlösung kombinieren, weil entweder bei den Betrachtungen weitergehende Einschränkungen gemacht wurden oder die Arbeiten nicht in einem generischen Gesamtkontext entwickelt wurden.

Integration bestehender Arbeiten Unmittelbar lassen sich die genannten Arbeiten nicht für die anstehende Aufgabe dieser Arbeit verwenden. Wohl aber werden sich später die Teilerkenntnisse der bestehenden Forschungsarbeiten durch diese Arbeit in einen Gesamtkontext einfügen lassen, wie in [Sektion 6.3](#) gezeigt werden wird.

3.3 GROSSPROJEKTE

Exemplarisch für viele Großprojekte zur Etablierung eines QoS-fähigen Netzverbundes werden in dieser Sektion zwei typische Vertreter vorgestellt.

Beide typischen Vertreter zeigen – in unterschiedlichem Ausmaß – das Problem, dass sich Großprojekte meist auf eine sehr enge Auswahl an konkreten Netz-QoS-Architekturen beschränken. Innerhalb des durch die konkreten Architekturen gesteckten Rahmens findet sich dann eine große Bandbreite hinsichtlich der allgemeinen Verwendbarkeit. Typische Großprojekte

Durch ihre Spezialisierung auf bestimmte Architekturen und Technologien können Großprojekte in der Regel keinen allgemeinen Beitrag zum Ansatz dieser Arbeit liefert. Spezialisierung beschränkt Verwendbarkeit

3.3.1 Projekt TEQUILLA

Das TEQUILLA-Projekt [[TEQUILLA-HP](#)] war ein von der Europäischen Union gefördertes Großprojekt zur Entwicklung einer domänen- und länderübergreifenden QoS-fähigen Netzarchitektur. Die Ausgangsbasis des TEQUILLA-Projekts war:

- Es wird nur DiffServ und IntServ betrachtet.
- Für IntServ werden nur die Standardmodelle Best Effort, Controlled Load und Guaranteed Service unterstützt.
- Für DiffServ werden nur die Standard PHBs Expedited Forwarding und Assured Forwarding unterstützt.
- RSVP-Signalisierung wird nur für IntServ unterstützt.
- Weder an IntServ noch an DiffServ werden Erweiterungen oder Veränderungen vorgenommen.
- Im Kernnetzbereich wird ausschließlich DiffServ unterstützt.

Misst man die Ergebnisse des TEQUILLA-Projekts [[DGCG+ D3-4a](#), [DGCG+ D3-4b](#), [DGCG+ D3-4c](#)] an den Anforderungen aus Sektion 2.4, kommt man zu folgendem Fazit: Abgleich mit Anforderungen

- Das TEQUILLA-Projekt ist durch seine Beschränkung auf DiffServ und IntServ nicht allgemein anwendbar.
- Durch die Beschränkung auf die Standard-PHBs von DiffServ und die Standardmodelle von IntServ werden Dienstspezifika, wie sie durch die Basisstandards durchaus vorgesehen wären, nicht betrachtet. Deshalb ist im Rahmen von TEQUILLA keine Produktorientierung möglich.
- Verschiedene Anwendungsszenarios und deren Einflüsse werden durch TEQUILLA durchaus betrachtet. Im Rahmen des Projektumfangs kann das Koppelungsergebnis szenariospezifisch gesteuert werden.

- Insgesamt ist die von TEQUILLA vorgeschlagene Lösung realisierbar, was als Teil des Projekts auch gezeigt wurde.
- Durch die Beschränkung von TEQUILLA auf zwei konkrete Architekturen findet keine allgemeine normalisierende Modellierung statt.
- Dienstorientierung, Trennung von Dienst und Realisierung und die Betrachtung von Management als integralen Systembestandteil werden in TEQUILLA umgesetzt, allerdings nur im beschränkten Projektrahmen und nicht allgemein.
- TEQUILLA deckt den gesamten Lebenszyklus der Koppelung ab. Besonders Fragen der Dienstvereinbarung werden im Projekt untersucht.

Kein generischer Beitrag Für den speziellen Fall eines Netzverbundes, in dem DiffServ in den Kernbereichen und DiffServ oder IntServ in den Außenbereichen Verwendung finden, liefert TEQUILLA einen umfassenden und sehr guten Lösungsansatz. Insgesamt kann TEQUILLA aber keinen allgemeinen Beitrag zur Lösung der Koppelungsproblematik leisten.

3.3.2 Projekt QUASAR

Das QUASAR-Projekt [QUASAR-HP] hatte zur Aufgabe eine Architektur zur Implementierung von QoS im Gigabit-Wissenschaftsnetz (G-WiN) des Deutschen Forschungsnetz (DFN) Vereins zu entwickeln. Ganz im Fokus des Projektes standen...

[BCKK+ M6]: „ ... eine pragmatische Herangehensweise, Anwendbarkeit in einer heterogenen Umgebung und Berücksichtigung der Anforderungen mehrfacher, von einander unabhängiger administrativer Netz-Domains. “

Ergebnisse und Projektverlauf sind in [BJMT+ M1, BCJM+ M2, BCCG+ M3, BCCE+ M4, BCCE+ M5, BCKK+ M6] dokumentiert.

Die im Projekt entworfene Architektur stützt sich bei der Definition seiner anspruchsvollsten von drei Dienstklasse auf die Expedited Forwarding (EF) PHB von DiffServ. Explizit offen gelassen wird, wie die von EF PHB geforderten Eigenschaften implementiert werden. U.a. werden neben klassischen reinen IP-Router-Implementierungen auch ATM-Implementierungen erwähnt.

Abgleich mit Anforderungen Im Lichte der Anforderungen aus Sektion 2.4 zeigt QUASAR das folgende Bild:

- Die QUASAR Architektur ist durch die strikte Beschränkung auf EF PHB nicht generisch einsetzbar.
- Innerhalb des Projektrahmens ist durch die Freiheit der konkreten Implementierung des EF PHBs eine gewisse Produktorientierung gegeben.

- Anpassungen auf unterschiedliche Szenarios sieht das Projekt nicht vor. Es ist spezifisch auf die Bedürfnisse des DFN zugeschnitten.
- Die Realisierbarkeit wurde als Teil des Projekts explizit gezeigt.
- Eine normalisierende Modellierung im generischen Sinn findet nicht statt.
- Was eine allgemeine Dienstorientierung angeht, zeigt das Projekt ein durchwachsenes Bild. Teilweise sind Ansprüche der Dienstorientierung gut verwirklicht (siehe z.B. nächster Punkt), andererseits werden wichtige Voraussetzungen zur Dienstorientierung (z.B. der Lebenszyklus) nicht betrachtet.
- Die Trennung von Dienst und Realisierung ist im konkreten Projektrahmen dadurch gegeben, dass für jede Domäne ein Verhalten nach EF PHB gefordert wird, dessen technische Realisierung jeder Domäne aber freigestellt ist.
- Managementfragestellungen nehmen einen breiten Raum in QUASAR ein. Insbesondere Fragestellungen der Messung werden intensiv erörtert.
- Eine Abdeckung des gesamten Lebenszyklus findet nicht statt. Getreu seiner „*pragmatischen Herangehensweise*“ fokussiert das Projekt klar die Betriebsphase.

Insgesamt bleibt festzustellen, dass QUASAR ein spezifisches Projekt ist, das an den meisten Stellen auf die speziellen Bedürfnisse des Szenarios zugeschnitten ist. Szenariospezifischer Ansatz

3.4 ZUSAMMENFASSUNG UND BENACHBARTE GEBIETE

In den vorangegangenen Sektionen wurde der Status quo der Koppelung von Netz-QoS-Architekturen in Architekturstandards, Forschungsarbeiten und Großprojekten aufgezeigt. Tabelle 3.2 stellt die Ergebnisse anhand der Anforderungen aus Sektion 2.4 im Überblick dar.

	Standards von Netz-QoS-Architekturen				Forschungsansätze		Großprojekte	
	ATM	IntServ	DiffServ	UMTS	[Schm]B 01	spez. Ansätze	TEQUILLA	QUASAR
nachhaltig und allgemein anwendbar (absolut)	x	x	x	x	✓	(%)	x	x
Produktorientierung	x	(%)	(%)	x	x	□	x	(✓)
szenariospezifische Ergebnisorientierung	x	x	x	x	%	□	(✓)	x
Realisierbarkeit (relativ)	✓	✓	%	✓	✓	(✓)	✓	✓
Normalisierung (absolut)	x	x	x	x	%	(%)	x	x
Dienstorientierung	(✓)	x	x	(✓)	x	□	(✓)	(%)
Trennung von Dienst und Realisierung	(✓)	(%)	x	(✓)	x	□	(✓)	(✓)
Management integraler Dienstbestandteil	(✓)	(%)	(%)	(✓)	x	□	(✓)	(✓)
Abdeckung des gesamten Lebenszyklus	(✓)	x	x	(✓)	x	□	✓	x
Heterogene Koppelung	x	✓	x	x	✓	(%)	✓	✓

✓	erfüllt
%	nur teilweise erfüllt
x	nicht erfüllt
(✓)	im Rahmen der Einschränkungen erfüllt
(%)	im Rahmen der Einschränkungen nur teilweise erfüllt
□	im Überblick nicht anwendbar

Tabelle 3.2: Erfüllung der Anforderungen durch den Status quo im Überblick

3.4 ZUSAMMENFASSUNG UND BENACHBARTE GEBIETE

Zusammenfassend lässt sich feststellen:

- Standards zeigen eine gewisse Ignoranz anderen Netz-QoS-Architekturen gegenüber. Von ihnen ist kein generischer Lösungsbeitrag zu erwarten. Standards ignorieren andere Standards
- Großprojekte fokussieren häufig auf die Implementierung und fixieren eine feste Menge konkreter Architekturen. Sie leisten keinen generischen Beitrag. Großprojekte nicht generisch
- Bei Forschungsarbeiten zeigt sich ein ähnliches Bild. Ausgangsbasis sind meist feste Architekturen oder Anwendungen. Dort, wo dies nicht der Fall ist, werden nur kleine Teilaspekte der Koppelungsproblematik behandelt. Forschungsarbeiten häufig nur Teilprobleme
- Dort, wo umfangreich die gestellten Anforderungen erfüllt werden, geschieht das nur in einem eng gesteckten Rahmen konkreter Architekturen. Forschungsarbeiten häufig eingeschränkt
- Überall dort, wo Realisierbarkeit gegeben ist, mangelt es an nachhaltiger und allgemeiner Anwendbarkeit. Realisierbarkeit bedeutet Mangel an Generik
- Im Sinne der Zielstellung dieser Arbeit ist die Arbeit von Schmitt [SchmJB 01] ein sehr interessanter Beitrag, weil darin ein generischer Standpunkt eingenommen wird. Allerdings wird von Schmitt die Dienstorientierung außer Acht gelassen. Schmitts Beitrag generisch aber nicht dienstorientiert

Damit ist deutlich geworden, dass die Koppelungsproblematik von Netz-QoS-Architekturen in einem generischen Sinn durch den aktuellen Stand in Standards und Forschung nicht zufriedenstellend gelöst wird. Somit besteht Raum und Bedarf für diese Arbeit. Bedarf für diese Arbeit

Im QoS-Bereich gibt es weitere benachbarte und spezielle Gebiete, die nicht im direkten Fokus dieser Arbeit stehen. Hier sind beispielsweise Queue Management, Constraint Based Routing, Intradomain Traffic Engineering / Management, QoS / Constraint-Based Routing, Congestion Avoidance / Management und Service Level Management zu nennen. Auf zwei weitere benachbarte Aspekte wird im Folgenden noch näher eingegangen.

3.4.1 Basistechnologie MPLS

Wie in Abschnitt 2.2.2 bereits erläutert, basieren reale Implementierungen von Netz-QoS-Diensten in einigen Fällen auf Basistechnologien oder Basis-Rahmenwerken. Sie stellen grundlegende Mechanismen zur Verfügung, die sich zwar zur Realisierung von Netz-QoS-Architekturen eignen, selbst aber noch keine Netz-QoS-Architektur im Sinne von Abschnitt 2.2.1 darstellen. In den letzten Jahren hat dabei **Multi Protocol Label Switching (MPLS)**, ein von der IETF Working Group [MPLS-WG] entwickelter Internet-Standard, eine bedeutende und noch stetig wachsende Verbreitung erfahren. Die steigende Marktdurchdringung dieser konkreten Basistechnologie ist der Grund dafür, weshalb sie hier explizit erläutert und eingeordnet wird. MPLS stellt grundlegende Mechanismen bereit

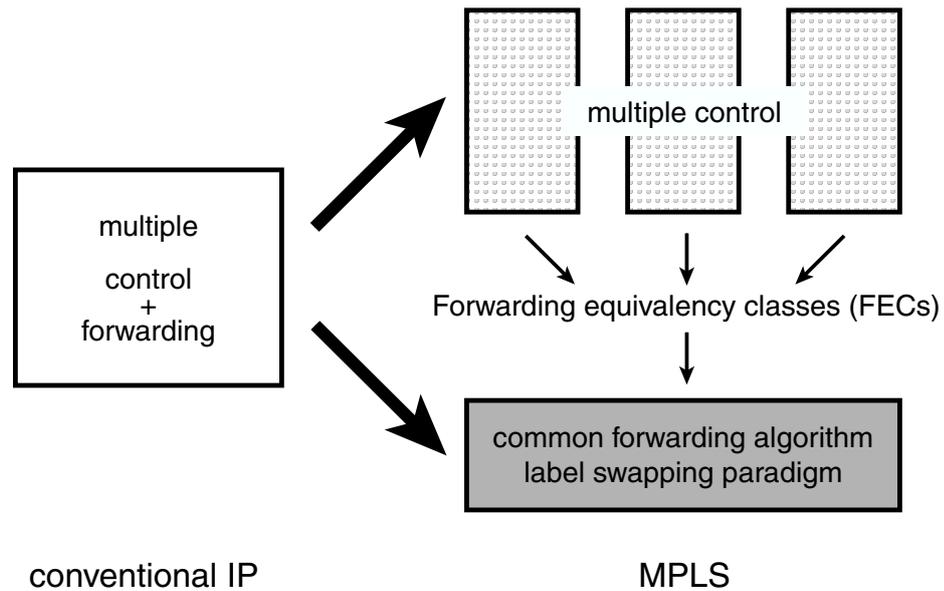


Abbildung 3.2: Grundidee von MPLS (nach [Roel 01])

Trennung von Forwarding-Entscheidung und Vorgang

In konventionellen IP-Netzen sind in Routern die Entscheidungen zu Wegwahl bzw. Weiterleitung einerseits und der Vorgang der Weiterleitung von Paketen andererseits untrennbar miteinander verwoben (linker Teil von Abb. 3.2). Die Grundidee von MPLS besteht darin, diese beiden Vorgänge explizit voneinander zu trennen (rechter Teil von Abb. 3.2). Dies ermöglicht es, mehrere verschiedene Kontrollmechanismen parallel nebeneinander zu betreiben. Die Kontrollmechanismen teilen die Pakete in sog. Forwarding Equivalency Classes (FEC) ein. Jede FEC definiert dabei eine Menge von Paketen, die auf gleiche Weise weitergeleitet werden sollen. Der Vorgang der Weiterleitung selbst basiert auf einem einheitlichen Algorithmus, nämlich dem der ATM-Welt entliehenen *label swapping* Prinzip. Der Weiterleitungsalgorithmus wird hier nicht näher betrachtet.

Getrennte Basisspezifikation und schichtspezifische Abbildungen

Im Umfeld von MPLS ist strikt zwischen der Basisspezifikation von MPLS selbst und Zusatzspezifikationen, die MPLS mit anderen Technologien kombinieren, zu trennen. MPLS definiert eine Zwischenschicht unterhalb der OSI-Schicht 3 und oberhalb von OSI-Schicht 2 (Abb. 3.3). In der Basisspezifikation [RFC 3031] wird ein schichtunabhängiger Teil definiert, in dem der Forwarding-Mechanismus und die Beschaffenheit der Labels festgelegt werden. Der Forwarding-Mechanismus benutzt alleine semantikleose Marken, die sog. Labels, sowie das Interface, an dem ein Paket ankommt, um zu entscheiden, an welches Interface das Paket weitergeleitet werden soll und mit welcher Marke es versehen wird. Dieser ATM-ähnliche Mechanismus ermöglicht es, analog zu ATM, für die Weiterleitung schnelle Switching-Komponenten anstatt vergleichsweise langsame Router einzusetzen. In Richtung der nach „oben“ und „unten“ angrenzenden Schichten werden Basismechanismen angeboten, die für eine schichtspezifische Abbildung auf und von MPLS benutzt werden können. Es ist zu betonen, dass die Basisspezifikation den-

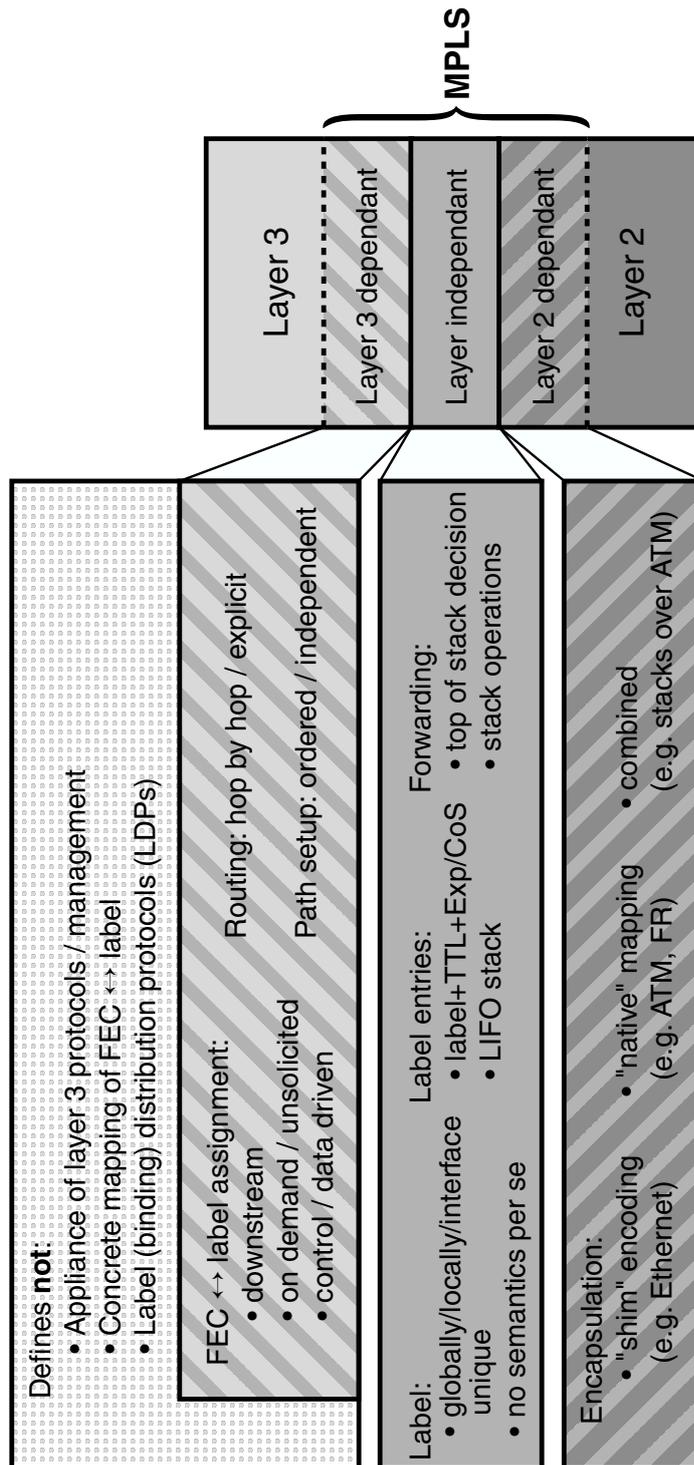


Abbildung 3.3: MPLS im OSI-Schichtenmodell (nach [Roel 01])

noch unabhängig von Schicht 3 und 2 gehalten ist (und das obwohl es sich um einen Internet-Standard handelt).

Die Zusatzspezifikationen von MPLS beziehen sich einmal auf die Abbildung des Basismechanismus auf Schicht 2. Hier bestehen beispielsweise Definitionen für Ethernet (in [RFC 3031] enthalten), Frame Relay (FR) [RFC 3034]

Schichtspezifische Abbildungen

und ATM [RFC 3035]. Speziell bezüglich dieser Arbeit sei erwähnt, dass ATM hier nicht primär in seiner Eigenschaft als QoS-Architektur genutzt wird, sondern als gewöhnliche Technologie der Schicht 2 gesehen wird. Auch zur Abbildung von Technologien der Schicht 3 auf MPLS existieren zusätzliche Standards, beispielsweise für die Netz-QoS-Architekturen IntServ / RSVP [RFC 3209] und DiffServ [RFC 3270]. Aber auch zur Realisierung von Anwendungen, die keine Qualitätseigenschaften umfassen, lässt sich MPLS einsetzen, z.B. VPN-Dienste (ohne dass diese QoS-fähig sind) [DaRe 00] und für Anwendungen im Bereich des reinen Traffic-Engineerings. Diese Beispiele unterstreichen den Charakter einer Basistechnologie.

MPLS liefert keinen Beitrag

Nachdem bereits deutlich geworden ist, dass MPLS nicht als eine Netz-QoS-Architektur sondern als Basistechnologie einzustufen ist, stellt sich die Frage, warum MPLS mit seinem schichtunabhängigen Teil nicht zur Lösung der Koppelungsproblematik beitragen kann. Die Antwort darauf liegt schlicht in der Tatsache begründet, dass MPLS weder für FECs noch für Labels eigene Semantiken oder auch nur Spezifikationsrichtlinien mitbringt. Dadurch werden die Architekturspezifika von Netz-QoS-Architekturen, die sich auf MPLS stützen, unmittelbar nach MPLS hinein übernommen. Somit kann MPLS selbst keinen allgemeinen Beitrag zur Lösung der Koppelungsproblematik leisten. Warum MPLS selbst bei gleichen zu koppelnden Architekturen keine generische Lösung bieten kann, wird in [Roel 01] gezeigt.

3.4.2 Spezifikation von QoS

Im späteren Verlauf der Arbeit wird klar werden, dass eine möglichst exakte Spezifikation von QoS bzw. von QoS-Parametern ein wichtiger Punkt in der Modellierung von Netz-QoS-Architekturen ist. Alleine das Gebiet der Spezifikation von QoS-Parametern ist ein so weites Feld, dass es im Rahmen dieser Arbeit nicht erschöpfend behandelt werden kann. In [Gars 04] werden exemplarisch 18 verschiedene Ansätze zur QoS-Spezifikation vorgestellt und anhand eines detaillierten Anforderungskatalogs untersucht.

Aspekte der QoS-Spezifikation

Dass für die QoS-Spezifikation selbst noch nicht alle Forschungsfragestellungen abgeschlossen sind, wird in [GaRo 04] anhand des Layer - Aspect - Life cycle (LAL)-Kubus (Abb. 3.4) aufgezeigt. Der Name rührt von den Achsen dieses Kubus her:

- *Layer of abstraction*: Abstraktionsebene im geschichteten System, von konkreten (Hard- und Software-) Ressourcen über Anwendungen hin zur Ebene der Anwender.
- *Service Aspect*: Neben den elementaren Dienstbestandteilen Funktionalität und Management (siehe auch Abschnitt 4.2.1) stellt die durch einen Dienst bereitgestellten / verarbeiteten Informationen einen eigenen Aspekt dar.
- *Life cycle phase*: Der Lebenszyklus eines Dienstes (siehe auch Abschn. 4.3.2)

3.4 ZUSAMMENFASSUNG UND BENACHBARE GEBIETE

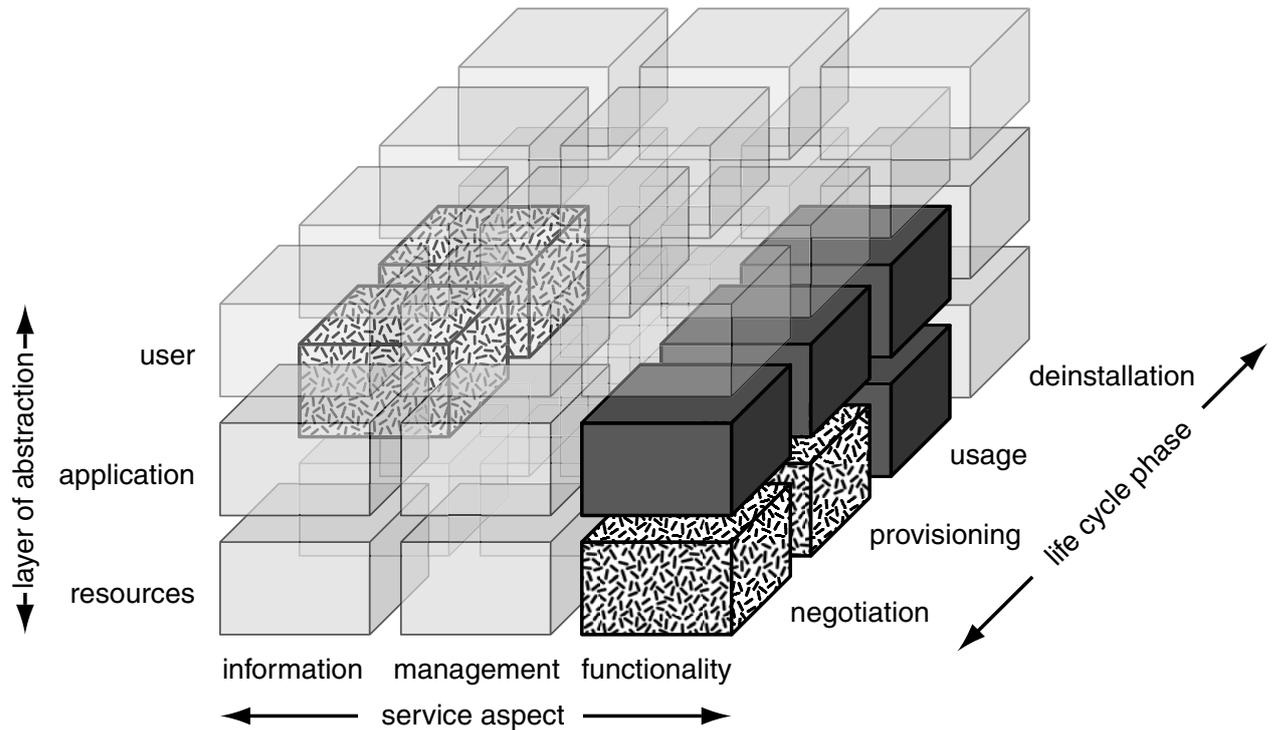


Abbildung 3.4: LAL-Kubus (nach [GaRo 04])

Nach [GaRo 04] muss für eine vollständige Behandlung der QoS-Spezifikation des gesamten LAL-Kubus abgedeckt werden. Trägt man die bisherigen Ansätze in den LAL-Kubus ein, wie in Abbildung 3.4 geschehen, erkennt man, dass dieses Ziel bei weitem noch nicht erreicht ist: Die dunkelgrauen Bereiche markieren jene, die ausreichend abgedeckt sind, die gepunkteten sind es nur teilweise. Erschwerend kommt hinzu, dass die Teilabdeckung nicht mit einem Ansatz oder zumindest mehreren integrierten Ansätzen erreicht wird, vielmehr stehen die einzelnen Ansätze noch alleine nebeneinander.

QoS-Spezifikation noch nicht völlig abgedeckt

Der neue Ansatz aus [Gars 04] stellt eine wesentliche Verbesserung hinsichtlich der Dienstorientierung im Allgemeinen und der Integration des Dienstlebenszyklus im Besonderen dar. Hervorzuheben ist, dass auch der Aspekt der Messung als Teil der Semantik von QoS-Parametern breite Beachtung findet. Auf den Ansatz von [Gars 04] wird später noch zurückgegriffen werden.

KAPITEL 3: STATUS QUO DER KOPPELUNG VON NETZ-QoS-ARCHITEKTUREN

KAPITEL 4

MODELLIERUNG VON NETZ-QoS-DIENSTEN

In diesem Kapitel wird der erste von zwei großen Teilen des Lösungsansatzes vorgestellt. Im Folgenden wird zuerst nochmals der gesamte Ansatz zusammenfassend gezeigt und dann ein Überblick über diesen ersten Schritt gegeben.

Erster Teil des Lösungsansatzes

Erste wichtige Erkenntnisse werden durch eine Vertiefung der Dienstorientierung gewonnen. Sie liefert die Fokussierung für die sich anschließenden beiden Modellierungsschritte. Diese beiden Modellbildungen behandeln dann nacheinander zuerst Netz-QoS-Architekturen und anschließend die darauf aufbauenden Netz-QoS-Dienste.

Dienstmodellierung
Zwei Modellierungsschritte

Im Ergebnis wird dieses Kapitel in Form einzelner Modelle die notwendige Ausgangsbasis für die dann im nächsten Kapitel entwickelte Methodik liefern.

4.1 LÖSUNGSDIEE UND VORGEHEN

Um das Vorgehen in diesem Kapitel motivieren zu können, wird im Folgenden die dieser Arbeit zugrunde liegende Lösungsidee erläutert. Im Anschluss daran wird dann das Vorgehen für das restliche Kapitel beschrieben.

Ansatz: Generischer Weg zur Gewinnung einer spezifischen Lösung

Vor dem Hintergrund der in Abschnitt 2.4.2 besprochenen Extremansätze verfolgt diese Arbeit einen Weg, der sich in einem Satz als „Generischer Weg zur Gewinnung einer spezifischen Lösung“ beschreiben lässt. Dabei werden im Wesentlichen zwei Hauptschritte durchgeführt (Abb. 2.15, S. 41).

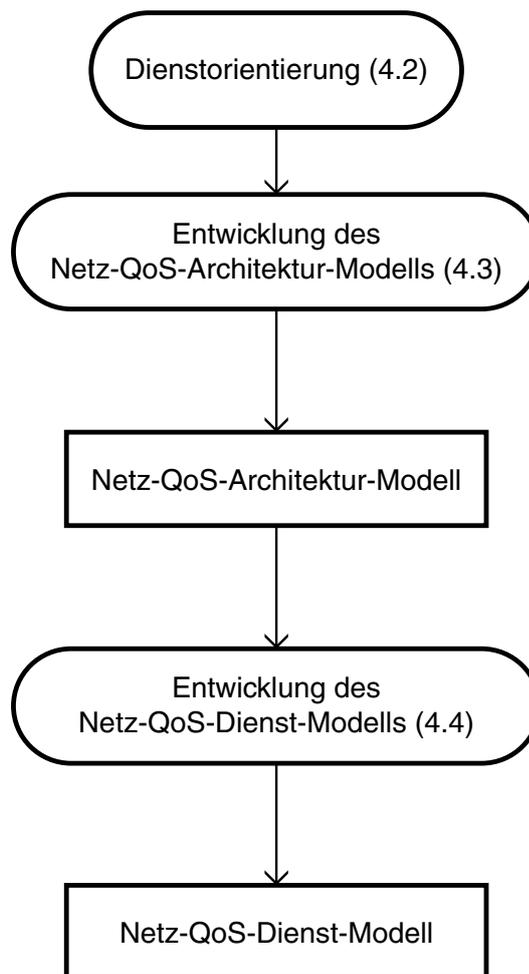


Abbildung 4.1: Aktivitätsdiagramm für das Vorgehen in diesem Kapitel

1. Hauptschritt: Normalisierung durch Modellierung

Ausgangspunkt des ersten Hauptschrittes sind die zu koppelnden Netz-QoS-Dienste. Sie werden durch Modellierung in eine normalisierte Darstellung überführt. Mit diesem Teil wird sich dieses Kapitel beschäftigen. Durch die Normalisierung ist die Grundlage gelegt, um mittels der generischen Methodik die fallspezifische Lösung gewinnen zu können, d.h. dort wird der eigentliche Koppelvorgang besprochen werden. Das nächste Kapitel wird diese Methodik erarbeiten.

4.1 LÖSUNGSDIEE UND VORGEHEN

Die Entwicklung der normalisierenden Modellierung in diesem Kapitel wird sich wiederum in Teilschritte untergliedern. Abbildung 4.1 stellt das Vorgehen als UML-Aktivitätsdiagramm [OMG 03-03-01]¹ dar:

- I) Ein grundlegender Ausgangspunkt dieser Arbeit ist die Ebene der Dienste. Der gewählte Lösungsansatz fokussiert also Netz-QoS-Dienste im Kontrast zu Netz-QoS-Architekturen. Die Dienstorientierung des Ansatzes fügt der Arbeit grundlegende Charakteristika hinzu, die im Teilschritt in Sektion 4.2 herausgearbeitet werden.
1. Teilschritt: Dienstorientierung zur Fokussierung
- II) Nach der Einordnung des Begriffs Netz-QoS-Dienst aus Sektion 2.2 basiert jeder Netz-QoS-Dienst auf genau einer Netz-QoS-Architektur. Abbildung 2.9, Seite 28 stellt dies auch bereits in einem Meta-Modell dar. Entsprechend wird im nächsten Teilschritt zunächst eine Modellierung von Netz-QoS-Architekturen vorgenommen (Sekt. 4.3) an dessen Ende ein Modell für Netz-QoS-Architekturen aus Dienstsicht steht.
2. Teilschritt: Modell für Netz-QoS-Architekturen

In der Anwendung bedeutet dies, dass bei mehreren Diensten, die auf derselben Architektur beruhen, deren Basis nur einmal modelliert werden muss.

- III) Der letzte Teilschritt in diesem Kapitel beschäftigt sich schließlich mit der Modellierung der eigentlichen Netz-QoS-Dienste. Dabei wird die im vorangegangenen Teilschritt gewonnene Architektur-Modellierung verfeinert, und die Möglichkeit geboten, diese um die für den individuellen Dienst spezifischen Aspekte zu erweitern. Das Ergebnis ist ein Modell für Netz-QoS-Dienste.
3. Teilschritt: Modell für Netz-QoS-Dienste

Formalisiert man die informelle Darstellung der Methodikanwendung aus Abbildung 2.15, Seite 41 ebenfalls in einem UML-Aktivitätsdiagramm, so erhält man die Darstellung aus Abbildung 4.2. Die beiden Netz-QoS-Dienste *A* und *B* sind jetzt als Objekte dargestellt. Der Schritt „dienstorientierte Modellierung“ aus Abbildung 2.15, Seite 41 ist entsprechend der Teilschritte verfeinert dargestellt.
Formalisierte Methodikanwendung

Im ersten Schritt werden die den Diensten zugrunde liegenden Netz-QoS-Architekturen α und β durch die Architektur-Modellierung normalisiert; in Abbildung 4.2 als *Modell α* und *Modell β* dargestellt. Im zweiten Schritt, der Dienstmodellierung nämlich, werden dann die Netz-QoS-Dienste *A* und *B* selbst unter Einbeziehung der normalisierten Architekturmodelle modelliert; man erhält *Modell A* und *Modell B*. Der sich daran anschließende Schritt der Methodikanwendung wird im nächsten Kapitel verfeinert werden.
Modellierung Architekturen
Modellierung Dienste

Abbildung 4.3 gibt einen gemeinsamen Überblick über die beiden Ebenen des Vorgehens. Dabei werden die Zusammenhänge zwischen den auftretenden Objekten mit den Begriffen aus Sektion 2.2 herausgearbeitet. In der mittleren und rechten Spalte sind die Vorgehensmodelle für Modellentwick-
Zusammenhänge zur Begriffsbildung

¹In dieser Arbeit wird die UML in der Version 1.5 verwendet. Die neue Version 2.0 ist zum Zeitpunkt der Erstellung der Arbeit zwar weit fortgeschritten, aber noch nicht endgültig verabschiedet.

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

lung und Anwendung des Modells dargestellt. Dazwischen ist ersichtlich, welches Modell in der Anwendung eine Instanz welchen Modellteils der generischen Modellierung ist. In der linken Spalte ist die Begriffsbildung platziert. Dabei sind jene Teile dargestellt, die von der generischen Modellierung abgebildet werden, sowie welche Teile der Anwendung Instanzen der jeweiligen Begriffe sind.

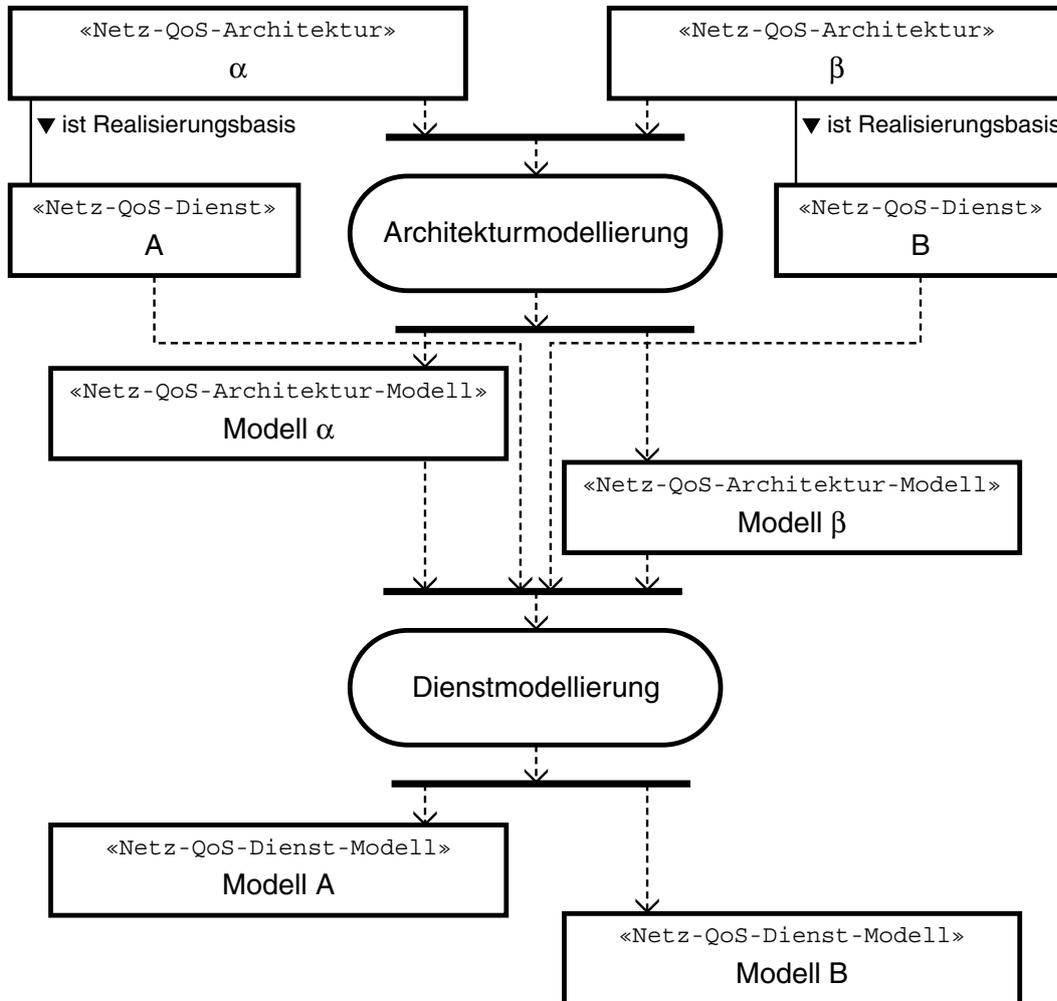


Abbildung 4.2: Aktivitätsdiagramm der Methodikanwendung

4.2 FOLGEN DER DIENSTORIENTIERUNG

Dem Vorgehensmodell aus der vorangegangenen Sektion folgend werden in dieser Sektion die Folgen der Dienstorientierung für die Modellierung eingehend betrachtet. Ausgehend von den Erkenntnissen aus dem MNM-Dienstmodell werden dann spezifische Eigenschaften für die dienstorientierte Modellierung von Netz-QoS-Architekturen und -Diensten herausgearbeitet.

4.2.1 Dienstorientierung anhand des MNM-Dienstmodells

Sichten im MNM-Dienstmodell	Das in Abschnitt 2.2.3 eingeführte MNM-Dienstmodell [GHHK+ 01, GHKR+ 01] hält neben des bereits bekannten Basis-Modells (<i>Basic Model</i>) (Abb. 2.7, S. 25) auch noch weitere sog. Sichten (views) bereit. Für ein gemeinsames Verständnis eines Dienstes auf Seiten von Dienstleister und
Dienstansicht des MNM-Dienstmodells	-nehmer wird der Dienst mittels der Dienstansicht (<i>Service View</i>) (Abb. 4.4) näher spezifiziert. Dabei zeigt die Dienstansicht detailliert die Grenzen zwischen den jeweiligen Seiten auf und ist somit geeignet, die Dienstorientierung für diese Arbeit näher zu untersuchen.
realisierungsunabhängige Eigenschaften	Dem Konzept der Dienstorientierung aus Dienstnehmersicht folgend, stehen realisierungsunabhängige Eigenschaften eines Dienstes bei der Dienstansicht im Vordergrund. Umgesetzt wird dies durch eine strikte Trennung zwischen Objekten auf Dienstleister / -nehmerseite und den davon unabhängigen Objekten. Hieraus ergibt sich die Unabhängigkeit des Dienstes an sich von seiner Realisierung.
Dienstnehmer / -leister im VoIP-Beispiel	Bezüglich des ATM-Dienstes aus dem Szenario von Abschnitt 2.1.1 bedeutet das, dass die ALV AG auf der Dienstnehmerseite steht und die Data-Systems auf der Dienstleisterseite.

4.2.1.1 Seitenunabhängiger Teil

Unterscheidung fundamentaler Funktionalitätsarten	Entsprechend den in [GHHK+ 01] durchgeführten Anforderungsanalyse wird ebenfalls die Unterscheidung der beiden fundamentalen Funktionalitätsarten vollzogen: Demnach besteht der Dienst (service) aus Nutzfunktionalität (usage functionality) und Managementfunktionalität (management functionality) sowie den an sie gestellten Qualitätsanforderungen. Beide Funktionalitätsarten müssen daher eine bestimmte Menge von QoS-Parametern einhalten. Die Klasse <i>QoS parameters</i> definiert dabei nur die Typen der QoS-Parameter und höchstens eine zu erbringende Mindestanforderung, damit der Service erfüllt ist.
---	--

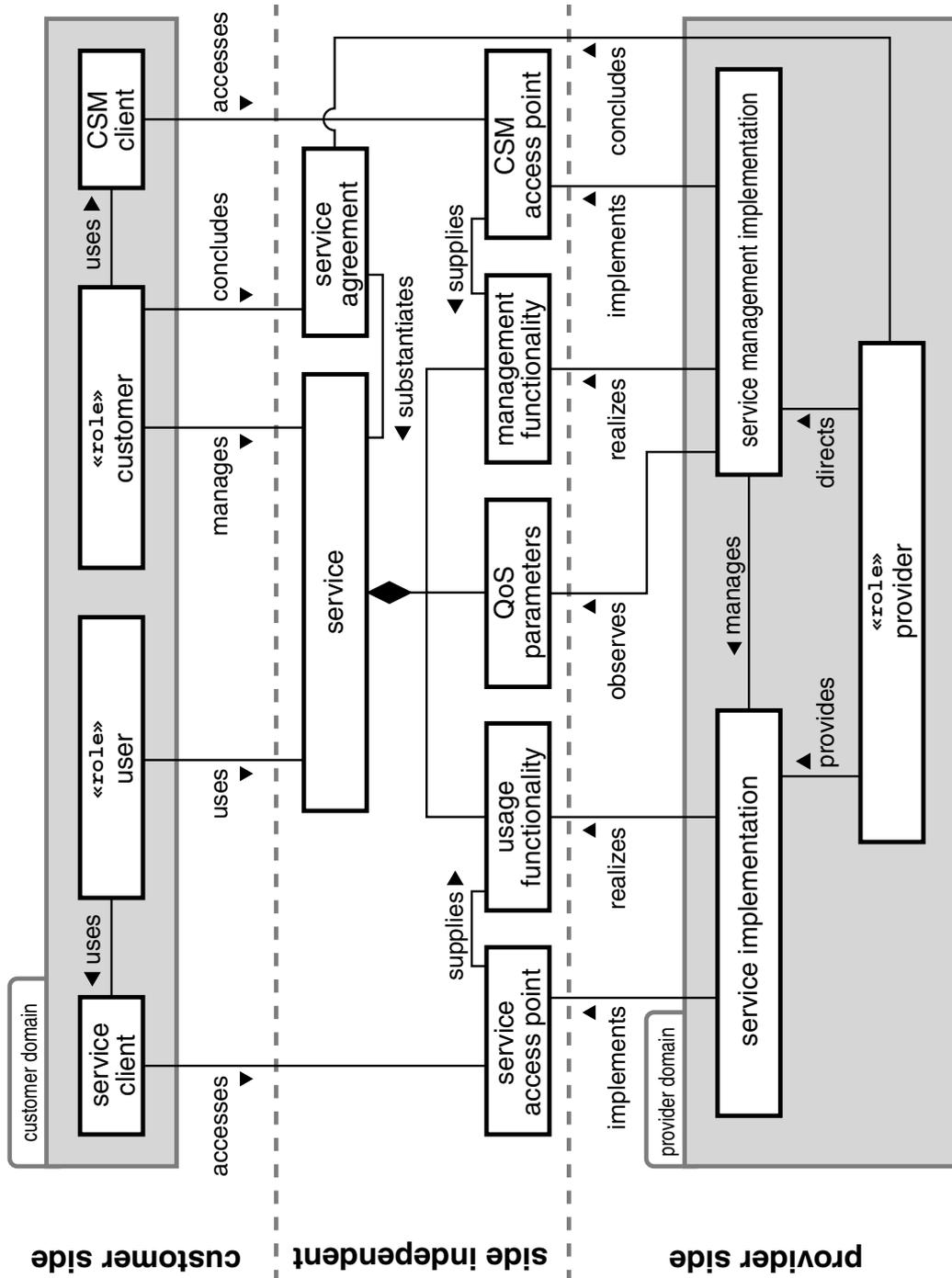


Abbildung 4.4: Dienstsicht (Service View) des MNM-Dienstmodells

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

Nutzfunktionalität	Die <i>usage functionality</i> stellt die eigentliche Dienstfunktionalität bereit. Sie umfasst alle konkreten Interaktionen zwischen Dienstnehmer- und Dienstleisterseite, die den Dienst ausmachen.
Nutzfunktionalität im VoIP-Beispiel	Die Nutzfunktionalität des ATM-Dienstes ist beispielsweise die Möglichkeit ATM-Verbindungen unter Bestimmung der QoS-Kategorie vom Standort M zum Standort L und umgekehrt aufzubauen. Die QoS-Parameter sind jene der verschiedenen ATM QoS-Kategorien aus Tabelle 2.1, Seite 21.
Managementfunktionalität	Interaktionen, die der Kunde zur Steuerung des Dienstes, zur Überwachung der Dienstqualität und Abrechnung und Bezahlung des Dienstes nutzt, sind dabei aber nicht Teil der Nutzfunktionalität der Dienstes, sondern werden durch die Managementfunktionalität (<i>management functionality</i>) abgedeckt.
Managementfunktionalität im VoIP-Beispiel	Für das ATM-Szenario könnte die Managementfunktionalität des Dienstes beispielsweise die Abrechnung der in Anspruch genommenen ATM-Verbindungen sein. Weiterhin sind Reporting-Funktionalitäten über in Anspruch genommene Verbindungen, über übertragene Datenmengen und über dienstnehmerseite Einhaltung der durch den ATM-Dienst verlangten Verkehrscharakteristika Beispiele von Managementfunktionalität.
Dienstzugriffspunkte bieten realisierungsunabhängig Schnittstellen	Mit den bisher vorgestellten Klassen ist der Dienst zwar realisierungsunabhängig spezifiziert, jedoch stehen dem Nutzer bzw. Kunden die Dienst- bzw. Managementfunktionalität noch nicht zum Zugriff zur Verfügung. Zu diesem Zweck werden Dienstzugriffspunkte spezifiziert, die wiederum realisierungsunabhängig Schnittstellen beschreiben, an denen die Funktionalität zugreifbar gemacht wird. Diese Schnittstellenbeschreibung kann in Abhängigkeit des Dienstes aus Dienstprimitiven, Protokollen, aber auch Hardware-Schnittstellen bis hin zu Steckerbelegungen bestehen.
Getrennte Zugriffspunkte für Nutz- / Managementfunktionalität	Entsprechend der Trennung zwischen Dienst- und Managementfunktionalität werden jeweils eigene Schnittstellen spezifiziert: Der Dienstzugangspunkt (<i>service access point</i>) für die Nutzfunktionalität und der sog. <i>customer service management (CSM) access point</i> [LaNe 00] für die Managementfunktionalität.
Dienstzugangspunkte im VoIP-Beispiel	Im Falle des ATM-Dienstes könnte der Dienstzugangspunkt für die Nutzfunktionalität über einen ATM-Standard zur Abbildung auf Medien der Bitübertragungsschicht spezifiziert werden, z.B. [ATM-PHY 01]. Würde die Managementfunktionalität Web-basiert angeboten werden, wäre die Spezifikation der <i>CSM access points</i> die URL des Web-Servers, sowie die für den Zugriff notwendigen Authentisierungsinformationen.
Dienstvereinbarung	Mit den bisher eingeführten Klassen besteht allerdings noch keine vertragliche Beziehung zwischen Kunden und Anbieterdomäne. Erst über eine Dienstvereinbarung (<i>service agreement</i>) wird eine vertraglich wirksame Beziehung zwischen Kunde und Anbieter beschlossen. In dieser Dienstverein-

barung werden, entsprechend der Definition des Dienstes, Nutz- und Managementfunktionalität aus Kundensicht beschrieben und den QoS-Parametern konkrete Wertebereiche zugeordnet, die durch den Dienstleister einzuhalten sind.

4.2.1.2 Dienstnehmerdomäne

Auf der Dienstnehmerseite finden sich analog zum Basis-Modell die beiden Rollen *customer* und *user* wieder. Der Nutzer des Dienstes nutzt diesen, indem er mittels eines **service client** über den *service access point* auf die Nutzfunktionalität zugreift. Der Kunde benutzt analog einen **customer service management (CSM) client** für den Zugriff auf die Managementfunktionalität des Dienstes, die über den *CSM access point* bereitgestellt wird. Weiterhin ist der Kunde, entsprechend seiner Rolle derjenige, der dem *service agreement*, z.B. in Form eines unterzeichneten Vertrages, zustimmt und damit die juristische Beziehung zum Dienst herstellt.

Fundamentalrollen Nutzer und Kunde „Clients“ für beide Funktionalitätsarten

Beide *clients* der Kundenseite können dabei physische Geräte wie Computer und Telefon, aber auch Software-Applikationen sein. Die *clients* müssen nur kompatibel zu physischen und logischen Aspekten der korrespondierenden *access points* sein. Aus Managementsicht ist entscheidend, dass die vollständige Verantwortung für die *clients* auf der Seite des Dienstnehmers liegt.

Ausprägung der „clients“

Für unser Beispiel des ATM-Dienst wäre der *service client* ein zu dem verwendeten Standard kompatibles Koppelement. Der *CSM client* wäre ein Web-Browser.

„Clients“ im VoIP-Beispiel

4.2.1.3 Dienstleisterdomäne

Die Hauptaufgabe auf Dienstleisterseite besteht darin, den Dienst zu realisieren, dem er in seiner Rolle als *provider* im *service agreement* zustimmt. Zu diesem Zweck stellt der Anbieter eine **Dienstimplementierung (service implementation)** bereit, die die Nutzfunktionalität realisiert und den Dienstzugangspunkt implementiert. Die *service implementation* ist dabei die Kombination aus Know-How, Personal, Hard- und Software, die benötigt wird, um den Dienst zu realisieren und ihn über den *service access point* zugreifbar zu machen.

Implementierung der Nutzfunktionalität

Neben der reinen Dienstfunktionalität zeichnet der Provider auch für das Management des Dienstes verantwortlich. Mittels seiner **Dienstmanagement-Implementierung (service management implementation)** managt er die eigentliche Dienstimplementierung und sorgt dabei für die Erbringung der in der Dienstvereinbarung festgelegten Leistungen. Dabei überwacht er die QoS-Parameter und sorgt für deren Einhaltung. Neben der reinen Vereinbarungssicherung dient die Dienstmanagement-Implementierung auch internen Zielen wie Optimierung des Betriebs, beispielsweise hinsichtlich Effizi-

Implementierung der Managementfunktionalität

enzsteigerung und Risikominimierung. Weiterhin implementiert die Dienstmanagement-Implementierung den *CSM access point* und realisiert die dort zugreifbare Managementfunktionalität.

4.2.1.4 Kernaussagen der Dienstsicht

Dienstsicht ist Verfeinerung von Basis-Modell	Vergleicht man das Basis-Modell mit der Dienstsicht, so erkennt man, dass alle Rollen des Basis-Modells in der Dienstsicht enthalten sind, ihre Assoziationen zum Dienst jedoch verfeinert wurden. Ebenfalls nun detailliert dargestellt ist der Dienst selbst in seinen Komponenten. Die für diese Arbeit wichtigen Kernaussagen der Dienstsicht des MNM-Dienstmodells sind:
Drei fundamentale Seiten	<ul style="list-style-type: none"> ■ In einer dienstorientierten Sichtweise gibt es drei Seiten. In der Dienstnehmerdomäne und der Dienstanbieterdomäne befinden sich realisierungsspezifische Komponenten. Die den Dienst definierenden Komponenten stehen unabhängig dazwischen.
Dienst unabhängig von Realisierung definiert	<ul style="list-style-type: none"> ■ Ein Dienst ist unabhängig von seiner Realisierung definiert. Er besteht aus der Nutzfunktionalität, der Managementfunktionalität und den für beide Funktionalitäten relevanten QoS-Parametern.
Dienst unabhängig von Zugriffspunkten	<ul style="list-style-type: none"> ■ Ein Dienst ist unabhängig von der Beschaffenheit der Zugriffspunkte. Werden die Zugriffspunkte eines Dienstes verändert ohne dabei die Funktionalitäten zu verändern, dann ist die Definition des Dienstes nicht verändert worden.

Basierend auf diesen Kernaussagen wird im nachfolgenden Abschnitt die von der in diesem Kapitel vorzunehmenden Modellierung einzunehmende dienstorientierte Sichtweise für Domänen im Netz-QoS-Dienstumfeld vorgenommen.

4.2.2 Reduktion auf Domänensicht

Dienstorientierung beschränkt Modellierungstiefe	Um in der nächsten Sektion die Modellierung von Netz-QoS-Architekturen vornehmen zu können, wird in diesem Abschnitt zunächst eine Grundlage zur Einschränkung der Modellierungstiefe gelegt. Diese Beschränkung motiviert sich in erster Linie aus den Betrachtungen zur Dienstorientierung aus dem vorangegangenen Abschnitt.
--	---

Ein typisches Beispiel für einen Netzverbund wurde bereits in Abschnitt 2.1.3 gezeigt. Stellt man ein solches Szenario etwas detaillierter dar, erhält man Abbildung 4.5. Darin sind neben den Anbietergrenzen sowohl von anderen Beteiligten wahrgenommene Komponenten als auch interne Komponenten abgebildet.

Domänenbegriff notwendig	Für eine dienstorientierte Sicht fehlt dem Szenario aus Abschnitt 2.1.3 zunächst der Begriff der Domäne. Um zu einer dienstorientierten Modellierung zu gelangen sind nach dem vorangegangenen Abschnitt die Domänengrenzen festzulegen. Betrachtet man die Dienstsicht (Abb. 4.4, S. 73) des MNM-
--------------------------	--

4.2 FOLGEN DER DIENSTORIENTIERUNG

- Dienstmodells so definieren sich die Domänengrenzen an zwei Charakteristika:
- | | |
|---|-------------------------------|
| I) Die Fundamentalrollen der jeweiligen Seite liegen in einer eigenen Domäne. | Jede Seite in eigenen Domänen |
| II) Der Dienst, der angeboten wird liegt außerhalb der Domänen. | Dienst selbst nicht in Domäne |

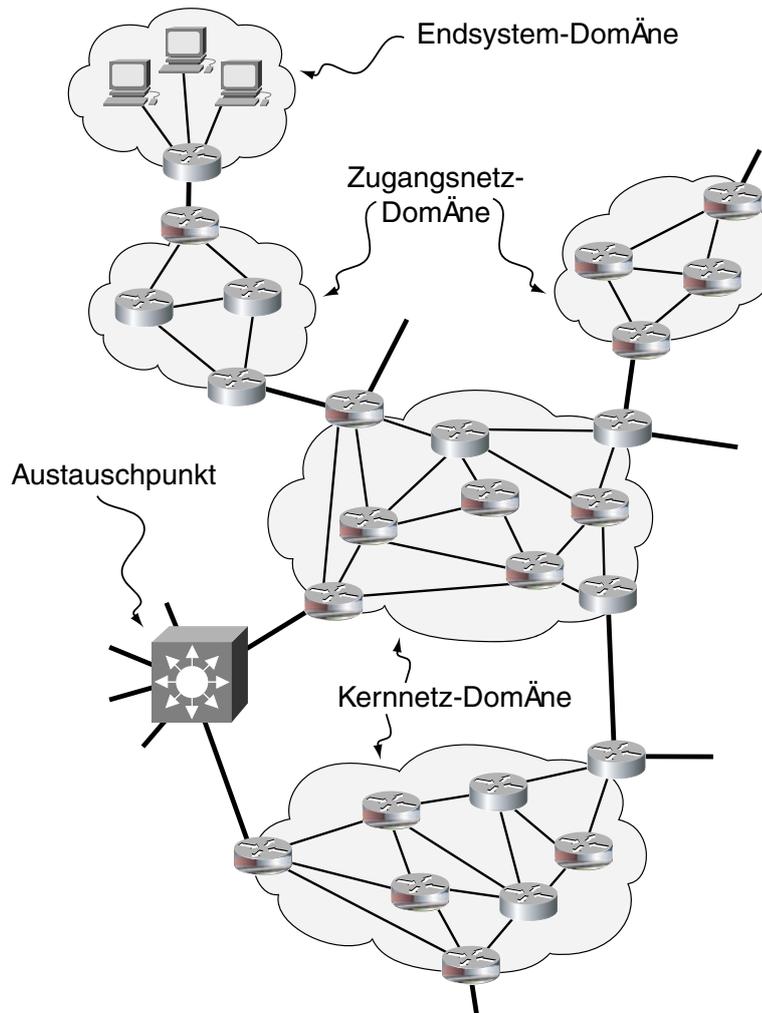


Abbildung 4.5: Exemplarische Detailsicht auf Netzverbund

Im vorliegenden Szenario treten die Beteiligten im Rahmen der Wertschöpfungskette jeweils in zwei verschiedenen Rollen auf. Bezüglich der Hierarchie sind sie nach oben Dienstanbieter, nach unten Dienstnehmer. Wegen dieser Dualität genügt zur Definition der Domänengrenzen die Definition der Dienstanbieterdomäne. Für das hier behandelte Gebiet der Netz-QoS-Dienste ergibt sich damit folgende Definition einer Domäne:

Rollendualität in Wertschöpfungskette

Die Anbieterdomäne eines Netz-QoS-Dienstansbieters definiert sich aus dem 2-Tupel aus Dienstanbieter (Provider) und dem von ihm angebotenen Dienst.

Definition Dienstanbieterdomäne

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

Detailfälle zur Definition Dienstanbieterdomäne	<p>Bietet ein und derselbe Provider mehrere unterschiedliche Dienste an, so sind diese nach der Definition als verschiedene Domänen zu betrachten. Der Fall, dass mehrere gleiche Dienste von unterschiedlichen Providern angeboten werden, führt ebenso zu mehreren unterschiedlichen Domänen. Zu beachten ist, dass dieser Domänenbegriff selbst keine geographische Ausdehnung impliziert. Diese wird allein durch den Bereich, auf den sich der angebotene Dienst erstreckt, gegeben. Entsprechend der Providerhierarchie sind so z.B. Domänen von Stadtgebietsgröße (PoP-Provider) bis hin zu weltumspannenden Domänen (Kernnetzprovider) möglich.</p> <p>Betrachtet man IP-Konnektivität als einen Dienst, findet sich in der Internet-Welt eine Entsprechung für den hier definierten Domänenbegriff: Ein Autonomous System (AS) umfasst ein oder mehrere Netze, die aus Sicht des globalen Routings als eine Einheit betrachtet werden. Die Größe oder die örtliche Ausdehnung des oder der Netze ist dabei unerheblich.</p>
Seitenunabhängiger Teil und Domänengrenze	<p>Nach dem MNM-Dienstmodell ist bei einem Dienst zwischen seiner Definition selbst, d.h. wie er dem Kunden angeboten wird (seitenunabhängiger Teil), und seiner Realisierung zu unterscheiden. Führt man diese Trennung für unseren Fall durch, resultiert daraus zunächst die Unterscheidung zwischen dem Äußeren oder Rand einer Domäne und ihrem Inneren.</p>
Kundensicht auf Domäne	<p>Ein Kunde kommt zunächst nur mit dem Rand einer Domäne in Kontakt. Dort befinden sich die sog. <i>Edge-Devices</i>. Diese sind Netzkomponenten (z.B. Router), die im Sinne des MNM-Dienstmodells einen Teil des Dienstzugangspunkt (<i>service access point</i>) der Nutzfunktionalität darstellen. Netzkomponenten im Inneren der Anbieterdomäne werden vom Kunden nicht wahrgenommen, sie sind Teil der Dienstimplementierung (<i>service implementation</i>). Damit entsteht die in Abbildung 4.6 gezeigte Black-Box-Sicht auf die Anbieterdomänen eines Netzverbundes. Interna der Anbieterdomäne sind hier ausgeblendet.</p>
Netzkomponenten im Inneren nicht wahrnehmbar	
Lage von Netzkomponenten alleine nicht ausreichend	<p>Allerdings genügt die Lage von Netzkomponenten relativ zum Domänenrand nicht, um die Black-Box-Sicht für Netz-QoS-Dienste präzise genug zu definieren, wie das folgende Beispiel zeigt.</p>
Parametrisierung innerer Router bei DiffServ aussen wahrnehmbar	<p>Für die DiffServ-Architektur sind die in den Routern verwendeten Algorithmen und deren Parametrisierungen häufig die Basis der Definition der von einem DiffServ-Dienst angebotenen Verkehrsklassen. Verändert man aber diese Algorithmen / Parametrisierungen auf Routern im Inneren einer Anbieterdomäne, dann hat das auch eine Änderung der vom Dienst angebotenen Verkehrsklassen zur Folge. Das Beispiel zeigt also, dass Koppelemente wie Router alleine keine zuverlässige Grundlage für eine Black-Box-Sicht auf den Dienst sind.</p>
Phänomenologische Black-Box-Sicht	<p>Für diese Arbeit ist also eine phänomenologische Definition der Black-Box-Sicht notwendig. Informell gesprochen bedeutet das, dass nur Vorgänge und Veränderungen im Inneren einer Anbieterdomäne, die keine Auswir-</p>

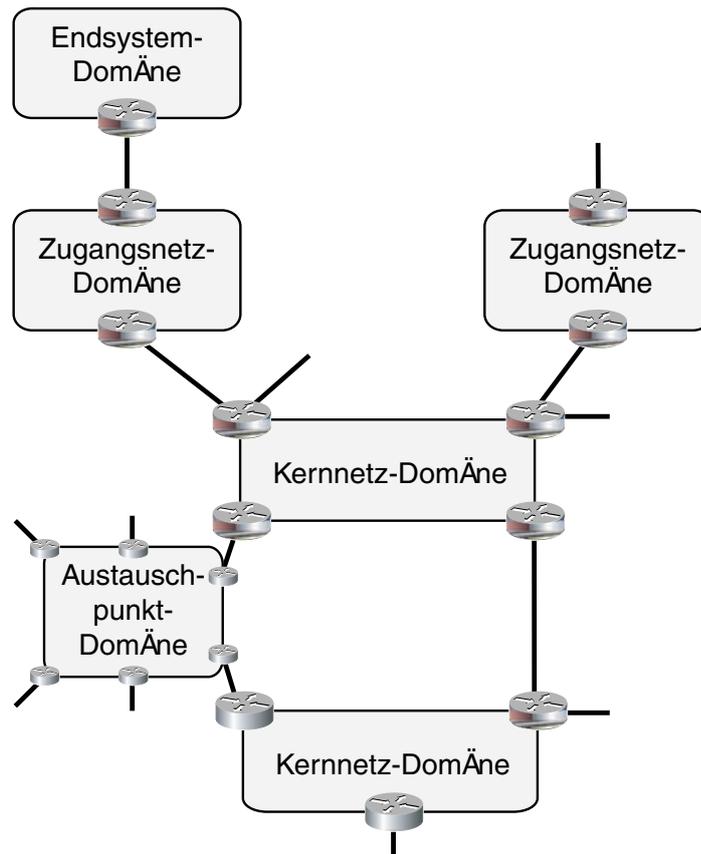


Abbildung 4.6: Black-Box-Sicht auf Domänen des Netzverbands aus Abb. 4.5

kung auf den Dienst haben, nicht weiter zu betrachten sind. Analog lässt sich dieser phänomenologische Ansatz auch auf die Grundlage der Netz-QoS-Dienste, nämlich die Netz-QoS-Architekturen, übertragen. Die exakte Festlegung der zu betrachtenden Eigenschaften einer Netz-QoS-Architektur im Sinne der dienstorientierten Black-Box-Sicht wird eine der wesentlichen Merkmale der nächsten Sektion sein.

Informell lassen sich aber bereits einige Beispiele dafür geben, was durch die Black-Box-Sicht verborgen werden wird:

■ Domänen-interne Dienste und Funktionalitäten

Hierbei handelt es sich um Dienste, die für einen Kunden nach aussen nicht sichtbar sind bzw. nicht angeboten werden; darunter fallen auch die internen Managementfunktionalitäten.

Verborgen:
Domänen-interne
Dienste und
Funktionalitäten

Ein Beispiel für eine interne Funktionalität eines Netz-QoS-Dienstes ist das gesamte interne Traffic Engineering. Ein weiteres Beispiel aus dem IP-Bereich sind Interior Routing Protokolle (z.B. Open Shortest Path First (OSPF) [RFC 2328] oder das Routing Information Protocol (RIP) [RFC 2453]).

■ Implementierung / Realisierung der nach außen angebotenen Dienste

Am MNM-Dienstmodell festgemacht sind dies alle Teile in den Klassen

Verborgen:
Realisierung der
Dienste

service implementation und *service management implementation* der Providerseite.

Für einen DiffServ-Dienst spielt es nach aussen beispielsweise keine Rolle, ob dieser klassisch mit einer reinen IP-Infrastruktur realisiert ist oder ob dabei MPLS als Basisarchitektur zum Einsatz kommt.

- Verborgene Koppellemente innerhalb der Domäne
- Koppellemente innerhalb der Domäne
Hiermit sind alle Switches, Router etc. gemeint, die im Inneren der Domäne verwendet werden. Insbesondere spielt z.B. deren Anzahl, Art und Hersteller keine Rolle.
- Andererseits sind bereits, wiederum nur exemplarisch, Punkte auszumachen, die auch im Weiteren eine Rolle spielen werden:
- Sichtbar: Angebotene Dienste
- Nach außen angebotene Dienste. Hieraus ergibt sich Notwendigkeit für ein dienstorientiertes Modell von Netz-QoS-Architekturen.
- Sichtbar: Koppelstellen
- Koppelstelle(n), an denen Dienste angeboten werden. Weitere Punkte, die dabei prinzipiell zu betrachten sind, da sie (potentiell) Einfluß nehmen auf die Koppelarchitektur:
 - ▼ Anzahl der Koppelstellen.
 - ▼ Örtliche Verteilung der Koppelstellen.
 - ▼ Kardinalitäten an den Koppelstellen
- Sichtbar: Stellung in Providerhierarchie
- Stellung des Anbieters in der Providerhierarchie. Dies hat Einfluss auf die Koppelarchitektur.

4.2.3 Zusammenfassung und Fokussierung der Modellierung

Zusammenfassend hat diese Sektion einige wichtige Festlegungen hinsichtlich der Dienstorientierung für die in der nächsten Sektion folgende Modellierung von Netz-QoS-Architekturen getroffen:

- Anbieterdomäne als Black-Box
- Die Anbieterdomäne wird für den Rest der Arbeit als Black-Box betrachtet. Interne Vorgänge werden keine weitere Rolle spielen. Auf einen kurzen Nenner gebracht heißt das, dass alles, was für den Kunden keine wahrnehmbare Auswirkung hat, nicht weiter betrachtet wird.
- Phänomenologische Sicht auf Netz-QoS-Architekturen
- Für Netz-QoS-Architekturen muss eine Modellierung gefunden werden, die eine phänomenologische Sicht auf sie ermöglicht. Dies bedeutet, dass nur solche Aspekte zu modellieren sein werden, die eine Auswirkung auf die Dienste haben, die auf die jeweilige Architektur aufbauen.
- Differenzierung intern / extern wichtige Aufgabe
- Die Differenzierung zwischen internen Komponenten und Vorgängen und solchen mit für den Kunden sichtbaren Effekten wird eine der wichtigsten Aufgaben der anstehenden Modellierung.

4.2 FOLGEN DER DIENSTORIENTIERUNG

Neben diesen Erkenntnissen ist es notwendig, einige weitere Festlegungen zu treffen, um das Koppelungsproblem im Rahmen dieser Arbeit bearbeiten zu können. Zur Schaffung eines Netzverbundes sind neben der Koppelung funktionaler Komponenten prinzipiell noch zwei weitere Voraussetzungen zu schaffen: gemeinsames Adressierungsschema und Wegewahl im Netzwerk.

Die Frage eines gemeinsamen Adressierungsschemas im Netzwerk wird im weiteren als gelöst betrachtet. Dies lässt sich aus zwei Gründen rechtfertigen. Zum einen stützen sich viele Netz-QoS-Architekturen auf den IP-Standard, dessen Adressierungsschema sich in globalen Netzen als de-facto Standard durchgesetzt hat. In diesem Fall stellen sich Fragen bzgl. des Adressierungsschemas also nicht. Zum anderen gibt es für andere Adressierungsschemen häufig bereits Umsetzungen, auf die dann zurückgegriffen werden kann.

Adressierungsproblematik wird als gelöst betrachtet

Das Problem der Wegewahl im Netzwerk unter Beachtung von QoS-Eigenschaften würde den Rahmen dieser Arbeit sprengen. Auf diesem Teilgebiet sind noch Forschungsarbeiten in einem ähnlichen Umfang notwendig, wie es zur Lösung der Koppelungsproblematik selbst der Fall ist. Allerdings sind hier bereits auch schon einige Ansätze vorhanden, z.B. im Rahmen des Border Gateway Protocol (BGP) [[RFC 1771](#)]: [[PHS 00](#), [CrJa 02](#), [XLWN 02](#), [XWLN 04](#), [Guti 04](#)]. Ohne Beschränkung der Allgemeinheit kann für diese Arbeit davon ausgegangen werden, dass (soweit überhaupt notwendig) ein sinnvolles Wegewahlverfahren existiert und funktionsfähig ist.

Wegewahl bleibt aussen vor

Zuletzt wird natürlich noch von einer weiteren, scheinbar trivialen, Randbedingung ausgegangen: Eine Netz-QoS-Architektur und darauf basierende Dienste funktionieren korrekt im Rahmen ihrer Spezifikation.

Dienste müssen funktionieren

Nachdem in dieser Sektion die Rahmenbedingungen für eine Modellierung aus Sicht der Dienstorientierung gesteckt worden sind, wird in der folgenden Sektion der zweite Schritt des Vorgehens aus Sektion [4.1](#) und Abbildung [4.1](#), Seite [68](#), nämlich die Modellierung der Netz-QoS-Architekturen, durchgeführt.

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

Dem Vorgehensmodell aus Abbildung 4.1, Seite 68 entsprechend, wird in dieser Sektion ein dienstorientiertes Modell für Netz-QoS-Architekturen entwickelt werden.

Will man Netz-QoS-Architekturen koppeln, muss zunächst kurz zusammengefasst die zentrale Frage beantwortet werden:

Schlüsselfrage zur Modellierung „ Was haben Netz-QoS-Architekturen gemeinsam und worin bestehen die spezifischen Unterschiede? “

Unterscheidung statischer und dynamischer Aspekte Der wichtigste Punkt zur Beantwortung dieser Frage wird die Unterscheidung zwischen statischen und dynamischen Aspekten von Netz-QoS-Architekturen sein. Im Verlauf der Sektion wird deutlich werden, dass dabei ...

- ... die statischen Aspekte die strukturellen Gemeinsamkeiten aller Netz-QoS-Architekturen bestimmen.
- ... die dynamischen Aspekte der Modellierung die wesentlichen Unterschiede zwischen verschiedenen Netz-QoS-Architekturen offenlegen.

Mit einer universell anwendbaren Darstellung der beiden Aspekte wird die normalisierende Modellierung geschaffen, die Ausgangspunkt für die generische Methodik zur Koppelung ist (Sekt. 4.1).

Schmitts Arbeit als Hilfsmittel Wie bereits in Abschnitt 3.2.1 festgestellt wurde, hat Schmitt in [Schm]B 01 ein grundlegendes Werk im Bereich Netz-QoS geschaffen. Obwohl es, wie ebenfalls schon festgestellt, bezüglich der für diese Arbeit gestellten Anforderungen nicht ausreichend ist, kann es als Hilfsmittel für die nun vorzunehmende Modellierung dienen. Entsprechend werden in den einzelnen Teilbereichen immer wieder Bezüge zu Schmitts Arbeit hergestellt werden.

Zur Veranschaulichung der Modellierung werden schwerpunktmäßig wieder die Netz-QoS-Architekturen ATM und DiffServ herangezogen.

4.3.1 Statisches Grundmodell für Netz-QoS-Architekturen

statische / dynamische Teile des Grobmodells Im Rahmen der Einordnung von Netz-QoS-Architekturen in Sektion 2.2 wurde ein Grobmodell erstellt (Abb. 2.6, S. 23). Die Bestandteile des Grobmodells lassen sich nach statischen und dynamischen Teilen unterscheiden:

- Statisch: Daten, QoS-Kategorien, QoS-Parameter
- Dynamisch: Steuerung, Prozeduren, Workflows

Der grundsätzliche Aufbau des statischen Modells aus Abbildung 4.7 folgt zunächst aus den Komponenten des Grobmodells aus Sektion 2.2. Auffällig

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

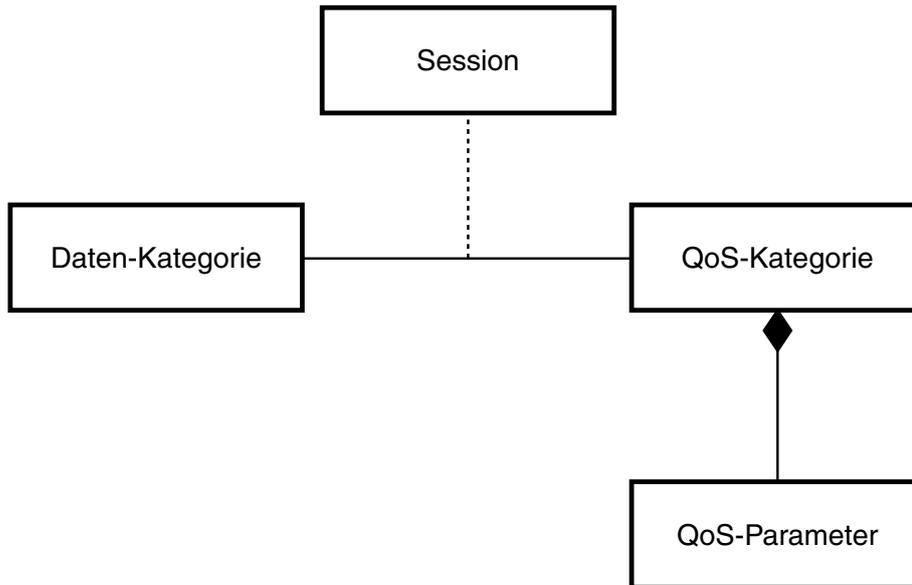


Abbildung 4.7: Statisches Grundmodell für Netz-QoS-Parameter

ist die strikte Trennung zwischen den Klassen *Daten-Kategorie* und *QoS-Kategorie*. Dabei modelliert *Daten-Kategorie* die Eigenschaften des Datentransports, die nicht mit Qualitätsmerkmalen behaftet sind. Die Klasse *QoS-Kategorie* fasst *QoS-Parameter* zusammen, die auf eine *Daten-Kategorie* angewendet werden können. Für diese Trennung gibt es mehrere Gründe:

Trennung zwischen Daten-Kategorie und QoS-Kategorie

- Bei der Entwicklung einer QoS-Architektur kann die Spezifikation der Eigenschaften von *Daten-Kategorien* und *QoS-Kategorien* unabhängig voneinander erfolgen. In bestehenden Netz-QoS-Architekturen ist dies historisch bereits so erfolgt.

Unabhängige Spezifikation von Daten- und QoS-Eigenschaften

Die beiden QoS-Architekturen DiffServ und IntServ aus der Internet-Welt haben genau diese Entwicklung durchgemacht. Beide sind Erweiterungen des IP-Protokolls, das die *Daten-Kategorie* beider Architekturen stellt. Für jede der beiden Architekturen wurden dann jeweils unterschiedliche *QoS-Kategorien* spezifiziert.

- Auch Schmitt nimmt in [Schm]B 01 diese Trennung vor. In seinem konzeptuellen ER-Modell (Abb. 3.1, S. 52) finden sich die beiden Teile *Control Path* und *Data Path*. Sie repräsentieren genau diese Trennung.
- Die Trennung stellt auch eine Modularisierung in der Modellierung dar, die in der späteren Anwendung den Vorteil der Wiederverwertbarkeit in sich birgt.

Schmitt nimmt Trennung bereits vor

Modularisierung bringt Wiederverwendbarkeit

Sollen beispielsweise die beiden Architekturen DiffServ und IntServ modelliert werden, dann muss ihre gemeinsame *Daten-Kategorie* (nämlich IP, siehe obiges Beispiel) nur einmal modelliert werden.

Bevor im Anschluss die einzelnen Teile des statischen Grundmodells erläutert werden folgen zunächst einige kurze Anmerkungen zu den benutzten Darstellungen in UML.

4.3.1.1 Modellierungstechnik mit UML

UML-
Klassendiagramme
für statisches
Grundmodell

Zur Darstellung des statischen Grundmodells werden statische Diagramme der UML [OMG 03-03-01], die auch als Klassendiagramme bezeichnet werden, benutzt. Es handelt sich hierbei um die Fortsetzung der Darstellungstechnik, wie sie bereits für das MNM-Dienstmodell z.B. in den Abbildungen 2.7 (S. 25) und 4.4 (S. 73) verwendet worden ist. Es sei an dieser Stelle nochmals betont, dass es sich bei diesen Diagrammen nicht um Klassendiagramme handelt, die für eine unmittelbare Umsetzung in Klassen einer objektorientierten Programmiersprache vorgesehen sind.

Kompatibilität mit
dem MNM-
Dienstmodell

Die von Schmitt benutzte ER-Modellierung dagegen ist nicht mächtig genug, insbesondere für die später wichtige Darstellung dynamischer Zusammenhänge. In dieser Arbeit wurde UML u.a. deshalb gewählt, weil es Darstellungsoptionen sowohl für statische als auch für dynamische Zusammenhänge bietet und weil mit UML die Darstellung kompatibel zum für diese Arbeit wichtigen MNM-Dienstmodell gehalten wird.

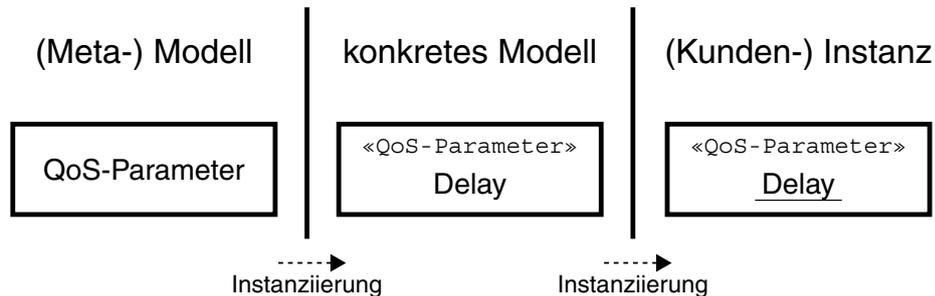


Abbildung 4.8: Ebenen der Modellierung

Modellierungsebenen

Für die Modelle in dieser Arbeit werden drei Ebenen verwendet, die für das statische UML-Diagramm in Abbildung 4.8 anhand der Klasse *QoS-Parameter* dargestellt sind:

■ Ebene des generischen (Meta-) Modells

Generisches Meta-Modell

Auf dieser Ebene befindet sich das in dieser Sektion entwickelte generische Modell für Netz-QoS-Architekturen und das Modell für Netz-QoS-Dienste aus der folgenden Sektion 4.4. Die Entitäten des Modells werden als Klassen des statischen UML-Diagramms abgebildet. Zur Darstellung der verschiedenen Beziehungen zwischen den Modellentitäten werden die von der UML für diesen Diagrammtyp vorgesehenen allgemeinen und spezialisierten Assoziationen verwendet.

Die in Abbildung 4.1, Seite 68 enthaltenen Klassen befinden sich auf dieser Modellierungsebene.

■ Ebene des konkreten Modells

Wendet man das generische Modell auf eine konkrete Netz-QoS-Architektur an, so wird eine Instanz des generischen Modells erzeugt. Der Zusammenhang zwischen einer Klasse im Meta-Modell und einer Klasse im konkreten Modell, die eine Instanz des Meta-Modells ist, wird durch einen entsprechenden Stereotyp dargestellt, der den Namen der Klasse im Meta-Modell trägt. Anhand dieser Stereotypen lassen sich Diagramme des Meta-Modells von solchen einer konkreten Modellierung unterscheiden.

Modell für konkrete Architekturen

In dieser Sektion werden zur Illustration wiederholt konkrete Modellteile für die Standardbeispiele ATM und DiffServ vorgestellt.

Im Aktivitätsdiagramm aus Abbildung 4.2, Seite 70 wird ebenfalls diese Kennzeichnung verwendet.

■ Ebene der (Kunden-) Instanz

Sollen bestimmte Netz-QoS-Architekturen in der konkreten Anwendung im Kundenumfeld gezeigt werden, dann kommt die dritte Modellierungsebene zum Einsatz. Die dabei erzeugten Instanzen der Klassen eines konkreten Modells werden durch Unterstreichungen gekennzeichnet; in der UML-Terminologie gesprochen werden sie als Objekte dargestellt. Diese Darstellung ist analog zur Beziehung zwischen Klassen und Objekten in objektorientierten Programmiersprachen.

Modellinstanzen für einzelne Kunden

4.3.1.2 Daten-Kategorie

Die *Daten-Kategorie* beschreibt jene Eigenschaften des Datentransports, die nicht mit Qualitätsmerkmalen behaftet sind, aber bei der Koppelung dennoch eine Rolle spielen. Von Kategorien wird deshalb gesprochen, da prinzipiell denkbar ist, dass im Rahmen einer Netz-QoS-Architektur unterschiedliche Ausprägungen der Datenkommunikation spezifiziert sind. Abbildung 4.9 zeigt die nähere Spezifikation der Klasse *Daten-Kategorie* aus dem statischen Grundmodell, die nun besprochen wird.

„Qualitätslose“ Eigenschaften des Datenstroms

Zunächst werden die zwei grundsätzlichen Eigenschaften „verbindungsorientiert / verbindungslos“ und „Paket / SDU²-Grenzen erhaltend“ besprochen; anschließend die unterschiedlichen Ausprägungen der Kommunikationsrelationen. Vervollständigt werden die Betrachtungen im Abschnitt „Inhaltsbezogene Eigenschaften“, insbesondere bezüglich solcher Eigenschaften, die mit „verbindungsorientiert“ häufig impliziert werden.

Einzelne Bestandteile der Daten-Kategorie

Eigenschaft: verbindungsorientiert / verbindungslos

Kommunikationsbeziehungen können prinzipiell zustandsbehaftet oder zustandslos sein. Für Datenkommunikationsbeziehungen wird diese Eigenschaft als verbindungsorientiert bzw. verbindungslos bezeichnet. Diese Ei-

Verbindungsorientierung

²Service Data Unit (SDU)

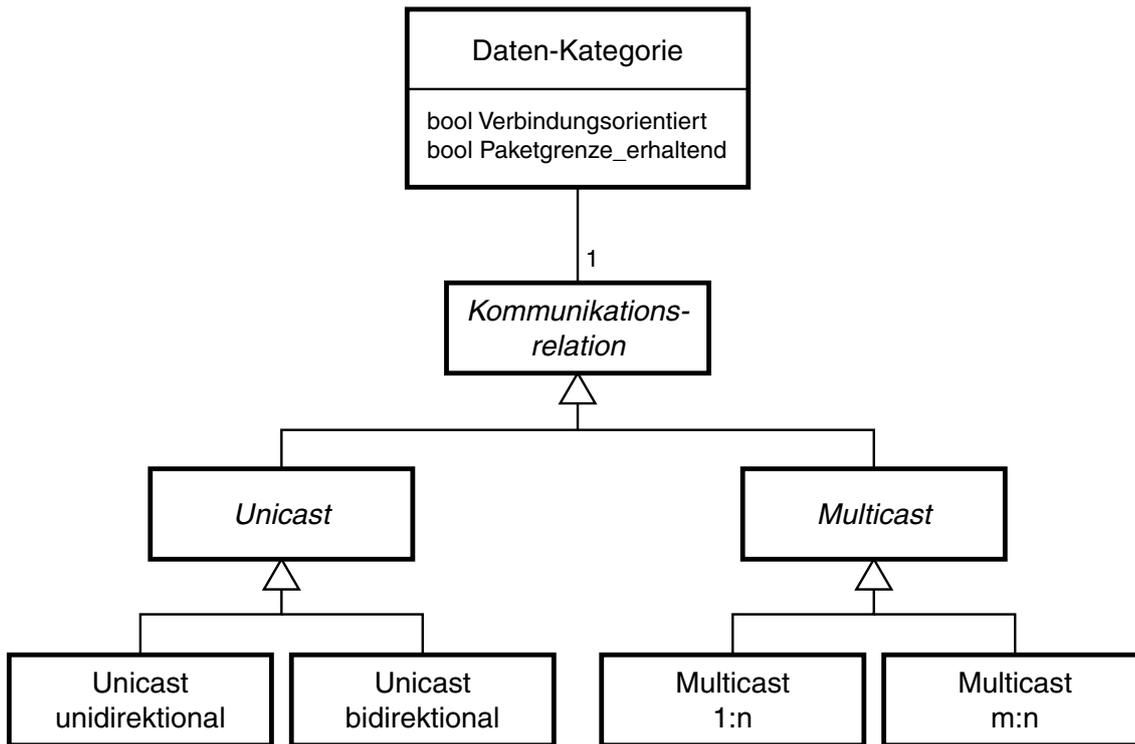


Abbildung 4.9: Klasse *Daten-Kategorie* des statischen Grundmodells

genschaft wird im Modell als boolesches Attribut *Verbindungsorientiert* dargestellt. Eine Belegung mit *false* wird entsprechend als verbindungslos, *true* als verbindungsorientiert aufgefasst.

Verbindungsorientierung nicht zwangswise identisch bei QoS-Kategorien

Zu beachten ist hier, dass sich dieses Attribut rein auf die Datenkommunikation bezieht. Dies impliziert nicht zwangswise, dass *QoS-Kategorien* ebenfalls die gleiche Eigenschaft tragen. In der IntServ-Architektur beispielsweise wird der Datentransport mittels IP-Datagrammen realisiert, damit ist die *Daten-Kategorie* verbindungslos bzw. zustandslos. Die Vereinbarung der Qualitätseigenschaften erfolgt über das Resource Reservation Setup Protocol (RSVP) [RFC 2205, RFC 2208, RFC 2209], womit sie im Gegensatz zur Datenkommunikation zustandsbehaftet ist.

Eigenschaft: Paket / SDU-Grenzen erhaltend

Erhaltung der Paketgrenzen

Häufig sind verbindungsorientierte Protokolle als sog. Strom-Protokolle ausgelegt. Sie sind u.a. dadurch gekennzeichnet, dass Pakete bzw. allgemeiner die Grenzen einer Service Data Unit (SDU), wie sie vom Sender übergeben worden sind, beim Empfänger nicht mehr nachvollzogen werden können. Bei den sog. Datagramm-Protokollen dagegen können die Paketgrenzen beim Empfänger wieder nachvollzogen werden. Da Protokolle höherer Schichten sich auf die Erhaltung der Paketgrenzen verlassen könnten, muss diese Eigenschaft im Netzverbund evtl. nachvollzogen werden. Prinzipiell ist aber denkbar, dass unabhängig von verbindungsorientiert / -los die Paketgrenzen, wie sie an den Dienst übergeben wurden, erhalten bleiben oder auch nicht. Diesem Umstand wird durch ein eigenes boolesches Attribut *Pa-*

Paketgrenzenerhaltung unabhängig von Verbindungsorientierung

ketgrenze_erhaltend Rechnung getragen.

Typische Vertreter für Paketgrenzen erhaltende Kategorien sind die Datagramm-Protokolle (z.B. User Datagram Protocol (UDP) [RFC 768]), Paketgrenzen nicht erhaltend sind Strom-Protokolle (z.B. Transmission Control Protocol (TCP) [RFC 793]).

Eigenschaft: Kommunikationsrelation

Daten-Kategorien können in ihren Kommunikationsrelationen unterschiedlich gestaltet sein. Hierbei sind zunächst die beiden Hauptvarianten zu unterscheiden: Kommunikationsrelationen

- **Unicast:** Datenaustausch zwischen genau zwei Kommunikationspartnern.
- **Multicast:** Datenaustausch zwischen mehr als zwei Kommunikationspartnern.

Bei beiden Hauptvarianten gibt es jeweils noch zwei Untervarianten:

- **Unicast unidirektional:** Datenstrom in nur eine Richtung. Die Kommunikationspartner können explizit als Sender und Empfänger bezeichnet werden.
- **Unicast bidirektional:** Datenströme in beide Richtungen. Beide Kommunikationspartner können als Sender und Empfänger auftreten.
- **Multicast 1:n:** Gerichtete Gruppenkommunikation, Daten werden ausschließlich von genau einem Teilnehmer zu den anderen Teilnehmern versandt. Dieser Teilnehmer kann eindeutig als Sender identifiziert werden, alle anderen sind Empfänger.
- **Multicast m:n** Ungerichtete Gruppenkommunikation, Daten können von allen Teilnehmern an die jeweils anderen Teilnehmer versandt werden. Alle Teilnehmer können in den beiden Rollen Sender und Empfänger auftreten.

Zwar sind die Unicast-Varianten letztlich Spezialfälle der Multicast-Varianten, und könnten deshalb im Sinne einer generischen Modellierung theoretisch weggelassen werden. In existierenden Netz-QoS-Architekturen unterscheiden sich die beiden Hauptvarianten häufig aber so drastisch, dass dieser Unterschied im Modell explizit nachvollzogen werden soll.

Explizite Unterscheidung Uni- / Multicast

Im Modell in Abbildung 4.9, Seite 86 ist eine *Daten-Kategorie* mit genau einer *Kommunikationsrelation* assoziiert. Die beschriebene hierarchische Struktur der Kommunikationsrelationen ist im Modell durch eine gleichlautende Verfeinerung dargestellt. Die Klasse *Kommunikationsrelation* sowie die Klassen der Hauptvarianten (*Unicast* und *Multicast*) sind als abstrakte Klassen modelliert (Klassenbezeichner kursiv gesetzt), während die vier weiteren Ausprägungen nicht-abstrakt modelliert sind. Dies zeigt an, dass in einer Instanzierung des Modells für eine konkrete Netz-QoS-Architektur eine Ausprägung von *Daten-Kategorie* nur mit einer der vier nicht-abstrakten Klassen assoziiert werden darf.

Modellierung der Kommunikationsrelation

Inhaltsbezogene Eigenschaften

Inhaltsbezogene Eigenschaften In der klassischen Literatur, wie z.B. [HAN 99, Tane 03], werden zum Datentransport noch weitere Eigenschaften genannt. Dabei handelt es sich um Eigenschaften, die den Inhalt des transportierten Datenstroms beeinflussen. Die folgenden Beispiele stammen aus dem Bereich des Fehlermanagements:

- Verfälschungsgesichert: Daten werden inhaltlich nicht verfälscht (typ.: Bitfehler).
- Reihenfolngesichert: Daten bzw. Datenpakete erreichen den Empfänger in der gleichen Reihenfolge wie sie abgesendet wurden.
- Verlustgesichert: Es gehen keine Daten(pakete) verloren.
- Duplikatgesichert: Daten(pakete) erreichen den Empfänger höchstens einmal.

Attribute müssen unabhängig voneinander betrachtet werden Häufig werden gerade mit dem Attribut „verbindungsorientiert“ auch die Eigenschaften „reihenfolngesichert“, „verlustgesichert“ und „duplikatgesichert“ verstanden. Bei klassischen Stromprotokollen wie TCP [RFC 793], wird diese Implikation häufig gemacht. Im Sinne einer generischen Modellierung müssen diese Attribute allerdings unabhängig voneinander betrachtet werden. Beispielsweise ist es durchaus denkbar, verbindungsorientierte *Daten-Kategorien* zu haben, die aber keine Garantien hinsichtlich eines Paketverlustes bieten. Gerade bei Datendiensten mit harten Dienstgütezusicherungen kann es notwendig werden, extreme Überlastsituationen durch Verwerfen von Paketen (*dropping*) zu bereinigen.

Inhaltsbezogene Eigenschaften als QoS-Parameter In dieser Arbeit werden zur Klasse *Daten-Kategorie* ausschließlich Eigenschaften gezählt, die den Inhalt der Daten **nicht** betreffen. Solche Eigenschaften, die Veränderungen am Inhalt des übertragenen Datenstroms bewirken, sind somit als Qualitätseigenschaften aufzufassen und werden den *QoS-Parametern* zugeordnet. Insbesondere werden die bereits explizit erwähnten inhaltsbezogenen Eigenschaften „reihenfolngesichert“, „verlustgesichert“ und „duplikatgesichert“ später als Spezialisierung der QoS-Parameter-Unterklasse *Korrekttheitsparameter* modelliert werden (siehe auch Abb. 4.13, S. 94).

Vermeidung von Anomalien in Modellen Diese Art der Modellierung, die zwar zunächst unorthodox vor dem Hintergrund der klassischen Literatur erscheint, ist aber in dieser Weise notwendig um schwerwiegende Anomalien bei der Modellierung konkreter Netz-QoS-Architekturen zu vermeiden. Würde man nämlich z.B. das Attribut „verlustgesichert“ in der Klasse *Daten-Kategorie* belassen, und modellierte man dann eine Architektur, in der sich eine Paketverlustrate als *QoS-Parameter* spezifizieren ließe, so ergäbe dies ein Modell, das die gleiche Eigenschaft an zwei völlig unterschiedlichen Stellen in zwei unterschiedlichen Ausprägungen modellierte. Das im weiteren Verlauf der Arbeit noch modellierte DiffServ mit Assured Forwarding (AF) Per Hop Behaviour (PHB) ist ein Beispiel hierfür. Um solchen Anomalien vorzubeugen, werden alle inhaltsbezo-

genen Eigenschaften als *QoS-Parameter* modelliert. Dort können diese dann individuell verfeinert werden und so Anomalien vermieden werden.

Daten-Kategorie im Modell von Schmitt

Im konzeptuellen Modell von Schmitt ist der *Data Path* ein Teil der hier modellierten *Daten-Kategorien*. Schmitts *Data Path* umfasst folgende Aufgaben [SchmJB 01, S.21ff]: Aufgaben nach Schmitt

- I) Packet Classification / Stamping: Zuordnung von QoS-Eigenschaften zum Datenverkehr.
- II) Packet Scheduling / Bandwith Sharing: Beschäftigt sich mit der Entscheidung, welches das nächste zu sendende Paket aus einer Reihe von gepufferten Paketen ist, und entscheidet deshalb auch über die Nutzung der zur Verfügung stehenden Bandbreite.
- III) Buffer Management: Während „Packet Scheduling / Bandwith Sharing“ sich mit der Nutzung einer Leitung beschäftigt, übernimmt das Buffer Management die Verwaltung von Pufferkapazitäten in den Kopplungskomponenten.
- IV) Traffic Policing / Shaping / Marking: Diese Komponente sorgt dafür, dass der Datenverkehr den vereinbarten Charakteristika entspricht, wenn er ein Netz betritt oder verlässt.
- V) Adaptivity / Reactivity: Mechanismen die auf Endsystemen zum Einsatz kommen, um auf sich verändernde Verhältnisse im Netz zu reagieren (z.B. Explicit Congestion Notification (ECN) [RFC 3168]).

Aus Schmitts Modell spielen die Punkte II bis V in dieser Arbeit keine Rolle, da es sich um Aufgaben handelt, die ...

- ... entweder im Inneren einer Domäne zur korrekten Funktion des Dienstes zu erbringen sind und durch die eingenommene Black-Box-Sicht keine Rolle spielen, Keine Internen Aufgaben
- ... oder auf den Endsystemen zu erbringen sind, die ebenfalls nicht Teil der hier angestellten Betrachtungen sind. Keine Endsysteme

Punkt I „Packet Classification / Stamping“ dagegen spielt sehr wohl eine Rolle in dieser Arbeit. Er wird aber nicht als Teil der Klasse *Daten-Kategorie* aufgefasst, sondern wird explizit durch die Assoziationsklasse *Session* behandelt, die die Verbindung zwischen einer *Daten-Kategorie* und einer *QoS-Kategorie* qualifiziert. Die Beschreibung dieser Assoziationsklasse erfolgt später.

Modellierung von Daten-Kategorien im Beispiel

Nachfolgend noch ein Beispiel zur Modellierung der *Daten-Kategorie* in Diff-Serv. Da es sich nun um ein konkretes Modell handelt, werden die Untereinheiten entsprechend Abbildung 4.8, Seite 84 als Stereotypen notiert. Stereotypen im konkreten Modell

Die «Daten-Kategorien» der DiffServ-Architektur basieren auf dem IP-Protokoll [RFC 791]. Abbildung 4.10 zeigt dessen Mo-

dellierung. Das IP-Protokoll ist ein Datagramm-Protokoll, das verbindungslos die versendeten Datagramme transportiert, ohne deren Paketierung zu verändern. Entsprechend sind die Attribute *Verbindungsorientiert* und *Paketgrenze_erhaltend* konstant (*static*) auf *false* bzw. *true* gesetzt. Diese gemeinsamen Attribute werden in der abstrakten Oberklasse *IP_Datagramm* dargestellt. IP kennt die zwei Varianten *Unicast bidirektional* und *Multicast m:n* als Kommunikationsrelationen. Für diese Varianten sind von der abstrakten Oberklasse die beiden Klassen *IP_Unicast* und *IP_Multicast* abgeleitet und mit den jeweiligen Klassen für die Kommunikationsrelation assoziiert.

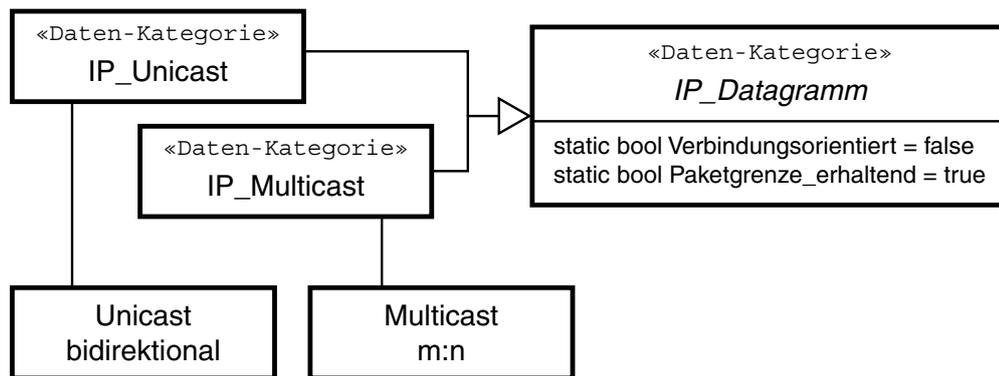


Abbildung 4.10: IP Daten-Kategorien von DiffServ

4.3.1.3 QoS-Parameter

Modellierung einzelner Qualitätseigenschaften Eine der Kernfunktionalitäten von Netz-QoS-Architekturen ist die Möglichkeit der Festlegung von Qualitätseigenschaften und deren Durchsetzung. Mit der Klasse *QoS-Parameter* werden die einzelnen Qualitätseigenschaften modelliert. Sie ist in Abbildung 4.11 dargestellt, ihre Teile werden in den nachfolgenden Absätzen erläutert.

Eigenschaft: Semantik

Semantik eines QoS-Parameters Um *QoS-Parameter* in einer Koppelung vergleichen zu können, ist ein exaktes Verständnis der *Semantik* eines *QoS-Parameters* erforderlich. Dass hierfür allein der Name eines Parameters nicht ausreichend ist, veranschaulicht ein Beispiel:

Ein QoS-Parameter, der mit „Delay“ bezeichnet wird, kann für *Daten-Kategorien* mit der Kommunikationsrelation *Unicast bidirektional* oder *Multicast m:n* grundsätzlich schon zwei verschiedene Bedeutungen haben:

- „Oneway Delay“: Die Verzögerung, die zwischen einem Sender und einem Empfänger auf einfachem Weg auftritt.
- „Roundtrip Delay“: Die Verzögerung, die zwischen einem Sender und einem Empfänger auf Hin- und Rückweg auftritt.

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

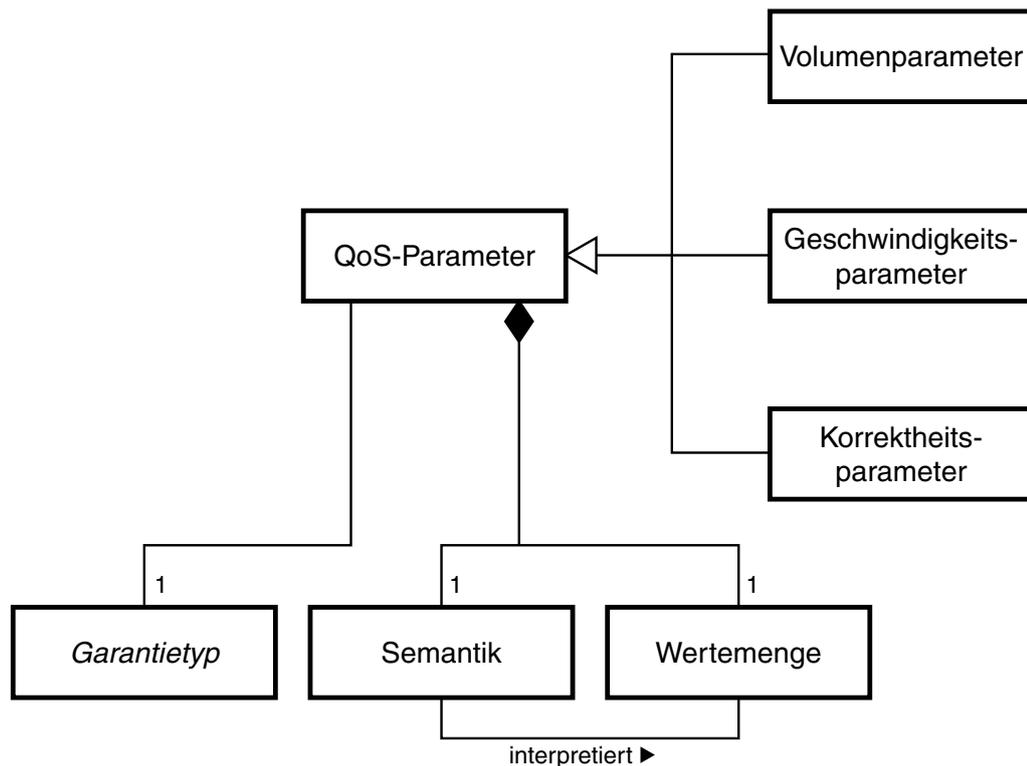


Abbildung 4.11: Klasse *QoS-Parameter* des statischen Grundmodells

Selbst wenn diese Unklarheit ausgeräumt ist, wird die Bedeutung von „Delay“ noch durch die Punkte bestimmt, an denen die Messungen zu seiner Bestimmung vorgenommen werden. Die Messpunkte können beispielsweise zwischen den letzten Routern, zwischen den Endsystemen oder zwischen den Applikationen sein. Unmittelbar vergleichbar sind Werte für „Delay“ nur dann, wenn sie zwischen den gleichen Referenzpunkten gemessen wurden, d.h. durch die gleiche Messvorschrift bestimmt wurden.

Für eine konkrete Modellierung sollte im optimalen Fall auf eine möglichst formalisierte Spezifikation der Parameter zurückgegriffen werden, sofern diese nicht bereits durch das der konkreten Architektur zugrunde liegende Spezifikationsdokument geschieht. Dies erleichtert die spätere Koppelung erheblich.

Formale Spezifikation

Wie das obige Beispiel auch gezeigt hat, sind *Semantik* eines *QoS-Parameters* und dessen Messung eng miteinander verknüpft. Durch die Beachtung dieser Tatsache sowie durch Einbeziehung dienstorientierter Aspekte ist beispielsweise die formale Spezifikationstechnik aus [Gars 04] sehr gut geeignet. Für weitere Hinweise zu diesem Themenbereich sei auf Abschnitt 3.4.2 verwiesen.

Verknüpfung von Messung und Semantik

Eigenschaft: Wertemenge

Selbst wenn *QoS-Parameter* in unterschiedlichen Architekturen die gleiche *Semantik* haben, können sie sich noch in ihrer *Wertemenge* unterscheiden.

Eigenschaft Wertemenge

Dies ist die Menge aus der ein *QoS-Parameter* belegt werden kann, wenn er instanziiert wird.

Semantik interpretiert Wertemenge

Die *Wertemenge* kann dabei z.B. als Liste diskreter Belegungen oder als ein Intervall der rationalen Zahlen zusammen mit einer Einheit angegeben werden. Für einen fest definierten Wert ist aber auch nur ein Skalar möglich. Die Interpretation der *Wertemenge* wird dabei immer auch von der *Semantik* bestimmt.

Für das vorherige Beispiel des „Oneway Delay“ könnten in einer Architektur nur die Werte „Niedrig“, „Mittel“ oder „Hoch“ bestimmt werden, in einer anderen Architektur dagegen ein skalarer Wert in Millisekunden.

Eigenschaft: Garantietyp

Garantietypen unterscheiden „Härte“

Wie bereits die Beispiele ATM und DiffServ gezeigt haben, unterscheiden sich Netz-QoS-Architekturen auch in der „Härte“ oder Güte, in der sie *QoS-Parameter* durchsetzen. Modelliert wird diese Eigenschaft mit der Klasse *Garantietyp*. Da eine Architektur nicht zwangsweise alle ihre Parameter in der gleichen Güte durchsetzt, ist diese Klasse eine Eigenschaft eines Parameters (modelliert durch eine Assoziation) und ist nicht als globales Attribut formuliert.

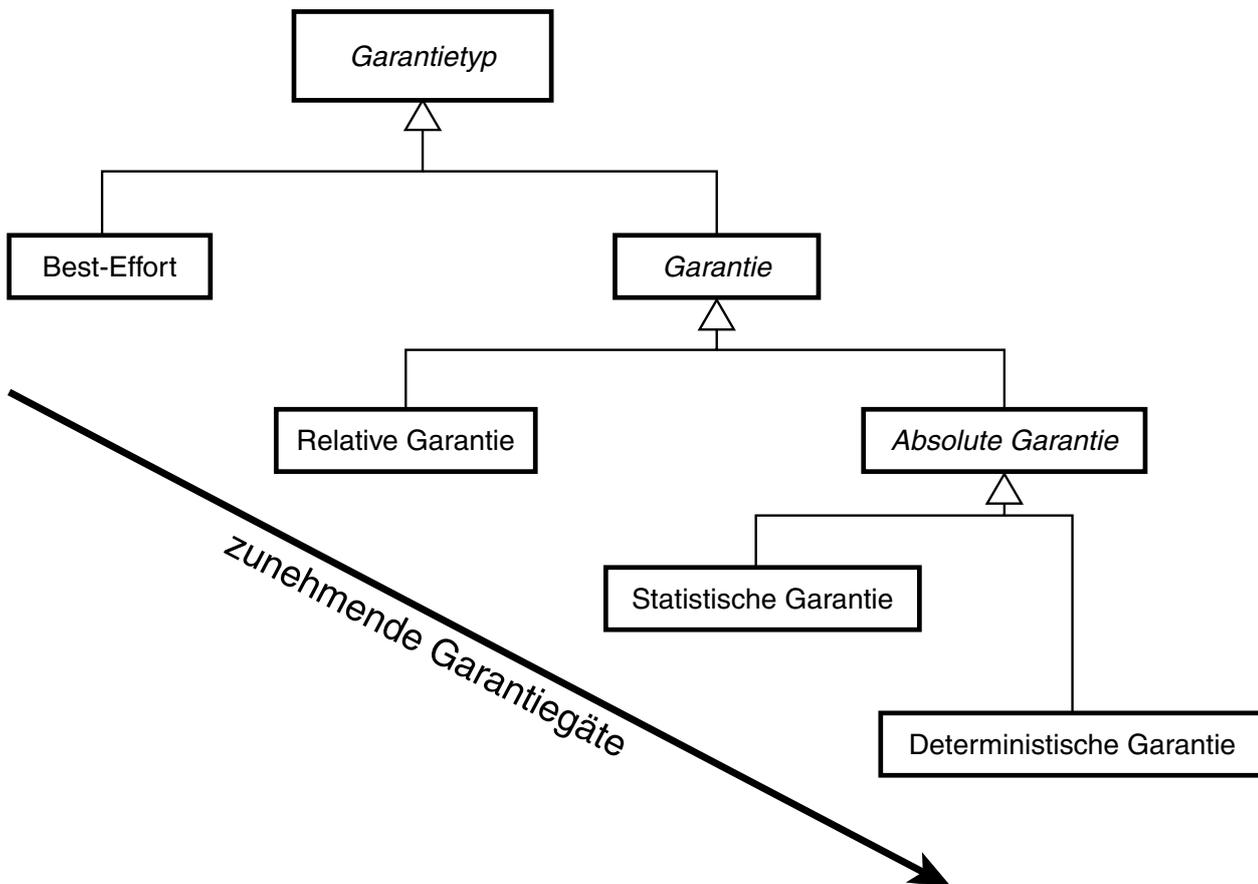


Abbildung 4.12: Statisches Modell des *Garantietyps* eines *QoS-Parameters* (nach [SchmJB 01])

Schmitt gibt in [SchmJB 01] eine Hierarchie von Garantietypen an (Abb. 4.12), die sich nahtlos in die hiesige Modellierung einfügt: Hierarchie der Garantietypen

- **Best-Effort:** Der Garantietyp ohne jegliche Zusicherung.
- **Relative Garantie:** Garantien werden nur in Relation zueinander unterschieden („besser als“), eine absolute (global) Einordnung findet nicht statt. Class-of-Service-Architekturen wie DiffServ geben solche Garantien.
- **Absolute Garantie:** Hier werden Garantien an einer absoluten (globalen) Skala gemessen und gegeneinander abgegrenzt. Dabei lassen sich noch zwei Varianten unterscheiden:
 - ▼ **Statistische Garantie:** Die Garantien werden nicht zu jeder Zeit im absoluten Rahmen gehalten, aber in einem definierten statistischen Mittel werden sie garantiert.
 - ▼ **Deterministische Garantie:** Die geforderten Werte werden zu jeder Zeit eingehalten.

Analog zur Modellierungstechnik der Klasse *Kommunikationsrelation* wird die Hierarchie als Verfeinerung dargestellt, die im Instanzmodell nicht verwendbaren Zwischentypen werden wieder als abstrakte Klassen dargestellt.

QoS-Parameter im Modell von Schmitt

Das Modell von Schmitt beinhaltet sog. *QoS Declarations*, die eine Analogie zu den *QoS-Parametern* besitzen. *QoS Declarations* bestehen bei Schmitt aus folgenden Teilen:

- **Service Classes:** Dies sind die oben bereits erläuterten Garantietypen und damit bereits im hier vorgestellten Modell integriert.
- **Performance Parameter:** Ist ein Parameter einer Netz-QoS-Architektur.
- **Specification Unit:** Einheit, in denen ein *Performance Parameter* spezifiziert wird. Einheiten werden modelliert, um Vergleichbarkeit vom Parametern zu erreichen.

Schmitts Modellierung muss, abgesehen von seinen *Service Classes*, leider als ungenau bezeichnet werden. Das Herausstellen von Einheiten zur Spezifikation eines Parameters ohne dabei explizit die Semantik des Parameters zu betrachten ist unzureichend. Wie das obige Beispiel des Parameters „Delay“ bereits gezeigt hat, muss die gesamte Semantik eines Parameters für einen Vergleich mit einem anderen herangezogen werden. Schmitts *Specification Units* sind in der vorliegenden Modellierung Teil der Klasse *Semantik*. Für diesen Themenkomplex sei hier nochmals auf [Gars 04] verwiesen. Ungenauigkeiten im Modell von Schmitt

Unterarten von QoS-Parametern

In [SchmH 01] definiert H. Schmidt drei Arten von QoS-Parametern:

- **Volumenparameter:** Parameter, die das transportierte Datenvolumen bestimmen.

■ **Geschwindigkeitsparameter:** Parameter, die spezifizieren, mit welcher Geschwindigkeit Daten transportiert werden.

■ **Korrektheitsparameter:** Parameter, die Korrektheit, „Unversehrtheit“ und Vollständigkeit der übertragenen Daten spezifizieren.

Spezialisierungen von allg. QoS-Parametern

Diese Arten stellen Unterarten eines allgemeinen *QoS-Parameters* dar, deshalb sind sie als Spezialisierungen des allgemeinen *QoS-Parameters* modelliert. Die Unterscheidung wird hier in das Modell übernommen, da die Unterarten in der Koppelung ein grundlegend anderes Verhalten zeigen, wie in Kapitel 5 gezeigt werden wird.

Standard-Parameter

Einteilung häufig zu findender QoS-Parameter

In vielen existierenden Architekturen finden sich immer wieder Parameter, die zumindest ähnlich und typisch für Netz-QoS-Architekturen sind. Diese werden hier als Standard-Parameter bezeichnet.

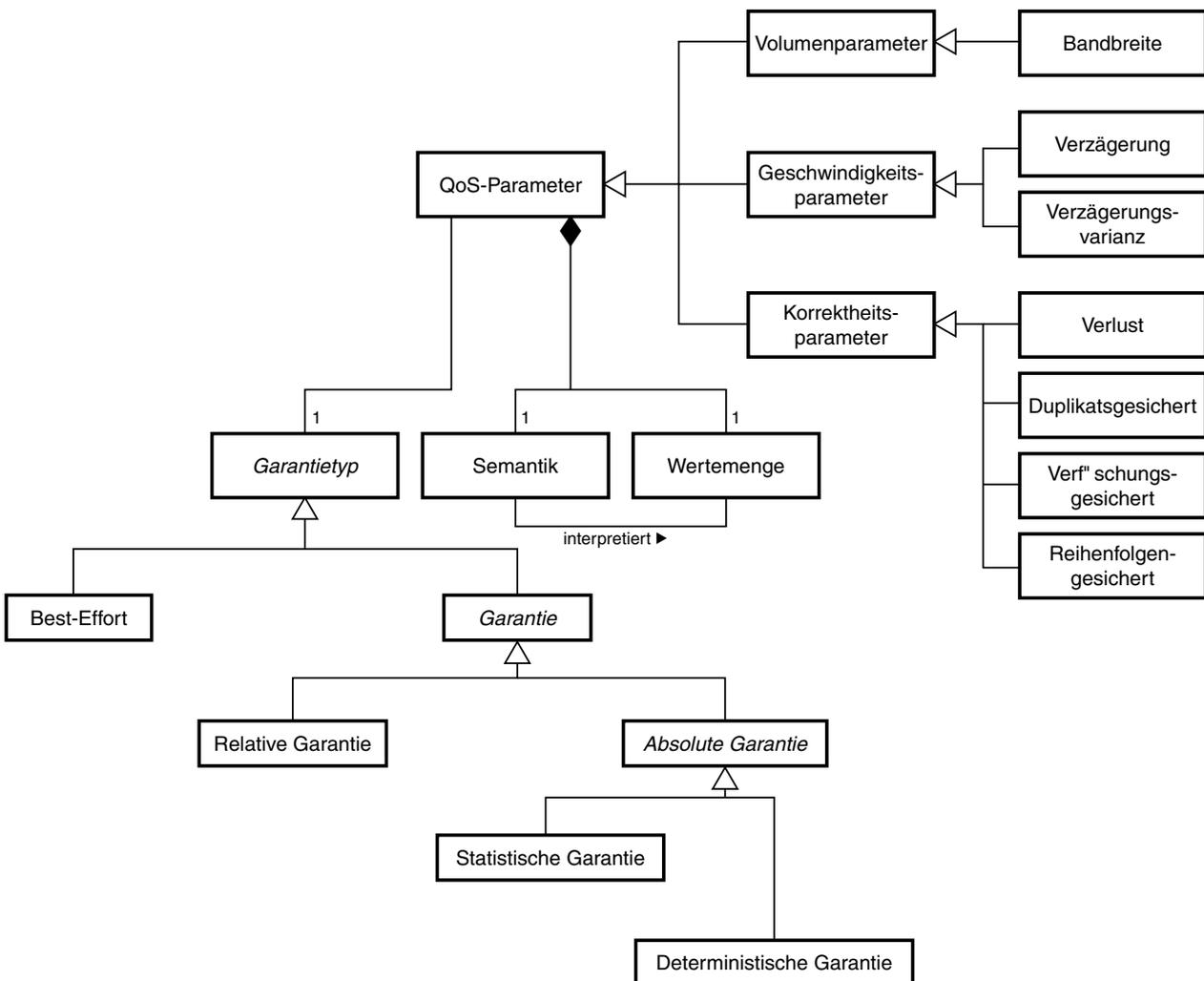


Abbildung 4.13: Vollständiges Modell für *QoS-Parameter* mit Standard-Parameter

Die folgende Liste ordnet sie den jeweiligen Unterarten zu, Abbildung 4.13 zeigt sie zusammen mit dem vollständigen Modell der Klasse *QoS-Parameter*:

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

- *Volumenparameter*
 - ▼ *Bandbreite (bandwidth)*: Für Übertragung verfügbare Kapazität.
- *Geschwindigkeitsparameter*
 - ▼ *Verzögerung (delay)*:
 - ▼ *Verzögerungsvarianz (jitter)*: Variabilität der Verzögerung.
- *Korrektheitsparameter*: In dieser Unterart von *QoS-Parameter* gehen nun auch die inhaltsbezogenen Eigenschaften aus der klassischen Literatur auf, die für die Klasse *Daten-Kategorie* nicht berücksichtigt wurden.
 - ▼ *Verlust (loss)*: Beschreibt die Wahrscheinlichkeit dass Daten verworfen werden. Der Spezialfall mit dem Wert 0 modelliert das Attribut „verlustgesichert“.
 - ▼ *Verfälschungsgesichert*: Daten werden inhaltlich nicht verfälscht (z.B. durch Bitfehler).
 - ▼ *Reihenfolngesichert*: Daten erreichen den Empfänger in der gleichen Reihenfolge wie sie abgesendet wurden.
 - ▼ *Duplikatgesichert*: Daten erreichen den Empfänger genau einmal.

Für konkrete Fälle müssen die genannten Standard-Parameter natürlich noch entsprechend der oben eingeführten Modellierung weiter spezifiziert werden. Ebenso müssen Architekturen keinen, alle oder auch nur einen dieser Parameter spezifizieren.

Modellierung von QoS-Parametern im Beispiel

Nachfolgend einige Beispiele zur konkreten Modellierung von *QoS-Parametern*:

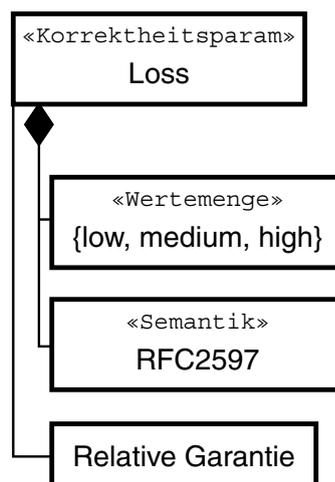


Abbildung 4.14: «QoS-Parameter» von DiffServ mit Assured Forwarding PHB

In [Abbildung 4.14](#) ist der einzige für DiffServ Assured Forwarding PHB festgelegte Parameter dargestellt, wie er in [[RFC 2597](#)] spezi-

QoS-Parameter von DiffServ Assured Forwarding PHB

fiziert wird. Es handelt sich um den «Korrektheitsparameter» *Loss* (Verlust). Seine Semantik ist in [RFC 2597] beschrieben, was im Diagramm in der entsprechenden «Semantik»-Klasse dargestellt ist. Eine mögliche Belegung des Parameters kann laut dem Standard aus der «Wertemenge» „low“ (niedrig) „medium“ (mittel) „high“ (hoch) gewählt werden. Dabei werden im Standard keinerlei konkrete Werte für die drei Belegungen angegeben, weder statistische noch absolute. Da im Standard weder konkretere Belegungen vorgegeben sind noch dies überhaupt vorgesehen ist, handelt es sich bei diesem «QoS-Parameter» um einen, der nur eine relative Garantie bietet. Deshalb ist *Loss* mit der Klasse *Relative Garantie* aus Abbildung 4.12, Seite 92 assoziiert.

Standardparameter von IP

Neben den für DiffServ Assured Forwarding PHB spezifischen «QoS-Parameter» sind auch noch die durch IP vorgegebenen Standard-Parameter zu modellieren, da DiffServ auf IP basiert. Für die Standard-Parameter *Bandbreite*, *Verzögerung* und *Verzögerungsvarianz* macht IP keine Zusagen, sie werden deshalb nicht modelliert. Neben dem «Korrektheitsparameter» *Verlust*, der durch Assured Forwarding PHB bereits verfeinert definiert wurde, werden durch IP weitere Standard-Parameter festgelegt (Abb. 4.15):

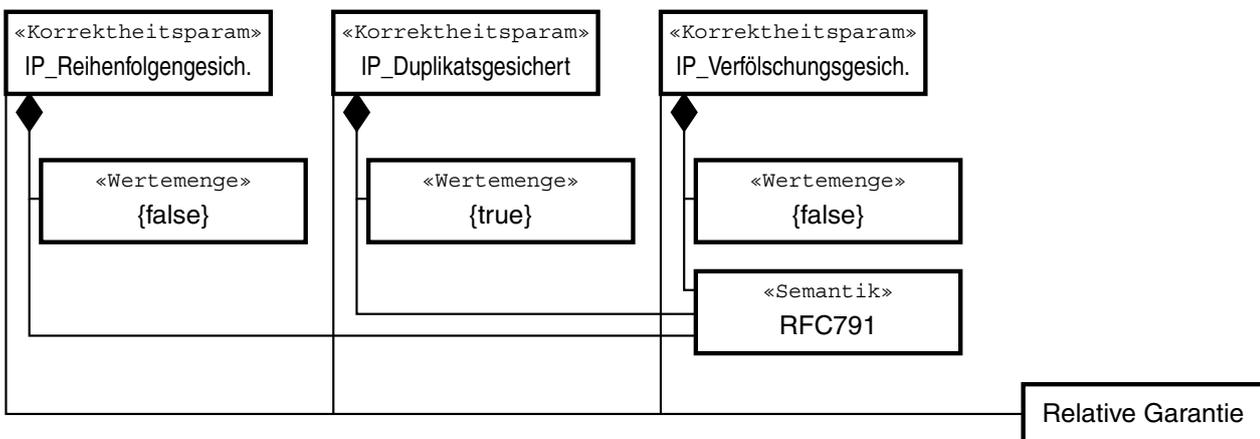


Abbildung 4.15: Restliche Standard-Parameter von IP

- Klasse „IP_Duplikatsgesichert“
- Klasse „IP_Verfälschungsgesichert“
- Klasse „IP_Reihenfolgesichert“

- Klasse *IP_Duplikatsgesichert*: Durch die Art der in IP benutzten Weiterleitungsmechanismen können keine Duplikate auftreten.
- Klasse *IP_Verfälschungsgesichert*: IP bietet nur für die Protokollheader-Informationen eine Überwachung auf Verfälschung, nicht jedoch für die transportierten Daten.
- Klasse *IP_Reihenfolgesichert*: Dadurch dass in IP prinzipiell auf jedes Paket eine eigene Routing-Entscheidung angewendet werden kann, bietet IP keine gesicherte Reihenfolge.

Allen Parametern gemein ist in der Modellierung ihre «Semantik»-Klasse, nämlich das RFC³-Dokument von IP, und ihre Klassifikation als relative Garantie. Die Assoziation zwischen «Semantik»-Klasse und den Klassen für die «Wertemengen» (siehe Abb. 4.11, S. 91) wurde der Übersichtlichkeit halber weggelassen.

gemeinsame
Semantik-Klasse

Die ATM-Architektur gibt ein gutes Beispiel für eine reale Architektur, die eine Reihe von Standard-Parametern realisiert. Die nachfolgende Übersicht ordnet die ATM-QoS-Parameter in die oben definierten Unterarten ein; die dabei eingeführten Abkürzungen sind jene, die in den Standarddokumenten [ATM-TM 99] verwendet werden:

ATM realisiert
Standard-Parameter

- «Volumenparameter»
 - ▼ Peak Cell Rate (PCR): Spitzenbandbreite
 - ▼ Sustained Cell Rate (SCR): Dauerhaft benötigte Bandbreite
 - ▼ Minimum Cell Rate (MCR): Minimale tolerierbare Bandbreite
 - ▼ Maximum Burst Size (MBS): Maximale Größe einer temporären Überanforderung
 - ▼ Maximum Frame Size (MFS): Maximale Größe eines Rahmens
- «Geschwindigkeitsparameter»
 - ▼ maximum Cell Transfer Delay (maxCDT): Maximale Übertragungsverzögerung
 - ▼ peak-to-peak Cell Delay Variation (peakToPeakCDV): Varianz der Verzögerung zwischen zwei Zellen
- «Korrektheitsparameter»
 - ▼ Cell Loss Rate (CLR): Verlustrate von Zellen

4.3.1.4 QoS-Kategorie

Die Klasse *QoS-Kategorie* fasst *QoS-Parameter* zu einem Satz zusammen, im Modell durch eine Komposition dargestellt. Eine bestimmte *QoS-Kategorie* muss dabei nicht alle in der Architektur verfügbaren *QoS-Parameter* umfassen. In einer Architektur finden sich für gewöhnlich mindestens zwei verschiedene *QoS-Kategorien*:

Gruppierung von
QoS-Parametern

- Die **Best-Effort-Kategorie** gewährt keine Qualitätzusicherungen. Mit dieser Kategorie ist genau ein *QoS-Parameter* assoziiert, der vom Grundtyp der Best-Effort-Klasse ist oder davon abgeleitet ist.

³Request For Comments (RFC)

- Eine *QoS-Kategorie*, die mindestens einen *QoS-Parameter* enthält, der nicht von der Best-Effort-Klasse abgeleitet ist.

QoS-Kategorien sind mit *Daten-Kategorien* assoziiert, sie weisen den Daten die gewünschten Qualitätskriterien zu.

Modellierung von QoS-Kategorien im Beispiel

Die Modellierung von *QoS-Kategorien* wird im folgenden an den beiden Standardbeispielen DiffServ und ATM illustriert.

QoS-Kategorien von DiffServ AF PHB

Die DiffServ-Architektur mit Assured Forwarding PHB gibt 4 «QoS-Kategorien» an, die mit *AF_1* bis *AF_4* bezeichnet werden.

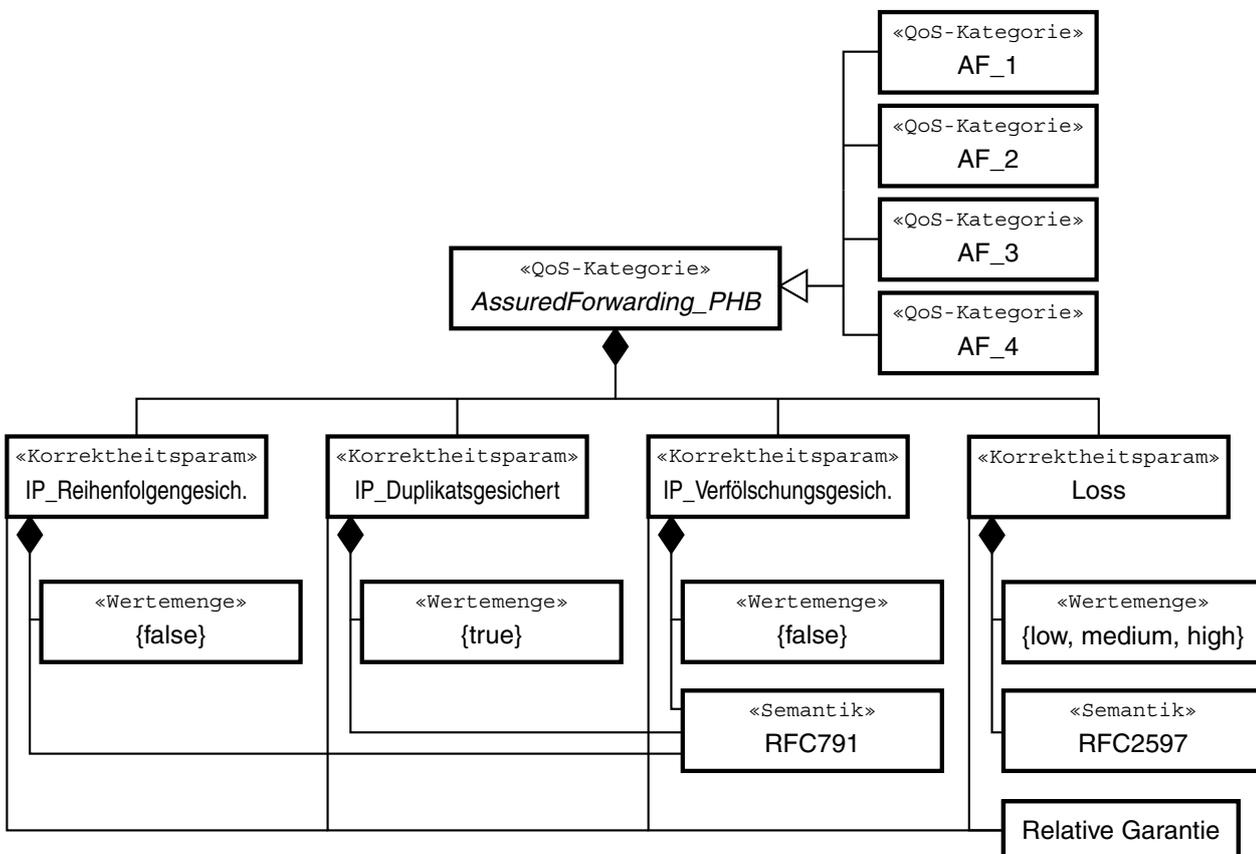


Abbildung 4.16: «QoS-Kategorien» von DiffServ mit Assured Forwarding PHB

Abbildung 4.16 modelliert diese QoS-Kategorien. Um zu kennzeichnen, dass diese Kategorien alle zu Assured Forwarding PHB gehören, sind diese als Verfeinerungen der gemeinsamen abstrakten Oberklasse *AssuredForwarding_PHB* modelliert. Weiterhin gibt der Standard selbst keinerlei Unterscheidungsmerkmale der einzelnen Kategorien an. Aus diesem Grund sind der einzige im Standard spezifizierte «QoS-Parameter» *Loss* (vgl. Abb. 4.14, S. 95) und die durch IP definierten Eigenschaften (vgl. Abb. 4.15, S. 96) an die Oberklasse gekoppelt, da diese Eigenschaften für alle Ka-

tegorien gelten. Sie werden durch die Oberklasse an die einzelnen Kategorien vererbt.

Die ATM-Architektur gibt ein gutes Beispiel für die selektive Auswahl von *QoS-Parametern* für unterschiedliche *QoS-Kategorien*. Abbildung 4.17 modelliert ATMs «QoS-Kategorien» mit ihren zugehörigen «QoS-Parametern», wie sie bereits in Tabelle 2.1, Seite 21 kurz vorgestellt wurden. Durch den Einsatz von abstrakten Kategorieklassen werden in der Modellierung die strukturellen Zusammenhänge zwischen verschiedenen Kategorien verdeutlicht. Mit der abstrakten Kategorie *VBR_Kategorie* ist eine sog. Mix-In-Klasse realisiert, die über den Mechanismus der Mehrfacherbung die zusätzliche Eigenschaft den erben Klasse hinzufügt. Nachfolgend werden der hierarchische Aufbau und die Kategorien genauer beschrieben. In der zugehörigen Abbildung 4.17 sind die zu den Kategorien gehörigen Parameter-Klassen zur besseren Übersicht grau hinterlegt dargestellt; aus gleichem Grund wird auf die detaillierte Darstellung der einzelnen Parameter verzichtet. Für die Parameter werden die Abkürzungen aus dem Abschnitt „Modellierung von QoS-Parametern im Beispiel“ verwendet.

Selektive Zuordnung von Parametern zu Kategorien bei ATM

Mix-In-Klasse

■ Abstrakte Kategorie *realTime_Kategorie*

Ausschließlich für Kategorien, die von dieser abstrakten Kategorie abgeleitet sind, können die Geschwindigkeitsparameter der ATM-Architektur (*maxCDT* und *peakToPeakCDV*) vereinbart werden.

Echtzeit Kategorien

▼ Kategorie Constant Bit Rate (CBR): *CBR_Kat*

Die CBR-Kategorie ist für Verbindungen gedacht, die sowohl konstante Bandbreite als auch enge Grenzen für die Verzögerung und deren Varianz über die gesamte Verbindungsdauer benötigen, z.B. Sprachübermittlung, Video, Leitungsemulation etc. Die Bandbreite wird über den «QoS-Parameter» *PCR* vereinbart, Verzögerung und deren Varianz über *maxCDT* und *peakToPeakCDV*.

▼ Kategorie real-time Variable Bit Rate (rtVBR): *rtVBR_Kat*

Für Echtzeitanwendungen, die im Gegensatz zu solchen der CBR-Kategorie eine variable Bandbreite benötigen, d.h. „bursty“ sind, steht die Kategorie *rtVBR* zur Verfügung. Hier können zusätzlich die durch die Klasse *VBR_Kategorie* geerbten Parameter *SCR* und *MBS* vereinbart werden.

■ Abstrakte Kategorie *nonRealTime_Kategorie*

Unterkategorien, die von dieser abstrakten Kategorie abgeleitet sind, zeichnen sich dadurch aus, dass keine Geschwindigkeitsparameter spezifiziert werden können. Mit Ausnahme der

Nicht-Echtzeit Kategorien

Kategorie *nrtVBR_Kat* kann auch keine Verlustrate über den Parameter *CLR* vereinbart werden.

▼ Kategorie non-real-time Variable Bit Rate (nrtVBR): *nrtVBR_Kat*

Die nrtVBR-Kategorie entspricht der rtVBR-Kategorie mit dem Unterschied, dass sie nicht für Echtzeitanwendungen gedacht ist. Entsprechend können die Geschwindigkeitsparameter *maxCDT* und *peakToPeakCDV* hier nicht angegeben werden.

▼ Kategorie Unspecified Bit Rate (UBR): *UBR_Kat*

UBR stellt die „schwächste“ aller ATM-Kategorien dar. Sie ist für klassische Anwendungen wie Dateitransfer oder Email gedacht, die auch in Best-Effort-Netzen korrekt funktionieren. Außer der *PCR* können keine weiteren Parameter vereinbart werden.

▼ Kategorie Available Bit Rate (ABR): *ABR_Kat*

ABR entspricht im wesentlichen UBR, nur kann hier über den Parameter *MCR* eine Mindestbandbreite spezifiziert werden.

▼ Kategorie Guaranteed Frame Rate (GFR): *GFR_Kat*

Die GFR-Kategorie stellt eine Ausnahme im Vergleich zu allen anderen Kategorien dar. Müssen zur Durchsetzung der QoS-Eigenschaften des Gesamtsystems Daten verworfen werden, dann werden in den anderen Kategorien ATM-Zellen verworfen ohne Zugehörigkeiten zu den PDUs höherer Schichten zu beachten. Bei GFR dagegen wird diese Zugehörigkeit beachtet und, wenn notwendig, werden Zellen verworfen, die zu einer gemeinsamen höherschichtigen PDU gehören.

■ Abstrakte Kategorie *VBR_Kategorie*

Mix-In-Kategorie

Die für die beide VBR-Kategorien *rtVBR_Kat* und *nrtVBR_Kat* charakteristischen «QoS-Parameter» *SCR* und *MBS* werden durch diese Kategorie zusammengefasst.

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

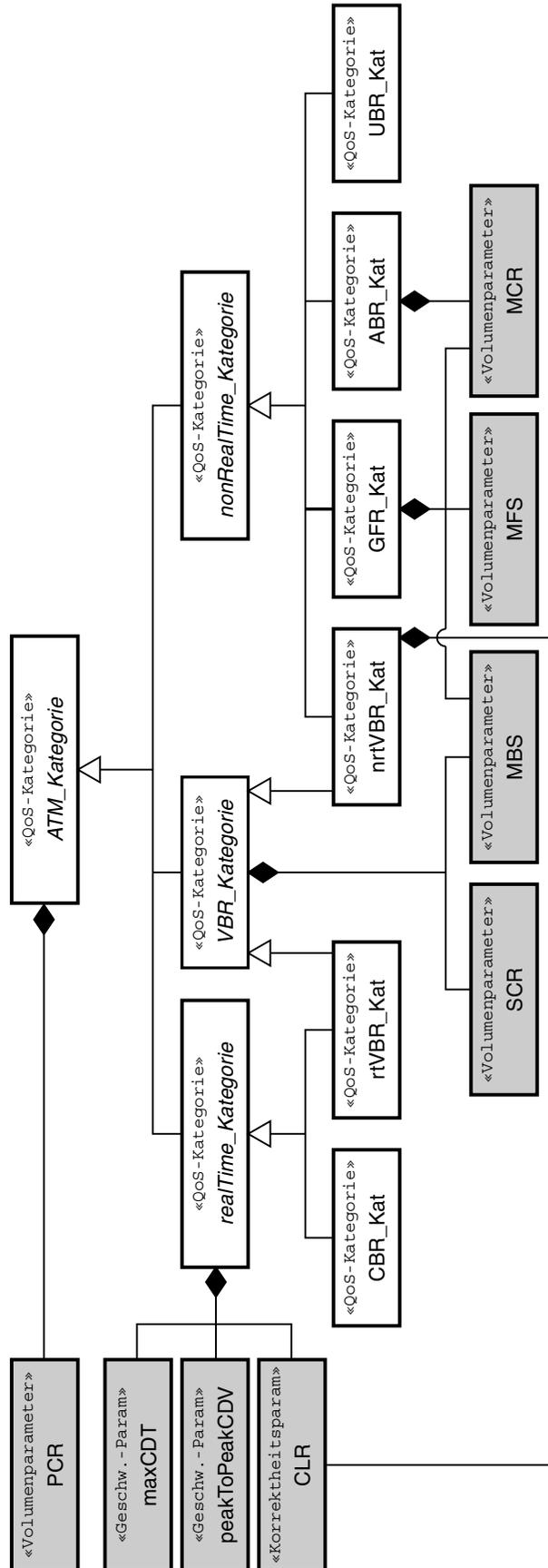


Abbildung 4.17: «QoS-Kategorien» von ATM

4.3.1.5 Session

Assoziation zwischen Daten- und QoS-Kategorien Um den Datentransport mit QoS-Eigenschaften zu versehen, werden *Daten-Kategorien* mit *QoS-Kategorien* im Modell assoziiert. Damit werden einer *Daten-Kategorie* auch die für sie bestimmenden *QoS-Parameter* zugewiesen.

In Anwendung der Black-Box-Sicht auf eine Netz-QoS-Architektur ist für diese Assoziation nur ein Attribut entscheidend. Grundlegend unterscheiden sich die Architekturen nämlich darin, ob die Zuordnung von Qualitätseigenschaften zum Datentransport ...

Explizite Zordnung ■ ... explizit ist. Dies bedeutet, dass unmittelbar im Datenprotokoll die Zuordnung zu den Qualitätseigenschaften vorgenommen wird. Die Qualitätseinstellungen lassen sich also durch Lesen des Datenprotokolls direkt ermitteln.

In der DiffServ-Architektur wurde im IP-Header das Type of Service (ToS) Byte in DiffServ (DS) Feld umbenannt [RFC 2474]. Hierin wird in den oberen 6 Bits ein sog. DiffServ Codepoint (DSCP) [RFC 3140] eingetragen. Dies ist ein Identifikator, der unmittelbar eine *QoS-Kategorie* von DiffServ identifiziert.

Implizite Zordnung ■ ... implizit ist. In diesem Fall ist im Datenprotokoll alleine nicht ermittelbar, welche Qualitätseinstellungen für die Daten getroffen worden sind. Vielmehr ist diese Zuordnung nur durch entsprechende Management-Protokolle nachvollziehbar.

Bei ATM sind den datentransportierenden Zellen nur ein Virtual Path Identifier (VPI) / Virtual Channel Identifier (VCI) zugeordnet. Die Qualitätseigenschaften der ATM-Verbindung sind nur implizit im ATM-Management mit den Zellen verknüpft.

Die Attributierung der Qualitätszuordnung geschieht im Modell durch die Assoziationsklasse *Session*. Sie besitzt das boolesche Attribut *Zuordnung_explicit*. Seine Belegung repräsentiert die beiden Fälle „implizit“ (=false) und „explizit“ (=true).

Eigenschaft aus Schmitts Arbeit Bezüglich der Arbeit von Schmitt wurde bei der Erläuterung der Klasse *Daten-Kategorie* festgestellt, dass die Aufgabe „Packet Classification / Stamping“ zu modellieren ist. Dies wird genau durch die Assoziationsklasse *Session* realisiert.

4.3.1.6 Vollständiges statisches Grundmodell

In Abbildung 4.18 ist das statische Grundmodell für Netz-QoS-Architekturen nochmals vollständig ohne Standard-Parameter dargestellt.

Vollst. Grundmodell DiffServ AF PHB In Abbildung 4.19 ist die vorher Schritt für Schritt entwickelte statische Modellierung der DiffServ-Architektur mit Assured Forwarding PHB zusammengefasst dargestellt. In dieser Darstellung ist

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

nun auch die Instanziierung des Grundmodells für die Klasse *Session* in Form der Klasse *Global_DS_Session* enthalten. Wie im vorhergehenden Abschnitt bereits beschrieben, handelt es sich bei DiffServ um eine Architektur, bei der die Zuordnung zu den QoS-Eigenschaften explizit vorgenommen wird; deshalb ist das Attribut *Zuordnung_explicit* auf konstant *true* gesetzt.

Am kompletten statischen Modell von AF PHB lassen sich bereits einige Eigenschaften dieser Spezifikation ablesen:

- Die «QoS-Kategorien» *AF_1* bis *AF_4* können hinsichtlich ihrer QoS-Eigenschaften nicht voneinander unterschieden werden, denn alle vom Standard definierten *QoS-Parameter* gelten für die gemeinsame Oberklasse, aber für die verfeinerten «QoS-Kategorien» sind keine weiteren Parameter definiert. Wenn mit Assured Forwarding PHB ein Netz-QoS-Dienst aufgebaut werden soll und dabei die Unterscheidung der QoS-Kategorien sinnvoll eingesetzt werden soll, dann muss der Dienst die einzelnen Kategorien um differenzierende «QoS-Parameter» erweitern.

QoS-Kategorien
nicht unterscheidbar

Der Standard alleine genügt also nicht, um für einen Dienst die vom Standard bereits vorgesehenen Eigenschaften vollständig zu nutzen. Ein solcher Standard provoziert somit die bei realen Netz-QoS-Diensten zu beobachtende Diversifizierung selbst. Diese wiederum unterstreicht die Notwendigkeit der vorliegenden Arbeit.

Standard provoziert
Diversifizierung

- Durch die gemeinsame abstrakte Oberklasse *AssuredForwarding_PHB* wird eine Modularisierung und Kapselung der für AF PHB benötigten Teile erreicht. In einem realen Dienst könnten zusätzlich noch andere PHBs in den Dienst integriert werden.

Abstrakte
Oberklasse zur
Modularisierung und
Kapselung

Auch die Modellierung anderer DiffServ-Varianten (wie z.B. Expedited Forwarding PHB [RFC 3246]) wird hierdurch erleichtert. Zu deren Modellierung muss einfach nur der Teil ab der abstrakten Oberklasse *AssuredForwarding_PHB* durch einen für die neue Variante spezifischen Teil ersetzt werden.

- Das Beispiel zeigt, wie bei einer realen Netz-QoS-Architektur Anomalien auftreten würden, wenn inhaltsbezogene Eigenschaften des Datentransports in bzw. mit der Klasse *Daten-Kategorie* modelliert würden. Für das abgebildete Beispiel würde dann die Eigenschaft „Verlustgesichert“ und der dazu duale *QoS-Parameter Loss* gleichzeitig im Modell auftreten.

Vermeidung von
Modellierungsanomalien

Jetzt sind die statischen Charakteristika aller Netz-QoS-Architekturen erklärt, die aus Dienst-Sicht relevant sind. Nun stellt sich die Frage, was Netz-QoS-Architekturen dann so unterschiedlich macht. Darauf geben die folgenden Abschnitte eine Antwort.

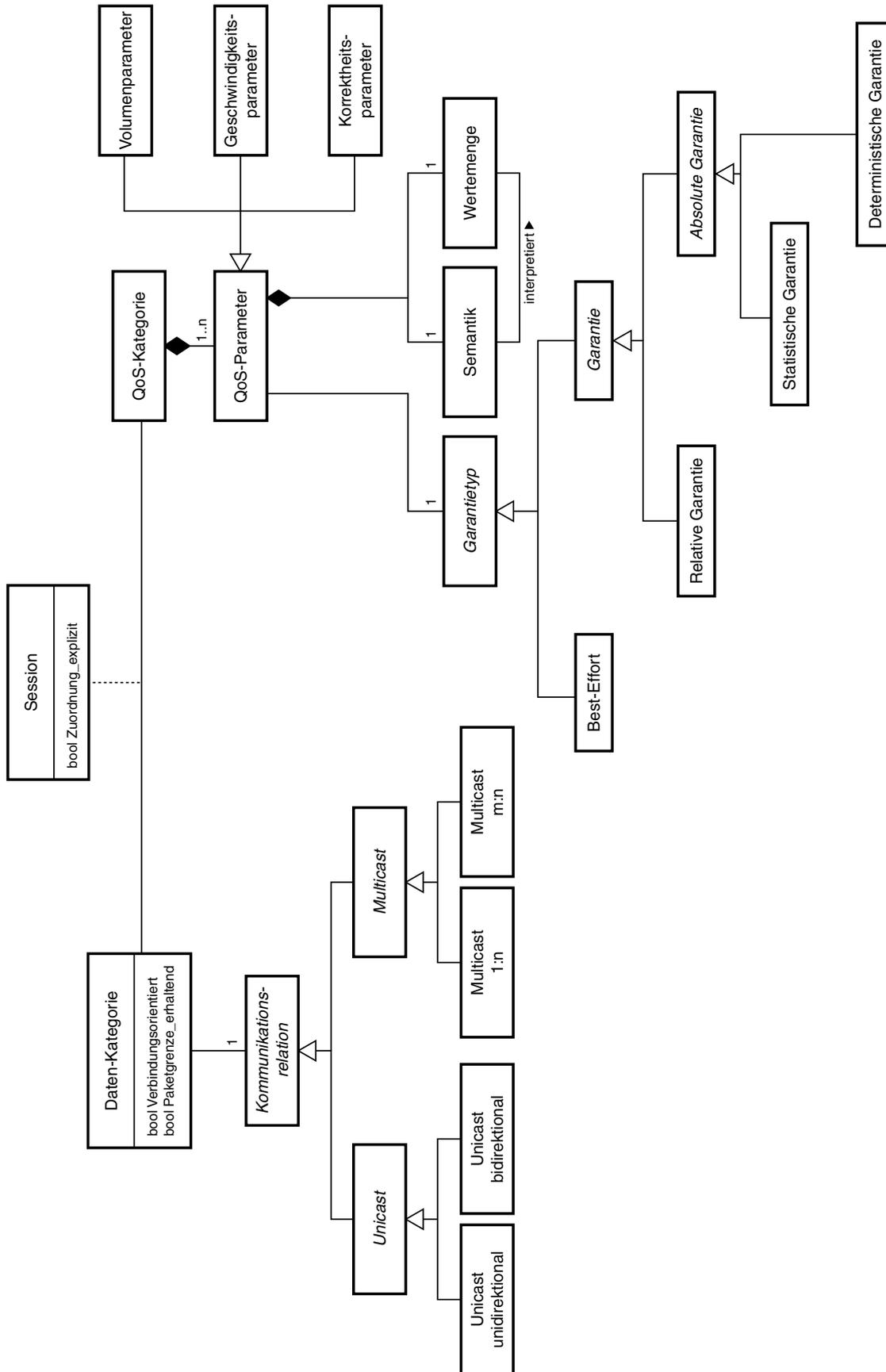


Abbildung 4.18: Vollständiges statisches Grundmodell ohne Standard-Parameter

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

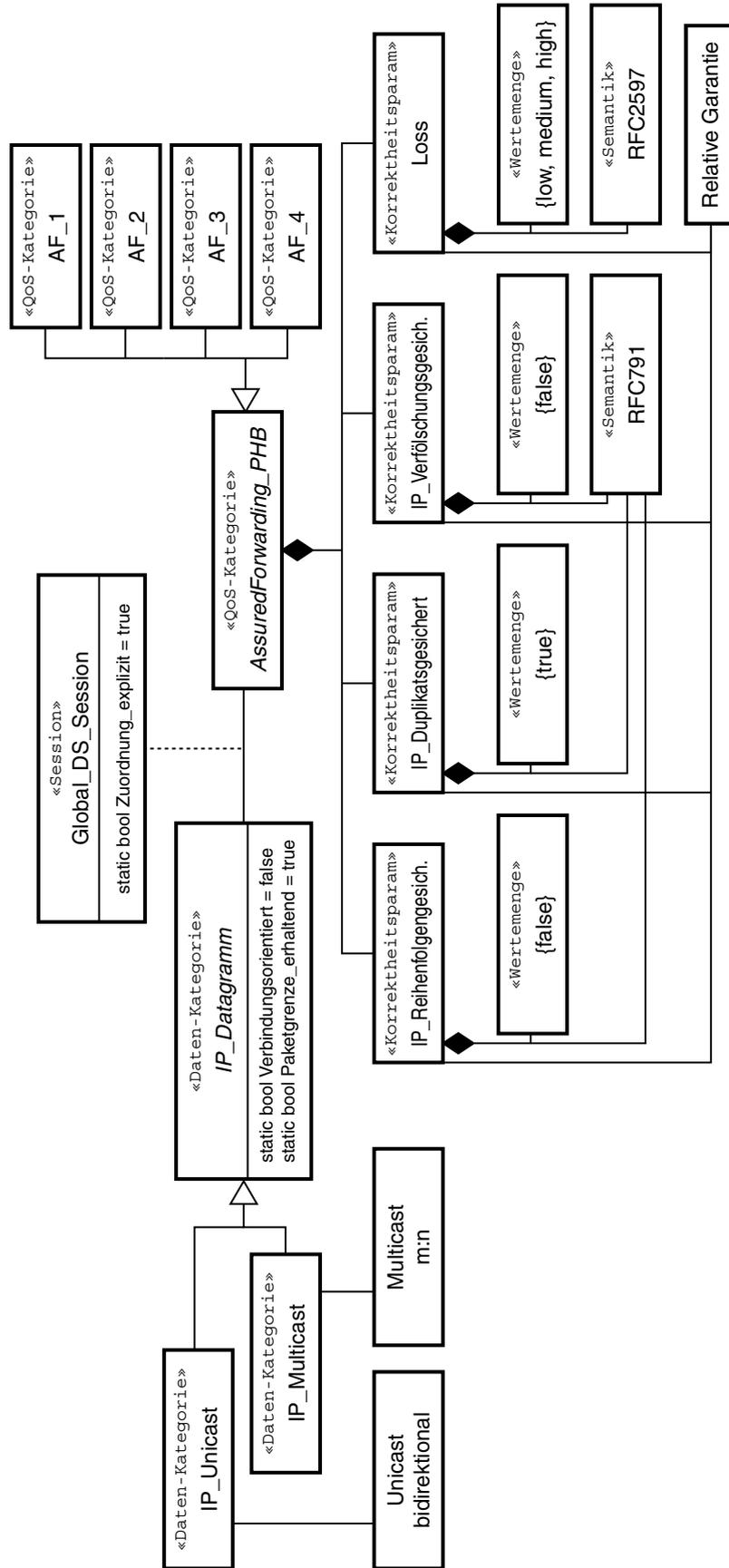


Abbildung 4.19: Statisches Grundmodell von DiffServ mit Assured Forwarding PHB

4.3.2 Lebenszyklus bei Netz-QoS-Architekturen

Zeitachse für dyn. Modell Um die in späteren Abschnitten erläuterten Teile einer Netz-QoS-Architektur auf der Zeitachse einordnen zu können, wird in diesem Abschnitt ein allgemeiner Lebenszyklus für Netz-QoS-Architekturen eingeführt.

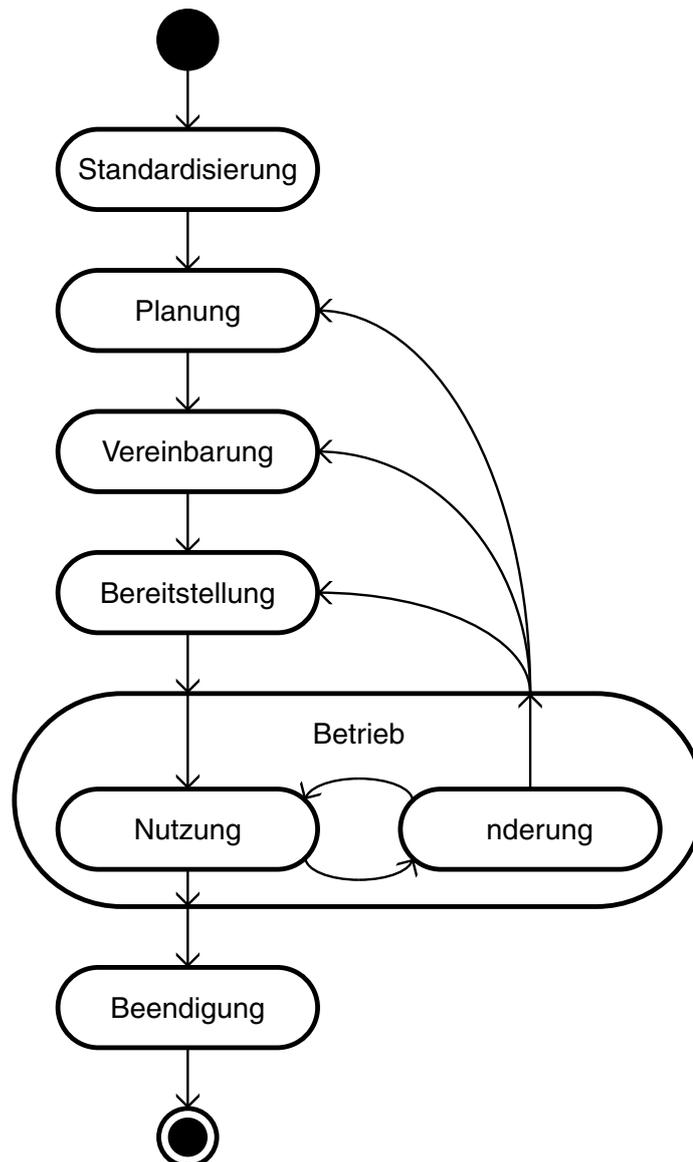


Abbildung 4.20: Lebenszyklus von Netz-QoS-Architekturen

Der in [Abbildung 4.20](#) als UML-Aktivitätsdiagramm dargestellte Lebenszyklus ist eine Erweiterung des in [\[Dreo 02\]](#) vorgestellten Dienst-Lebenszyklus. In den folgenden Erläuterungen zu den einzelnen Phasen werden auch erste Bezüge zur Dienst-Modellierung hergestellt.

4.3.2.1 Lebenszyklusphase Standardisierung

Der Lebenszyklus einer Netz-QoS-Architektur beginnt mit ihrer **Standardisierung**. Die Phase der Ausarbeitung eines Standards wird, im Unterschied zu anderen Darstellungen, im hiesigen Lebenszyklus explizit modelliert. Mit dem Abschluss der Phase kann dieser Abschnitt im Lebenszyklus nicht mehr betreten werden.

In den Ebenen der Modellierung aus Abbildung 4.8, Seite 84 ausgedrückt, wird in dieser Phase ein konkretes Modell einer Netz-QoS-Architektur geschaffen, die sich auf Basis des Meta-Modells darstellen lässt.

4.3.2.2 Lebenszyklusphase Planung

Soll eine Netz-QoS-Architektur eingesetzt werden, beginnt die **Planungsphase**. Im Fall einer Dienstrealisierung kann der Anstoß dabei sowohl von Seiten des Providers als auch von Seiten des (zukünftigen) Kunden kommen. In dieser Phase liegt der Fokus auf generellen Betrachtungen zur Realisierbarkeit, zu Änderungen in Infrastruktur, Workflows und der Unternehmensausrichtung.

Bezüglich der Modellierungsebenen aus Abbildung 4.8, Seite 84 beginnt hier der Weg vom konkreten Modell hin zur Kundeninstanz.

4.3.2.3 Lebenszyklusphase Vereinbarung

Wenn aus den allgemeineren Überlegungen der Planungsphase kein negativer Bescheid resultiert, dann können konkrete Verhandlungen über den Dienst in der **Vereinbarungsphase** erfolgen. An der Dienstsicht des MNM-Dienstmodells (Abb. 4.4, S. 73) erklärt: Zwischen *Provider* und zukünftigen *Kunden* finden Verhandlungen über den zu erbringenden *service* statt. Dabei werden sowohl die Nutzfunktionalität *usage functionality* als auch die Managementfunktionalität (*management functionality*) des Dienstes vereinbart. Eine umfassende Behandlung der hier zu erfüllenden Aufgaben und deren Abarbeitung liefert [SchmH 01].

Für einen ATM-basierten Dienst werden hier beispielsweise die für den Kunden zur Verfügung stehenden «QoS-Kategorien» aus den durch den Standard definierten Kategorien festgelegt. Weiterhin werden dann die für die jeweiligen Kategorien tatsächlich zur Verfügung stehenden Wertebereiche der einzelnen Parameter verhandelt.

Das Ergebnis der Vereinbarungphase ist eine Dienstvereinbarung (*service agreement*), eine Niederschrift der Qualitäts- und Betriebsrichtlinien zwischen Kunde und Provider. Diese enthält ein sog. **Service Level Agreement (SLA)**, in dem u.a. die vereinbarten QoS-Parameter (*QoS parameters*) und deren Werte bzw. Grenzen festgehalten sind.

4.3.2.4 Lebenszyklusphase Bereitstellung

Bereitstellung Mit der Festlegung der Funktionalitäten in der vorherigen Phase ist der Ausgangspunkt für deren tatsächlicher Realisierung gelegt. In der **Bereitstellungsphase** implementiert der Provider beide Teile der Dienstfunktionalität auf seiner Infrastruktur (*service implementation* und *service management implementation*). Auf Seite des Kunden werden durch diesen die für den Zugriff auf die Funktionalitäten benötigten Clients (*service client* und *CSM client*) bereitgestellt.

4.3.2.5 Lebenszyklusphase Nutzung

Nutzung ist Teil der Betriebsphase Wenn eine Netz-QoS-Architektur benutzt werden kann, dann ist die Phase der **Nutzung** erreicht. Damit ist auch die erste Aktivität der **Betriebsphase** beschritten. Bei einem Dienst stehen dem Kunden die vereinbarten Funktionalitäten an den beiden Dienstzugriffspunkten für Nutz- und Managementfunktionalität zur Verfügung.

Für das ATM-Beispiel bedeutet das, dass jetzt ATM-Verbindungen aufgebaut werden können, deren «QoS-Parameter» sich im Rahmen der im SLA festgelegten Grenzen bewegen dürfen.

Kundeninstanz ausgeprägt Im Rahmen der Modellierung ist in dieser Phase die Ebene der Kundeninstanzen voll ausgeprägt (Abb. 4.8, S. 84).

4.3.2.6 Lebenszyklusphase Änderung

Die zweite Teilaktivität der Betriebsphase ist die **Änderungsphase**.

Folgephasen abhängig von Art der Änderung Die nach der Änderungsphase folgende Aktivität hängt individuell von der Art der Änderung und den daraus resultierenden Folgen ab. Bei einfachen Änderungen kann mit der Nutzungsphase fortgefahren werden. Die Folgen einer Änderungsanforderung können aber so umfangreich sein, dass bis in die Planungsphase zurückgegangen werden muss. Im Dienstkontext gesehen kann eine Änderung sowohl durch die Kundenseite als auch durch die Providerseite angeregt werden.

Auch im ATM-Beispiel können in Abhängigkeit von der Art der Änderung unterschiedliche Folgephasen notwendig sein:

- Für eine bestehende ATM-Verbindung sollen neue Parameter in den im SLA vereinbarten Grenzen festgelegt werden
 - ⇒ Neue ATM-Verbindung ersetzt die alte
 - ⇒ Zurück in die Nutzungsphase.
- Erneuert ein Provider seine Infrastruktur, dann muss der ATM-Dienst erneut auf der neuen Infrastruktur realisiert werden
 - ⇒ Zurück in die Bereitstellungsphase.

- Die im SLA festgelegten Grenzen für die QoS-Parameter sind nicht mehr ausreichend, der Provider könnte aber die neuen Grenzen mit seiner Infrastruktur realisieren
⇒ Zurück in die Vereinbarungsphase.
- Es werden ATM-Eigenschaften benötigt, die der Provider nicht mit seiner vorhandenen Infrastruktur realisieren kann
⇒ Zurück in die Planungsphase.
- Der Dienst wird nicht mehr benötigt
⇒ Weiter in die Beendigungsphase.

4.3.2.7 Lebenszyklusphase Beendigung

Wenn eine Netz-QoS-Architektur nicht mehr benötigt wird, ist die **Beendigungsphase** erreicht. Die in der Bereitstellungsphase für die Realisierung getätigten Aktionen werden hier „zurückgebaut“. Bei einem Dienst verliert auch das zwischen Kunden und Provider geschlossene *service agreement* seine Gültigkeit. Beendigung

4.3.2.8 Zeitpunkte im Lebenszyklus

Im späteren Verlauf der Arbeit werden die Zeitpunkte für Start (t_{S_X}) und Ende (t_{E_X}) der einzelnen Lebenszyklusphasen X benötigt werden. Tabelle 4.1 führt ihre Bezeichnungen ein und gibt gleichzeitig Bedingungen für deren Werte an.

Phase	Zeitpunkt, immer gilt: $t_{S_X} < t_{E_X}$	
	Start	Ende
Standardisierung	$t_{S_{Standard}}$	$t_{E_{Standard}}$
Planung	$t_{S_{Planung}} > t_{E_{Standard}}$	$t_{E_{Planung}}$
Vereinbarung	$t_{S_{Vereinbarung}} > t_{E_{Planung}}$	$t_{E_{Vereinbarung}}$
Bereitstellung	$t_{S_{Bereitstellung}} > t_{E_{Vereinbarung}}$	$t_{E_{Bereitstellung}}$
Betrieb	$t_{S_{Betrieb}} > t_{E_{Bereitstellung}}$	$t_{E_{Betrieb}}$
	Nutzung $t_{S_{Nutzung}} \geq t_{S_{Betrieb}}$	$t_{E_{Nutzung}}$
	Änderung $t_{S_{Aenderung}} > t_{E_{Nutzung}}$	$t_{E_{Aenderung}} \leq t_{E_{Betrieb}}$
Beendigung	$t_{S_{Beendigung}} > t_{E_{Betrieb}}$	$t_{E_{Beendigung}}$

Tabelle 4.1: Zeitpunkte im Lebenszyklus

Nachdem jetzt eine universelle Zeitachse definiert ist, werden in den folgenden Abschnitten die dynamischen Teile der Modellierung erläutert.

4.3.3 Kardinalitätsanalyse im statischen Grundmodell

Kardinalitäten der Assoziationen
 Mit der bisherigen Ausprägung des statischen Grundmodells können Netz-QoS-Architekturen nun strukturell äquivalent dargestellt werden. Wie die Beispiele für ATM und DiffServ gezeigt haben, unterscheiden diese sich natürlich in ihren jeweiligen Verfeinerungen, in der bisherigen Variante des Grundmodells sind aber die Basisbausteine jeweils gleich. Im Grundmodell ist jedoch der Punkt der Kardinalitäten der Assoziationen noch nicht vollständig betrachtet worden, und darin wird sich nun ein weiterer Unterschied zwischen Architekturen manifestieren.

Die Kardinalitäten der Assoziationen zwischen

- *Daten-Kategorie* und *Kommunikationsrelation*
- *QoS-Kategorie* und *QoS-Parameter*
- *QoS-Parameter* und *Garantietyp*
- *QoS-Parameter* und *Semantik*
- *QoS-Parameter* und *Wertemenge*

sind bereits im vorangegangenen Abschnitt einleuchtend abgeklärt worden. Offen geblieben ist die Assoziation zwischen *Daten-Kategorie* und *QoS-Kategorie*. Um diese erklären zu können muss zunächst der Begriff des Flows im Kontext dieser Arbeit allgemein definiert werden.

Definition des Begriffs Flow
 Unter einem **Flow** soll auf Ebene der Netz-QoS-Architekturen verstanden werden: Ein Flow ist ein logisch zusammengehöriger Strom von Daten, die alle mit den gleichen Qualitätseigenschaften versehen sind und von einem Sender zu einer Menge von Empfängern im Kontext der *Kommunikationsrelation* einer *Daten-Kategorie*.

Flow im verbindungsorientierten Fall
 Für den konkreten Fall ist zur Unterscheidung von Flows die Eigenschaft der logischen Zusammengehörigkeit entscheidend: Für den Fall einer verbindungsorientierten *Daten-Kategorie* ist ein Flow direkt mit dem Verbindungsbegriff assoziiert: Genau ein Flow entspricht genau einer Verbindung und umgekehrt. Wird eine neue Verbindung erstellt, entspricht dies einem neuen Flow, auch dann, wenn die Empfängermenge identisch zu einem bereits bestehenden Flow ist.

Flow im verbindungslosen Fall
 Im verbindungslosen Fall muss wegen des Fehlens eines Zustands zur Unterscheidung von Flows die Empfängermenge hinzugezogen werden. Wie in der Definition festgelegt, bestimmt sich die Empfängermenge aus der Klasse *Kommunikationsrelation*. Für *Unicast*-Varianten ist dies genau ein Empfänger (1:1 Relation), für den *Multicast*-Fall besteht die Empfängermenge aus den n Empfängern in der 1:n oder m:n Multicast-Relation. Solange im verbindungslosen Fall die Empfängermenge konstant ist, spricht man vom gleichen Flow. Ändert sich die Empfängermenge, dann entspricht dies einem neuen Flow. Entsprechend der Definition erhält man

immer einen neuen Flow, sobald sich der Sender oder die den Daten anhaftenden Qualitätseigenschaften ändern.

Es sei an dieser Stelle darauf verwiesen, dass der Flow-Begriff im allgemeinen prinzipiell abhängig von der betrachteten Schicht im Protokollstapel ist. Beispielsweise für das HTTP-Protokoll⁴ der Anwendungsebene würde ein Flow deutlich feingranularer definiert werden. Hier wären unterschiedliche Flows auch noch unterscheidbar, wenn z.B. Sender und Empfänger gleich bleiben.

Ergänzung: Flow-Begriff ist schichtabhängig

Zur Festlegung der Kardinalität und der Klasse *Daten-Kategorie* untersucht man zunächst, wie viele Flows über eine Instanz einer *Daten-Kategorie* abgewickelt werden. Die grundsätzlichen Fälle hierfür sind:

- I) Alle Flows werden über genau eine Instanz der *Daten-Kategorie* abgewickelt.

Es existiert während der Betriebsphase immer genau eine Instanz der *Daten-Kategorie*. Häufig ist diese Variante bei verbindungslosen *Daten-Kategorien* (Attribut *Verbindungsorientiert* mit *false* belegt) vorzufinden, bei denen ein Instanzierung nicht einmal explizit, sondern nur implizit stattfindet.

Alle Flows werden über genau eine Instanz

Typische Beispiele hierfür sind die Internet-Architekturen Diff-Serv und IntServ. Für deren «Daten-Kategorie» *IP-Datagramm* (Abb. 4.10, S. 90) wird für die ganze Betriebsphase nur genau eine Instanz erzeugt. Sie entsteht implizit durch die Inbetriebnahme der Infrastruktur.

- II) Für mehrere Flows werden mehrere Instanzen einer *Daten-Kategorie* erzeugt.

Im Laufe der Betriebsphase sind mehrere Instanzen einer *Daten-Kategorie* vorhanden. Sie werden während der Betriebsphase erzeugt und auch während dieser Phase wieder zerstört. Es lassen sich Untervarianten unterscheiden:

Mehrere Flows über mehrere Instanzen

- a) Mehrere, aber nicht alle Flows werden einer *Daten-Kategorie* zugeordnet. Es werden mehrere Instanzen einer *Daten-Kategorie* erzeugt, diese sind dann jeweils für mehrere Flows zuständig.

Mehrere Flows teilen sich Instanz

Hierzu gibt es kein aktuelles Beispiel, die Variante ist theoretisch aber denkbar und wird im Sinne der Vollständigkeit der Modellierung erwähnt. Ein denkbare Beispiel wäre eine Architektur, die explizit nutzergesteuerte Aggregationsmechanismen bereitstellt.

- b) Für jeden Flow wird eine eigene Instanz einer *Daten-Kategorie* erzeugt. Das bedeutet, dass mehrere Instanzen einer *Daten-Kategorie* erzeugt werden, wobei jeder Instanz einer *Daten-Kategorie* für genau einen Flow zuständig ist. Dies ist typisch für verbindungsori-

Jeden Flow über eigene Instanz

⁴Hypertext Transfer Protocol (HTTP)

enterte *Daten-Kategorien* (Attribut *Verbindungsorientiert* mit *true* belegt).

ATM ist ein klassisches Beispiel dafür.

Graphischer Stereotyp für Singleton

Um in der Modellierung die Fälle I und II unterscheiden zu können, ist eine spezielle Kennzeichnung in den Klassendiagrammen notwendig. Der Fall mehrere Instanzen ist durch die Standard-Semantik der UML-Diagramme abgedeckt. Das Erzeugen von genau einer Instanz (Fall I) ist ein sog. Singleton-Pattern [GHJV 95]. Um dies im UML-Diagramm ausdrücken zu können, wird in dieser Arbeit ein graphischer Stereotyp verwendet, nämlich ein weißes „S“ auf rechteckigem schwarzem Grund.

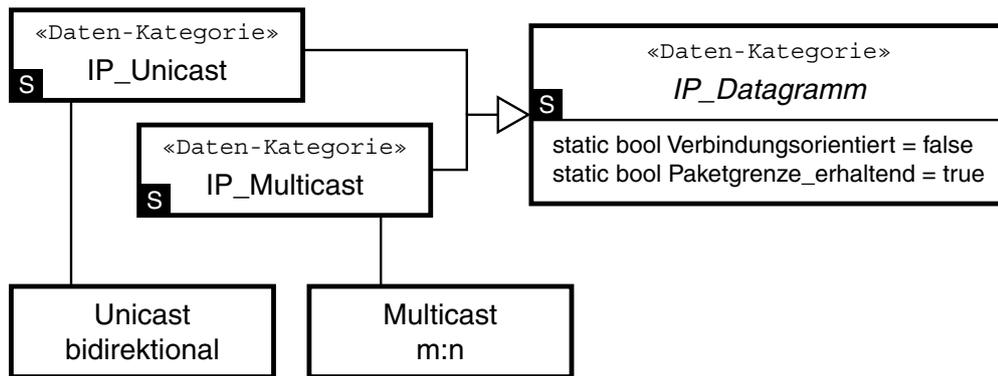


Abbildung 4.21: IP «Daten-Kategorien» mit Singleton-Notation

Wie im obigen Beispiel bereits gezeigt, ist die Klasse *IP-Datagramm* aus Abbildung 4.10, Seite 90 eine Daten-Klasse, für die das Singleton-Pattern gilt. Abbildung 4.21 zeigt die entsprechende Anwendung des graphischen Stereotypen.

Analoge Überlegung für QoS-Kategorien

Die gerade angestellten Überlegungen zur Anzahl der Instanzen bei *Daten-Kategorien* lassen sich völlig analog auf die Klasse *QoS-Kategorien* übertragen. Zu beachten ist, dass für die *QoS-Kategorie* nicht zwangsweise der gleiche Fall gilt wie für die *Daten-Kategorie*. Die Untersuchung der Instanzmenge muss also für beide Klassen getrennt erfolgen. Bei der Gruppe der sog. Class-of-Service (CoS)-Architekturen ist häufig der Singleton-Fall für *QoS-Kategorien* gegeben (so auch bei DiffServ).

Die IntServ-Architektur liefert ein Beispiel für unterschiedliche Instanzenfälle bei *Daten-Kategorien* und *QoS-Kategorien*. Für die *Daten-Kategorie* gilt Fall I, da IntServ auch auf IP basiert (siehe oben). Für die Vereinbarung eines QoS wird aber RSVP verwendet. Für jeden Flow ist mittels RSVP eine eigene Reservierung vorzunehmen, deren Einstellungen entlang des Pfades eines Flows von jedem Router gespeichert werden muss. Hier werden also für die *QoS-Kategorien* für jeden einzelnen Flow Instanzen erzeugt, womit Fall II.b (natürlich für *QoS-Kategorie* anstatt *Daten-Kategorie*) gegeben ist.

4.3.3.1 Auffaltungstransformation

Wenn bei *QoS-Kategorien* der Singleton-Fall vorliegt, dann muss auf die korrekte Semantik der Modellierungsstruktur geachtet werden. Besteht nämlich eine Singleton-Kategorie aus wenigstens einem *QoS-Parameter*, dessen Wertemenge aus mehr als einem Element besteht, dann muss das Modell evtl. transformiert werden. Für Abbildung 4.22 wird in der Betriebsphase genau eine Instanz der Kategorie *K* erzeugt, wobei der Parameter *P* entweder den Wert W_1 oder W_2 oder W_3 hat.

Anpassung des Modells für Singleton-QoS-Kategorien

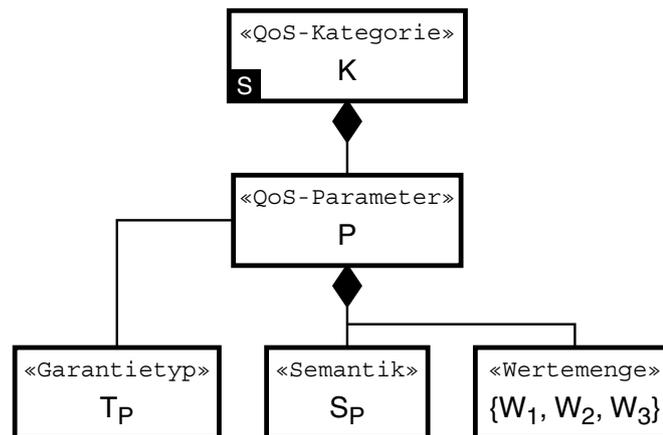


Abbildung 4.22: Singleton-«QoS-Kategorie»

Bei typischen CoS-Architekturen mit endlichen Wertemengen würden aber für jeden Wert aus der Wertemenge genau eine Instanz in der Nutzungsphase erzeugt werden. Ist diese Semantik gewünscht, dann muss dies im Modell explizit gemacht werden. Dies geschieht durch „Auffalten“ einer jeden *QoS-Kategorie*, dargestellt in Abbildung 4.23 für die Klasse *K*:

Auffaltung der QoS-Kategorien

- I) Zu Beginn wird für jeden «QoS-Parameter» *P* eine abstrakte Klasse P_a erstellt. Sie stellt die Verbindung zur «Semantik» S_P und zum «Garantietyp» T_P her.
- II) Anschließend wird für jeden Wert aus der «Wertemenge» eines jeden Parameters ein eigener «QoS-Parameter» erstellt, in Abbildung 4.23 sind das die Parameter P_1 , P_2 und P_3 . Der Zusammenhang zum ursprünglichen Parameter wird durch die zuvor erstellten abstrakten Klassen (hier P_a) hergestellt, von denen abgeleitet wird.
- III) Im dritten Schritt wird die ursprüngliche «QoS-Kategorie» durch eine abstrakte «QoS-Kategorie» K_a dargestellt.
- IV) Abschließend werden durch Bildung des Kreuzprodukts aus den Mengen der konkreten Klassen, die durch die «Wertemengen» der Parameter entstanden sind, konkrete «QoS-Kategorien» gebildet (hier K_1 , K_2 und K_3). Alle gebildeten Klassen werden von der vorher erstellten abstrakten «QoS-Kategorie» K_a abgeleitet.

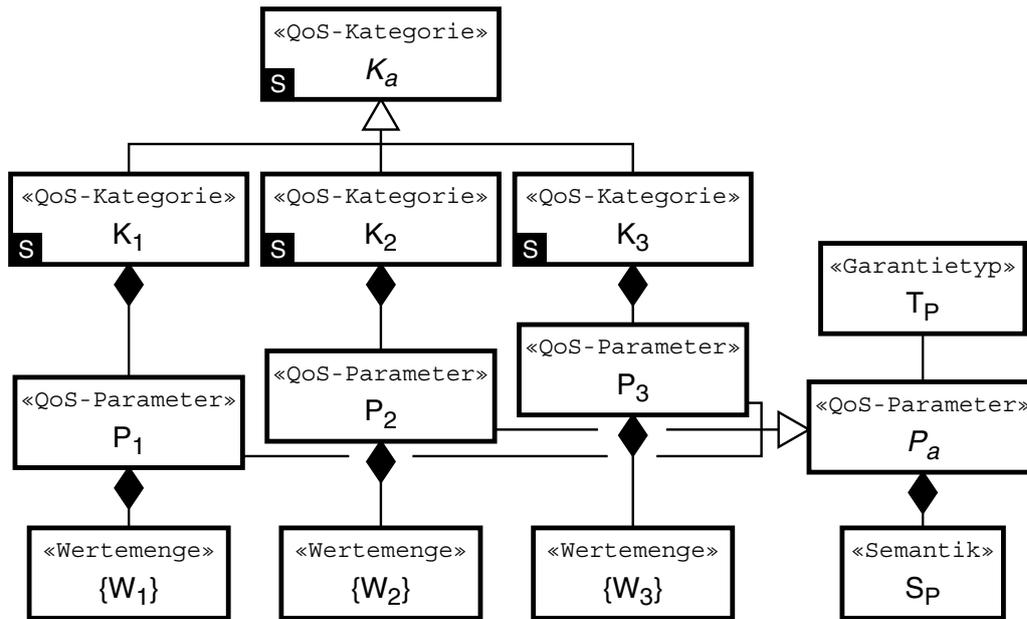


Abbildung 4.23: Aufgefaltene Singleton-«QoS-Kategorie»

Auf das Standardbeispiel DiffServ angewendet:

Für die in Unterabschnitt 4.3.1.4 in Abbildung 4.16, Seite 98 modellierte «QoS-Kategorie» *AssuredForwarding_PHB* von DiffServ Assured Forwarding PHB ist die eben beschriebene Auffaltung notwendig: Es wird für die «QoS-Kategorie» prinzipiell nur eine Instanz erstellt (CoS-Architektur), aber für alle möglichen Werte des «QoS-Parameter» *Loss* wird eine eigene Instanz gebildet. Die Abbildung 4.24 zeigt das Ergebnis der Auffaltung für DiffServ AF PHB.

4.3 MODELL FÜR NETZ-QoS-ARCHITEKTUREN

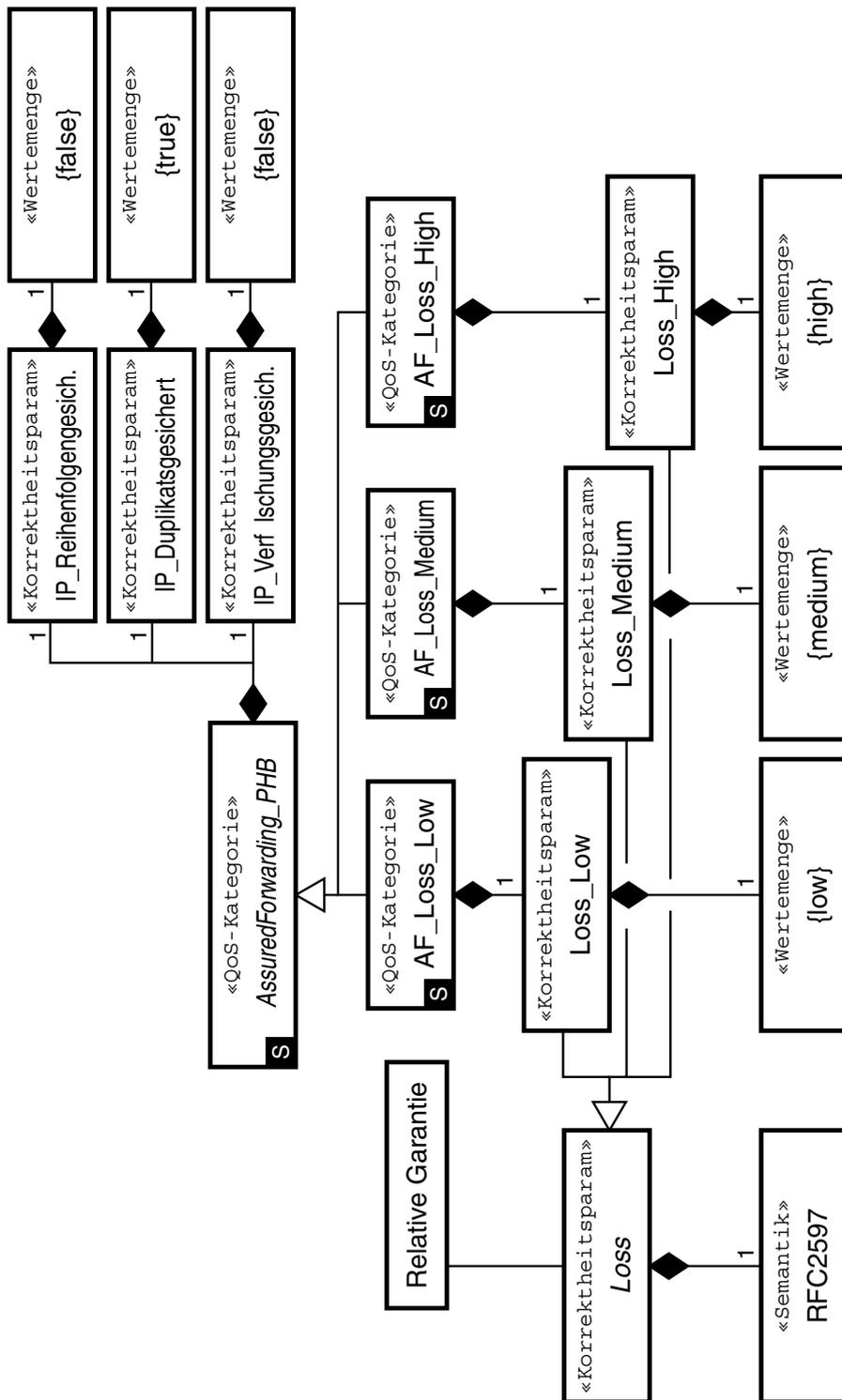


Abbildung 4.24: Aufgefaltene «QoS-Kategorien» von DiffServ AF PHB

4.3.3.2 Kardinalitäten der Session-Assoziation

Kardinalitäten der Session-Assoziation Nachdem für die beiden Klassen *Daten-Kategorie* und *QoS-Kategorie* die Anzahl der Instanzen zur Betriebsphase jeweils einzeln geklärt wurden, können diese in Relation zueinander gesetzt. Dies geschieht, wie von UML vorgesehen, schlicht durch Eintragung von Kardinalität an der Assoziation zwischen den beiden Klassen. Damit wird ein weiteres wesentliches strukturelles Merkmal zur Unterscheidung von Netz-QoS-Architekturen modelliert.

Häufigste Fälle Zwar sind prinzipiell beliebige Relationen zwischen *Daten-Kategorie* und *QoS-Kategorie* (d.h. beliebige Kardinalitäten) möglich und auch modellierbar, in den allermeisten Fällen wird aber eine der folgenden Varianten anzutreffen sein:

- *Daten-Kategorie* S – *QoS-Kategorie* S:
Von *Daten-Kategorien* und *QoS-Kategorien* gibt es jeweils genau eine Instanz (Abb. 4.25).

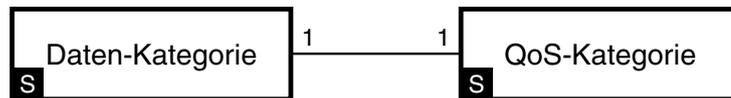


Abbildung 4.25: Kardinalitäten: Jeweils genaue eine Instanz

Wie am obigen Beispiel bereits gezeigt, ist DiffServ ein Vertreter dieser Variante.

- *Daten-Kategorie* S – *QoS-Kategorie* *:
Alle Instanzen von *QoS-Kategorien* teilen sich jeweils genau eine Instanz der *Daten-Kategorien* (Abb. 4.26).

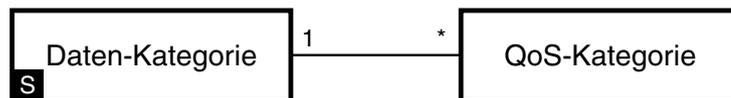


Abbildung 4.26: Kardinalitäten: Alle *QoS-Kategorien* teilen sich genau eine *Daten-Kategorie*

Die Internet-Architektur IntServ benutzt die gleiche «Daten-Kategorie» *IP_Datagramm* wie DiffServ (Abb. 4.10, S. 90). Für die QoS-Signalisierung wird RSVP benutzt, das für jede Reservierung einen eigenen Status hält [RFC 2205, RFC 2208, RFC 2209]. Das bedeutet, dass für jeden Flow bei IntServ eine eigene Instanz einer «QoS-Kategorie» erzeugt wird.

- *Daten-Kategorie* * – *QoS-Kategorie* S:
Alle Instanzen von *Daten-Kategorien* teilen sich jeweils genau eine Instanz der *QoS-Kategorien* (Abb. 4.27).

Eine solche Konstellation lässt sich erreichen, indem man bei DiffServ eine verbindungsorientierte «Daten-Kategorie» einführt, beispielsweise TCP.

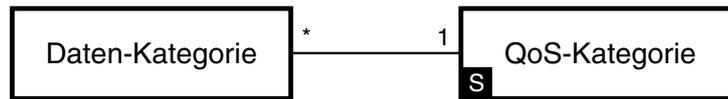


Abbildung 4.27: Kardinalitäten: Alle *Daten-Kategorien* teilen sich genau eine *QoS-Kategorie*

- *Daten-Kategorie 1 – QoS-Kategorie 1:*
Eine Instanz einer *Daten-Kategorie* ist mit genau einer Instanz einer *QoS-Kategorie* verbunden (Abb. 4.28).



Abbildung 4.28: Kardinalitäten: Jeweils eine *Daten-Kategorie* mit einer *QoS-Kategorie*

In der verbindungsorientierten ATM-Architektur werden QoS-Anforderungen im Zuge des Verbindungsaufbaus für eine «Daten-Kategorie» signalisiert. Damit wird für jeden Flow für «Daten-Kategorie» und «QoS-Kategorie» eine eigene Instanz erzeugt.

In diesem Abschnitt ist eine weitere Eigenschaft von Netz-QoS-Architekturen deutlich geworden, nämlich in den Kardinalitäten der Klassen *Daten-Kategorie* und *QoS-Kategorie*. Dieser Unterschied kann selbst dann bestehen, wenn zwei Architekturen ansonsten sehr ähnlich sind, beispielsweise in allen ihren *QoS-Parametern*. Damit sind die Kardinalitäten ein wesentliches Kennzeichen von Netz-QoS-Architekturen.

Unterschiedliche Kardinalitäten in sonst ähnlichen Architekturen

4.3.4 Dynamisches Modell durch Lebenszyklusanalyse

Neben den Kardinalitäten unterscheiden sich Netz-QoS-Architekturen wesentlich darin, wann ihre einzelnen Teile definiert, instanziiert und beendet / vernichtet werden. In weiteren Unterkapiteln wird hier erklärt, welche Vorgänge zu beachten sind und wie sich diese in den Lebenszyklus aus Abschnitt 4.3.2 einordnen.

Einordnung in Lebenszyklus

4.3.4.1 Variabilitätsanalyse und Grundvorgänge

Der nächste Schritt zur Erstellung des dynamischen Teils der Modellierung ist eine allgemeine Analyse der Veränderlichkeit über die Lebenszeit der einzelnen Klassen im statischen Grundmodell.

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

Die Klassen ...

- ... *Kommunikationsrelation* und alle davon abgeleiteten Klassen ...
- ... *Garantietyt* und alle davon abgeleiteten Klassen ...

Markierungsklassen nicht veränderlich ... sind reine Markierungsklassen, die zur Kennzeichnung bestimmter Eigenschaften dienen. Diese Klassen tragen die im Laufe von Abschnitt 4.3.1 definierte feste Semantik und haben keine zu belegenden Werte. Bei diesen Klassen besteht also keinerlei Veränderlichkeit, auch nicht bei der Instanziierung des statischen Modells für eine konkrete Netz-QoS-Architektur.

Gründe für Variabilität Anders verhält es sich mit den Klassen *Daten-Kategorie*, *Session*, *QoS-Kategorie*, *QoS-Parameter*, *Semantik* und *Wertemenge*. Diese sind aus einer der folgenden Gründe variabel:

I) Konkretisierung durch spezifische Semantik

Ein Standard lässt (bewusst oder auch nicht) Freiheitsgrade in der Definition von Semantiken, die erst bei konkreter Umsetzung definiert werden müssen (z.B. *QoS-Kategorien* bei DiffServ AF PHB Unterabschnitt 4.3.1.4).

II) Belegung von Attributen

Erst beim Erzeugen einer Instanz werden Attribute mit Werten belegt (z.B. *QoS-Parameter*).

III) Kompositionsrelation zu einer variablen Klasse

Eine Klasse, die sich aus einer variablen Klasse zusammensetzt, wird erst festgelegt, wenn auch die variable Klasse festgelegt ist (z.B. *QoS-Parameter* oder *QoS-Kategorie*).

Grundvorgänge Analysiert man die Variabilitätsgründe I und II systematisch auf Klassen und deren Instanzen, dann ergeben sich fünf charakteristische **Grundvorgänge** für das dynamische Modell:

■ Definition

Festlegung der Teilklassen einer Klasse, deren Semantik und der Semantik der Klasse selbst.

■ Belegung

Festlegung von Klassen und Attributen, die zwar im Rahmen der Definition noch variabel sind, aber für alle Instanzen der Klassen gleich sein müssen, d.h. vor der Erzeugung einer Instanz belegt werden müssen.

■ Instanziierung

Erzeugung einer Instanz und Festlegung von Werten, die konstant über die Lebenszeit der Instanz sind. Die Werte können sich aber zwischen unterschiedlichen Instanzen unterscheiden.

■ Veränderung

Manipulation von Werten einer Instanz während ihrer Lebenszeit.

■ Beendigung

Löschen bzw. zerstören einer Instanz.

4.3.4.2 Lebensabschnittskorridore der Grundvorgänge

Für jede Klasse im statischen Modell können jetzt charakteristische Zeiten unterschieden werden, zu denen die fünf entscheidenden Grundvorgänge einer jeden Klasse stattfinden. Hierbei handelt es sich nicht um singuläre Zeitpunkte, sondern vielmehr um Zeitabschnitte. Für jeden der im folgenden erläuterten Grundvorgänge gibt es demnach jeweils einen frühesten bzw. spätesten Zeitpunkt, wann der Grundvorgang auftreten bzw. ausgeführt werden kann. Die Zeitspanne zwischen dem frühesten und spätesten Zeitpunkt wird als **Lebensabschnittskorridor** bezeichnet.

Einordnung von Grundereignissen in Zeitkorridore

Die nachstehende Tabelle 4.2 gibt für jeden Grundvorgang einen eigenen Korridor an. Zusätzlich soll mit dem Begriff **Lebenskorridor** der Zeitraum zwischen Instanziierung und Beendigung bezeichnet werden.

Korridor	Grundvorgang	Zwingend vorhanden	Relative Lage	Lage bzgl. Lebenszyklus
Definitions-korridor	Definition	ja	–	$tS_{DK} \geq tS_{Standard}$ $tE_{DK} < tS_{Betrieb}$
Belegungs-korridor	Belegung	nein	$tS_{BelK} > tE_{DK}$	$tS_{Belegung} \geq tS_{Planung}$ $tE_{BelK} < tS_{Betrieb}$
Instanziierungs-korridor	Instanziierung	ja	$tS_{IK} > tE_{DK}$ $tS_{IK} > tE_{BelK}$	$tS_{IK} \geq tS_{Bereitstellung}$ $tE_{IK} \leq tE_{Nutzung}$
Veränderungs-korridor	Veränderung	nein	$tS_{VK} > tE_{IK}$	$tS_{VK} \geq tS_{Aenderung}$ $tE_{VK} \leq tE_{Aenderung}$
Beendigungs-korridor	Beendigung	ja	$tS_{BeendK} > tE_{IK}$ $tS_{BeendK} > tE_{VK}$	$tS_{BeendK} > tS_{Betrieb}$ $tE_{BeendK} < tE_{Beendigung}$

Tabelle 4.2: Korridore der Grundvorgänge

Allerdings ist für eine konkrete Netz-QoS-Architektur nicht jeder Korridor für jede Klasse zwingend vorhanden:

Optionale Korridore

- Der Belegungskorridor wird nur benötigt, wenn Attribute belegt werden müssen, die für alle Instanzen identisch sein müssen. In den Klassendiagrammen sind solche Attribute mit *static* gekennzeichnet, aber es darf noch kein konkreter Wert zugewiesen sein.
- Nur wenn während der Lebenszeit einer Instanz Veränderungen an ihr vorgenommen werden können, dann ist der Veränderungskorridor vorhanden.

In der Tabelle finden sich ebenfalls Bedingungen an die frühesten Startzeiten tS_X und die spätesten Endezeiten tE_X . Diese werden zum einen relativ zu den anderen Korridoren angegeben, zum anderen wird auch ihre Lage bezüglich des Lebenszyklus (Abschn. 4.3.2) bestimmt.

Zeitraumen für Korridore

Korridormodelle für Klassen Bei einer Instanziierung des Modells für eine konkrete Netz-QoS-Architektur müssen für fast jede Klasse im statischen Grundmodell die zugehörigen Korridormodelle angegeben werden. Nur für die reinen Markierungsklassen macht dies keinen Sinn, also für die Klassen *Kommunikationsrelation*, *Garantiertyp* und deren Spezialisierungen.

Korridore unterscheiden sich zwischen Architekturen Notwendig ist die Modellierung der Korridore, weil sich die Zeitspannen dieser Korridore bezüglich des Lebenszyklus in den verschiedenen Architekturen erheblich unterscheiden können. Es ist ein wesentlicher Punkt der Koppelungsmethodik, diese Unterschiede festzustellen und adäquat zu behandeln.

Beispielsweise der Instanzierungskorridor von *Daten-Kategorien*: Bei ATM-SVCs liegt er innerhalb der Grenzen der Nutzungsphase, da ATM-Verbindungen erst dann auf- und abgebaut werden können.

Bei DiffServ dagegen müssen die Singleton-«Daten-Kategorien» zu Beginn der Nutzungsphase bereits erstellt sein, damit sie genutzt werden können. Ihre Instanzierung geschieht nämlich implizit durch die Installation und Konfiguration der DiffServ-Netzkomponenten. Also ist der Instanzierungskorridor der «Daten-Kategorien» bei DiffServ in die Bereitstellungsphase eingebettet.

Beim Vergleich zwischen ATM und DiffServ erkennt man, dass die Instanzierungskorridore nicht nur unterschiedliche Ausdehnung haben, sondern sie besitzen nicht einmal eine Zeitspanne der Überschneidung im Lebenszyklus.

Mit den Korridormodellen wurde eine für jede Netz-QoS-Architektur individuelle Unterteilung des Lebenszyklus geschaffen. Diese wird auch im folgenden Abschnitt benötigt und dort als Referenzpunkte für Ereignisse dienen.

4.3.4.3 Ereignisse in den Korridoren

Ereignisse modellieren dynamisches Verhalten Zur näheren Modellierung des dynamischen Verhaltens einer Netz-QoS-Architektur kommen Ereignisdiagramme zum Einsatz.

Ereignisse sind Nachrichten, die eine eigene Semantik tragen und optional mit zusätzlichen Daten angereichert werden können. Die zusätzlichen Daten werden z.B. gebraucht, um Optionen zu spezifizieren. In dieser Modellierung werden zwei Grundarten von Ereignissen unterschieden:

■ Spontanereignisse

Spontanereignisse So werden Ereignisse bezeichnet, die von der Netz-QoS-Architektur ohne vorangegangene Anforderung durch den Nutzer an diesen geschickt werden.

Ein typisches Beispiel für ein Spontanereignis findet sich bei der «QoS-Kategorie» *ABR_Kat* (Abb. 4.17, S. 101) von ATM. Hier können nach [ATM-TM 99, Abschn. 2.1.5 und 5.10] Veränderungen in den Qualitätseigenschaften der ATM-Verbindung an den Nutzer signalisiert werden.

■ Interaktionsereignisse

Als Interaktionsereignisse werden solche Ereignisse benannt, die von einem Nutzer an eine Netz-QoS-Architektur gesandt werden. Häufig, aber nicht verpflichtend, folgt ein Rückereignis von der Netz-QoS-Architektur zum Nutzer, um Ergebnisse oder Bestätigungen zu übermitteln. Interaktionsereignisse

ATM-Verbindungen liefern hier ein klassisches Beispiel für Interaktionsereignisse. Zum Aufbau einer Verbindung sendet der Nutzer ein entsprechendes Ereignis an die ATM-Architektur. Mit den zusätzlichen Daten des Ereignisses spezifiziert der Nutzer die gewünschte *QoS-Kategorie* und die Belegung ihrer *QoS-Parameter*. Die ATM-Architektur antwortet mit einem Rückereignis, bei dem angegeben wird, ob die Verbindung mit den gewünschten Eigenschaften aufgebaut werden konnte.

Die zu modellierenden Ereignisse finden sich in den Architekturen an hauptsächlich zwei Stellen. Zum ersten sind dies die konkreten Protokolle, die die Kommunikation mit dem Nutzer steuern. Die Aufrufprimitiven, die vom Nutzer verwendet werden können, werden als Interaktionsereignisse modelliert. Benachrichtigungen und Events, die von der Architektur an den Nutzer geschickt werden, sind als Spontanereignisse zu modellieren. Ereignisquelle
Protokolle

Die zweite Quelle für Ereignisse sind die Workflows, die (meist nur implizit) in den Standards enthalten sind. Wenn in einem Ablauf in einer Netz-QoS-Architektur eine Interaktion mit dem Nutzer möglich ist, dann muss diese als entsprechendes Ereignis modelliert werden. Um Nutzerinteraktionen zu finden, müssen die Standardisierungsdokumente manuell untersucht und interpretiert werden. Ereignisquelle
Workflows

Bei beiden Quellen für Ereignisse sind die gefundenen Ereignisse den Klassen aus der Modellierung zuzuordnen. Es sei angemerkt, dass nicht nur Ereignisse der Betriebsphase zu modellieren sind, sondern jegliche Ereignisse im gesamten Lebenszyklus. Ereignisse werden
Klassen zugeordnet

Um die Ereignisse, die für eine Klasse auftreten, strukturieren zu können, muss eine passende Zeiteinteilung gefunden werden. Zunächst würde sich dafür der Lebenszyklus aus Abbildung 4.20, Seite 106 anbieten. Eine solche Einteilung wäre aber nicht optimal: Später in der Methodik gilt es zu kopelnde Ereignisse aufzufinden. Gruppiert man Ereignisse aber in die Lebenszyklusphasen, dann müssten zusammengehörige Ereignisse auch in unterschiedlichen Phasen gesucht werden. Das Beispiel der Instanziierung von DiffServ und ATM hat das bereits belegt. Zeitachse für
Ereignisse benötigt

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

Korridormodelle als Zeiteinteilung für Ereignisse	Stattdessen werden die Korridormodelle zur Gruppierung der Ereignisse herangezogen. Durch die Einteilung nach den Korridoren werden die Ereignisse semantisch so gruppiert, dass ein Vergleich zweier gleicher Korridore zur Koppelung ausreichend sein wird. Dabei geht aber der Bezug zum Lebenszyklus trotzdem nicht verloren, da nach dem vorangegangenen Abschnitt die Korridore relativ zum Lebenszyklus eingeordnet wurden.
	Zur Spezifikation der Ereignisse wird im Prinzip für jeden Zeitkorridor ein eigenes Ereignisdiagramm erstellt. Dabei müssen nicht zwangsweise für jeden der in Tabelle 4.2 genannten Korridore Ereignisse vorliegen. Dem Definitionskorridor wird man wahrscheinlich extrem selten Ereignisse zuordnen müssen. Die entscheidenden Korridore für den Ereignisaustausch zwischen Architektur und Nutzer sind der Instanziierungs- und der Veränderungskorridor.
Einzelne Ereignisse sehr individuell	Da Anzahl, Art und Semantik der einzelnen Ereignisse sich bei konkreten Netz-QoS-Architekturen stark unterscheiden können, ist hier eine Grenze der generischen Modellierung erreicht. Außer der Einteilung in Spontan- und Interaktionsereignisse wird an dieser Stelle keine nähere Klassifizierung getroffen.
Übernehmen von Ereignisdiagrammen aus Standards	Die Ereignisdiagramme der einzelnen Klassen können manchmal direkt aus den Standards gewonnen werden. Beispielsweise machen ATM in [ATM-PNNI 02] und UMTS in [TS 123 207, TS 123 246] selbst massiv Gebrauch von Sequenzdiagrammen. Allerdings können diese Diagramme meist nicht unmittelbar übernommen werden, da sie sowohl Interaktionen an der Dienstschnittstelle und interne Aktionen miteinander vermischen. Für die Black-Box-Sicht sind nur Ereignisse an der Dienstschnittstelle von Interesse. Bevor die Diagramme genutzt werden können, ist diese Trennung manuell vorzunehmen. Durch das bereits erstellte statische Modell kann ermittelt werden, welche Teile aus den Sequenzdiagrammen der Standards für die dynamische Modellierung zu übernehmen sind. Dies geschieht, indem nur Ereignisse und Sequenzen übernommen werden, die in direkter Verbindung mit einer im statischen Modell vorhandenen Klasse stehen.

4.3.5 Zusammenfassung des Modells für Netz-QoS-Architekturen

In dieser Sektion sind mehrere Teilmodelle zur Darstellung von Netz-QoS-Architekturen entwickelt worden:

- Mit dem **statischen Grundmodell** (Abschn. 4.3.1) sind die ersten wichtigen Unterschiede zwischen verschiedenen Netz-QoS-Architekturen aufgezeigt worden.
- Mit den **Modellen der Lebensabschnittskorridore** (Unterabschn. 4.3.4.2) wurde eine zeitliche Unterteilung der dynamischen Abläufe in einer Netz-QoS-Architektur erreicht und gleichzeitig wurde der Zusammenhang zu den Lebenszyklusphasen aufgezeigt.
- Eine Feinmodellierung der Aktionen in den Lebensabschnittskorridoren ist schließlich durch **Ereignisdiagramme** (Unterabschn. 4.3.4.3) realisiert worden.

Diese Teilmodelle zusammen bilden das Modell für Netz-QoS-Architekturen, wie es in Abbildung 4.1, Seite 68 als Ergebnis des zweiten Teilschritts angekündigt wurde. Mit den Teilmodellen wird auch die phänomenologische Sicht auf Netz-QoS-Architekturen definiert. Rein interne Ereignisse einer Architektur spielen in der durch die Modellierung gewonnenen Darstellung keine Rolle mehr.

Teilmodelle bilden zusammen Architekturmodell

Damit ist die in Abschnitt 4.2.2 geforderte Black-Box-Sicht zumindest für die Ebene der Architekturen realisiert. In der nächsten Sektion wird auf Basis der Teilmodelle für Architekturen ein Modell für Netz-QoS-Dienste entwickelt. Mit diesem Modell wird die Black-Box-Sicht auch auf die Ebene der Dienste ausgedehnt.

Black-Box-Sicht für Architekturen umgesetzt

4.4 MODELL FÜR NETZ-QoS-DIENSTE

Nachdem mit der vorangegangenen Sektion Netz-QoS-Architekturen im Sinne einer dienstorientierten Black-Box-Sicht modelliert werden können, ist es die Aufgabe dieser Sektion ein Modell für Netz-QoS-Dienste zu entwerfen. Die grundlegenden Teile zur Modellerstellung sind zum einen die Teilmodelle der Architekturmodellierung, zum anderen ist es das MNM-Dienstmodell.

4.4.1 Individualisierung des Architektur-Modells

- Funktionsumfang im Dienst nicht zwangsweise identisch zur Architektur
- Wie bereits in Abschnitt 2.2.4 festgestellt, können sich die Eigenschaften eines Netz-QoS-Dienstes deutlich von denen der Netz-QoS-Architektur, auf der er beruht, unterscheiden. Damit besteht der erste Schritt zur Erstellung eines Modells für Dienste in der Anpassung der Modellierung der Architektur auf den Dienst. In der Dienstausprägung kann die zugrunde liegende Architektur im Prinzip eingeschränkt oder auch erweitert werden. Für beide Fälle werden im folgenden Beispiele angegeben werden.
- Änderungen theoretisch beliebig
- Da die für einen Dienst möglichen Änderungen theoretisch nahezu beliebig sind, können diese in den folgenden Abschnitten nur exemplarisch aufgezeigt werden. Generell können bei einer Dienstausprägung alle Teile einer Architektur und damit des Architekturmodells verändert werden.
- Änderungsmöglichkeiten in der Praxis begrenzt
- In der Praxis werden sich die Modifikationen allerdings in Grenzen halten, da massive Änderungen an einer Architektur der Definition einer neuen QoS-Architektur gleichkommen würde. Einschränkungen der Modifikationsvielfalt sind insbesondere auch durch die auf dem Markt befindlichen Netzkomponenten gegeben. Sie orientieren sich nämlich sinnvollerweise an den gegebenen Standards und lassen meist wenig Raum für individuelle Modifikationen. Anders verhält es sich in der Praxis nur dort, wo Standards zu ungenau sind oder sogar bewusst individuelle Erweiterungen zulassen oder erzwingen.

4.4.1.1 Anpassungen am statischen Grundmodell

- Typische Anpassungen von Daten-Kategorien
- Typische Anpassungen im Bereich der *Daten-Kategorie* liegen in der Beschränkung der *Kommunikationsrelation*. Hier werden seitens der Provider getrennte Dienste für Uni- und Multicast angeboten, obwohl sie auf der gleichen Infrastruktur (d.h. gleichen Netz-QoS-Architektur) realisiert sind. Für die Modellierung eines reinen Unicast-Dienstes werden also alle *Daten-Kategorien*, die mit *Multicast*-Varianten assoziiert sind, gestrichen. Parallel muss dabei darauf geachtet werden, dass im Bereich der *QoS-Kategorien* die exklusiv für *Multicast*-Varianten benötigten Teile gestrichen werden.

Für das Entfernen der Multicast-Kommunikation aus einem UMTS-Dienst müssten gestrichen werden:

- Die Klassen *MBMS_Multicast*, *UMTS_MBMS_Session* und *UMTS_MBMS_QoS* aus Abbildung A.1
- Alle Klassen aus Abbildung A.4
- Alle Klassen aus Abbildung A.5

Die Ausprägungen der Attribute *Verbindungsorientiert* und *Paketgrenze_erhaltend* sind meist so tief in den Architekturen verwurzelt, dass eine Anpassung nicht möglich ist. Tief verwurzelte Eigenschaften

Die Möglichkeiten der Anpassung sind auf Seiten der *QoS-Kategorie* weit vielfältiger. Im Bereich der *QoS-Parameter* sind Modifikationen in allen drei der einen Parameter definierenden Klassen denkbar, beispielsweise: Typische Anpassungen von QoS-Kategorien

- Der *Garantietyp* könnte aufgeweicht werden, z.B. von *Deterministische Garantie* zu *Statistische Garantie*
- Die *Wertemenge* könnte verändert werden, weil die Implementierungsplattform den seitens des Standards vorgesehenen Bereich nicht abdecken kann.
- Die Semantik könnte angepasst werden, weil andere Messpunkte in der Dienstimplementierung benutzt werden als von Seiten des Standards vorgesehen.

Eine *QoS-Kategorie* kann modifiziert werden, indem man einen oder mehrere *QoS-Parameter* herausstreicht. Selbstverständlich könnten auch ganze *QoS-Kategorien* von einem Dienst nicht unterstützt werden, beispielsweise keine *realTime_Kategorien* (Abb. 4.17, S. 101) der ATM-Architektur. Streichen einzelner Parameter

Teilweise sind aber auch Ergänzungen notwendig, um den vollen vom Standard definierten Umfang auszuschöpfen. Modifikation durch notwendige Ergänzungen

Der DiffServ Standard Assured Forwarding PHB [RFC 2597] (siehe Unterabschn. 4.3.1.4 und Abb. 4.16) ist ein Beispiel hierfür. Im Standard sind zwar die «QoS-Kategorien» *AF_1* bis *AF_4* definiert, es werden aber keine *QoS-Parameter* definiert, die die Kategorien voneinander unterscheidbar machen. Stattdessen wird im Standard explizit dazu aufgefordert, eigene Unterscheidungsmerkmale, sprich *QoS-Parameter*, zu definieren.

4.4.1.2 Anpassungen der dynamischen Modellteile

Die Anpassungen an den dynamischen Modellteilen (Korridor- und Ereignismodelle) orientieren sich zunächst an jenen, die im statischen Teil vorgenommen worden sind. Bei Streichungen im statischen Teil sind die korrespondierenden Teile im dynamischen Modell natürlich ebenfalls vorzunehmen. Wurden Ergänzungen gemacht, sind die dynamischen Teilmodelle ebenfalls zu ergänzen. Anpassung im dyn. Teil folgen dem stat. Teil

Wenn die Modellierung eines Dienstes möglichst eindeutig sein soll, dann ist darauf zu achten, dass in den Korridormodellen keine Definitionskorrido-

Keine Definitionskorridore mehr re vorhanden sind, die über die Standardisierungsphase hinausgehen. Würde dies nämlich der Fall sein, bedeutete das, dass im Dienst noch Freiheitsgrade in der Definition vorhanden sind.

Zeitl. Beschränkung von Korridoren Eine weitere typische Modifikation an den Korridormodellen ist die Verlegung bzw. Beschränkung von Korridoren in bestimmte Lebenszyklusphasen.

Soll beispielsweise ein ATM-basierter Dienst angeboten werden, der nur feste, unveränderliche Verbindungen über die gesamte Betriebsphase gestattet (z.B. für ein VPN-Szenario), dann kann dies sehr einfach dargestellt werden. Dazu begrenzt man die Instanzierungskorridore aller Klassen auf das Ende der Bereitstellungsphase. Das bewirkt, dass alle Verbindungen vor der Nutzungsphase aufgebaut werden müssen. Um Veränderungen an den Verbindungen zu unterbinden streicht man dann noch alle Veränderungskorridore aus den Modellen heraus. Somit bekommt man insgesamt das gewünschte Verhalten.

Modifikationen in Ereignismodellen Detailmodifikationen sind auch in den Ereignismodellen denkbar. So könnten in realen Diensten beispielsweise bestimmte Funktionalitäten während der Betriebsphase gewollt nicht angeboten werden. In den Ereignismodellen fände das dann seinen Niederschlag, indem die entsprechenden Interaktionsereignisse herausgestrichen würden.

4.4.2 Einbettung in das MNM-Dienstmodell

Basis MNM-Dienstmodell Die Basis für das eigentliche Modell von Netz-QoS-Diensten ist das MNM-Dienstmodell, genauer gesagt die Dienstsicht (Abb. 4.4, S. 73) davon.

Modellierung nur im seitenunabhängigen Teil Für eine realisierungsunabhängige Modellierung, wie sie als eine der Anforderungen aus Abschnitt 2.4.3 gefordert ist, muss in der Dienst-Sicht nur der seitenunabhängige Teil (Unterabschn. 4.2.1.1) spezifiziert werden. Die Modellteile in der Dienstnehmerdomäne und der Dienstleisterdomäne enthalten nur realisierungsspezifische Teile (Unterabschn. 4.2.1.2 und Unterabschn. 4.2.1.3). Beide bleiben deswegen unbetrachtet.

In den nächsten beiden Abschnitten wird nun das MNM-Dienstmodell so weiterentwickelt, dass es zum Modell für Netz-QoS-Architekturen wird.

4.4.2.1 Nutzungsspezifische Teile

Die linken Teile der Dienstsicht des MNM-Dienstmodells beschreiben jene Aspekte eines Dienstes, die für die Nutzfunktionalität von Belang sind. Da an dieser Stelle nur der seitenunabhängige Teil relevant ist, sind im einzelnen die Modellteile *QoS parameters*, *usage functionality* und *service access point* (SAP) zu verfeinern.

Der Teil *usage functionality* wird im Fall eines Netz-QoS-Dienstes genau durch die Klasse *Daten-Kategorie* des statischen Grundmodells für Netz-QoS-Archi-

tekturen beschrieben. Mit ihr werden die unterschiedlichen vom Dienst zur Verfügung gestellten Varianten zur Übermittlung von Daten beschrieben – und genau das ist die Nutzfunktionalität eines Netz-QoS-Dienstes.

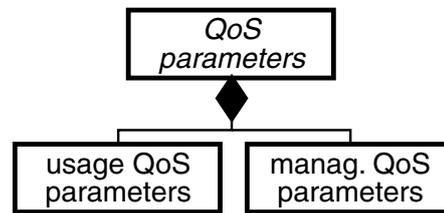


Abbildung 4.29: Aufgeteilte Klasse *QoS parameters*

Bevor die Dienstgüteeigenschaften eines Datenstroms im Modell dargestellt werden können, muss zuerst eine kleine Erweiterung der Dienstsicht vorgenommen werden. Die im MNM-Dienstmodell vorhandenen *QoS parameters* beziehen sich implizit sowohl auf die *usage functionality* als auch die *management functionality*. Durch die in [Abbildung 4.29](#) dargestellte Erweiterung der Klasse *QoS parameters* wird dieser Umstand explizit ausgedrückt: Die Klasse *QoS parameters* besteht nun aus getrennten Parametern für beide Funktionalitäten.

Auftrennung der „QoS parameters“

Nach der Auftrennung der *QoS parameters* werden die *usage QoS parameters* durch die entsprechende Klasse aus dem statischen Grundmodell ersetzt, nämlich der *QoS-Kategorie*. Es sei angemerkt, dass nicht die Klasse *QoS-Parameter* aus dem statischen Grundmodell an die Stelle der *usage QoS parameters* treten kann, da sonst die durch die *QoS-Kategorien* vorgenommene Gruppierung von Parametern nicht mehr im Dienstmodell vorhanden wäre.

Usage QoS parameters

Zwischen den in die Dienstsicht integrierten Klassen *Daten-Kategorie* und *QoS-Kategorie* kommt wie im statischen Grundmodell die Assoziationsklasse *Session* zum Einsatz. So wird der Zusammenhang zwischen *Daten-Kategorie* und *QoS-Kategorie* auch im Dienstmodell explizit ausgedrückt.

Jetzt ist das statische Grundmodell vollständig in die Dienstsicht und damit in das MNM-Dienstmodell integriert. In [Abbildung 4.30](#) ist die erweiterte Dienstsicht komplett dargestellt. Die dunkelgrau markierten Klassen stammen aus dem statischen Grundmodell, die hellgrau markierten sind durch die Aufspaltung der *QoS parameter* verändert oder eingeführt worden.

Statisches Grundmodell vollständig integriert

Bei der Modellierung konkreter Dienste werden natürlich nicht die ursprünglichen Modelle der Netz-QoS-Architekturen in die Dienstsicht eingebettet. Stattdessen kommen die nach den Dienstspezifika individualisierten Modelle (siehe den vorangegangenen Abschnitt) zum Zuge.

Verblieben ist noch die Frage nach der Spezifikation des *service access point* (SAP). Hierfür werden zusätzliche Spezifikationen benötigt. Zwar ist die am *service access point* zugreifbare Funktionalität durch die Klasse *Daten-Kategorie* beschrieben, aber die technischen Aspekte des *service access point* sind noch offen geblieben. Im Fall eines Netz-QoS-Dienstes stehen hier zwei Dinge im Vordergrund:

Service access point

KAPITEL 4: MODELLIERUNG VON NETZ-QoS-DIENSTEN

- Konkrete Datenformate zur Übermittlung von Daten (d.h. zur Nutzung der *Daten-Kategorie* spezifizierten Funktionalität)
- Konkrete Datenformate zur Anforderung und Zuordnung von Qualitätseigenschaften (d.h. zur Nutzung der *QoS-Kategorien*).
- Beschreibungen der technischen Gestaltung des Anschluss- oder Übergabepunktes zum Kunden. Häufig wird auch eine Spezifikation der im Schichtenmodell unterhalb der Netz-QoS-Architektur befindlichen Technologie beteiligt sein.

Datenformate am SAP Die Datenformate sind in den allermeisten Fällen in den gleichen Spezifikationsdokumenten beschrieben, in denen die Netz-QoS-Architektur (auf der der Dienst basiert) beschrieben wird. Zur Beschreibung dieses Aspekts des *service access point* kann direkt auf die entsprechenden Dokumente verwiesen werden.

Techn. Aspekte des SAP Die Beschreibung des technischen Übergabepunktes ist absolut dienstspezifisch und kann deshalb stark variieren. Für die „unteren“ Schichten stehen wiederum die jeweiligen Standardisierungsdokumente bereit. Teilweise werden solche Spezifikationen schon von Netz-QoS-Architekturen mitgebracht, ATM ist ein Beispiel dafür.

Für einen DiffServ-AF-PHB-basierten Dienst kann der *service access point* z.B. so beschrieben sein: Beim Kunden wird durch den Provider ein Router aufgestellt. Dieser hat als Übergabepunkt zum Kunden ein 100MBit Ethernet Interface (Spez. der unteren Schicht). Die Datenformate werden durch die IP- und DiffServ-AF-PHB-Standards beschrieben.

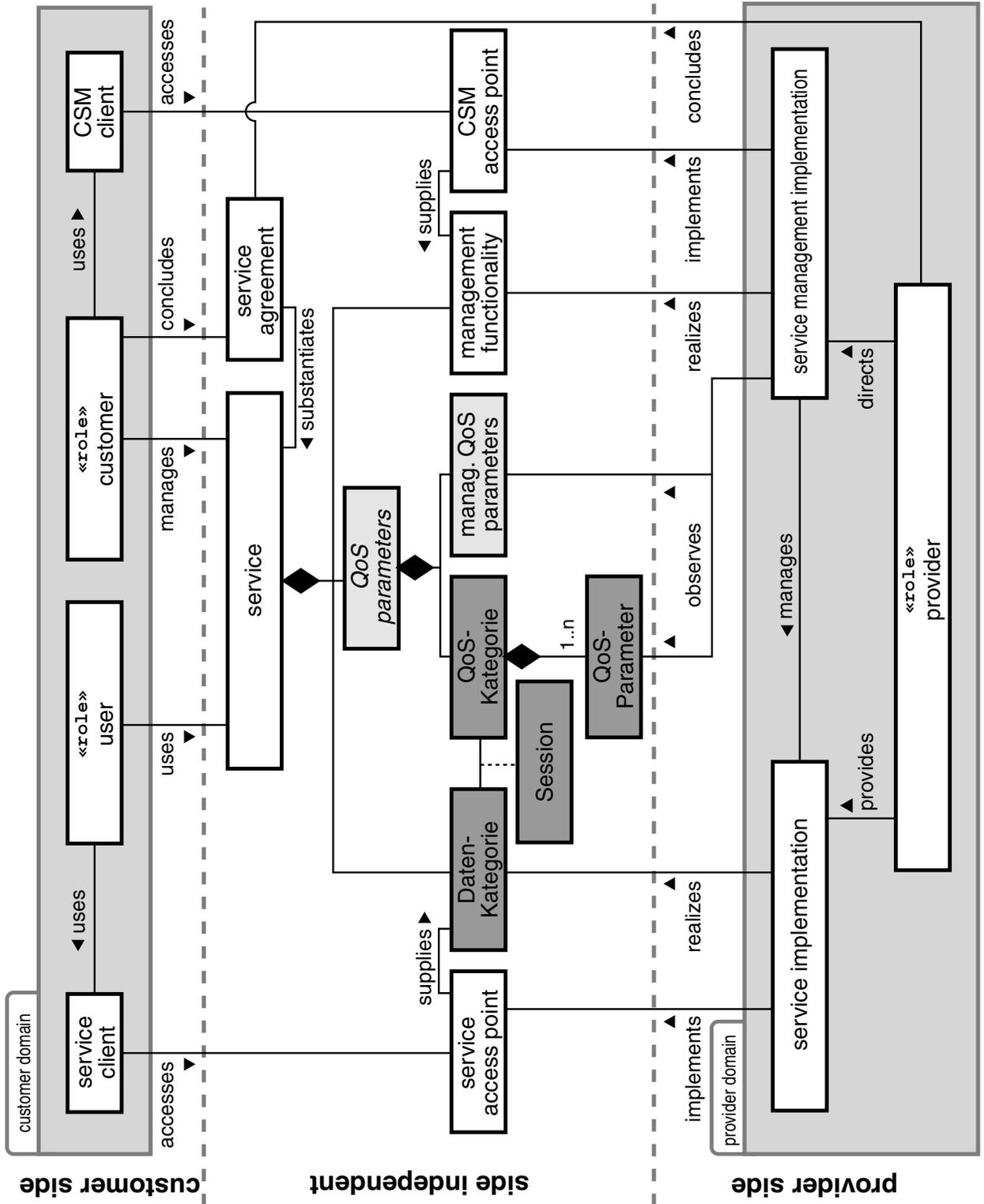


Abbildung 4.30: Erweitertes MNM-Dienstmodell in der Dienstsicht

4.4.2.2 Managementspezifische Teile

Managementfunktio-
nalität Der managementspezifische Teil eines Netz-QoS-Dienstes wird durch die rechte Hälfte der erweiterten Dienstsicht im MNM-Dienstmodell (Abb. 4.30) repräsentiert. In der *management functionality* werden ausschließlich Funktionalitäten modelliert, die dem Kunden zur Verfügung gestellt werden. Daher rührt auch der Begriff des „Customer Service Managements“. Managementfunktionalitäten, die ein Provider intern im Rahmen der Realisierung des Dienstes benutzt, sind hier nicht enthalten.

CSM individuell Die Spannbreite der potentiell auftretenden Funktionalitäten umfasst dabei alle Teile der fünf Funktionsbereiche wie sie im Funktionsmodell der OSI-Managementarchitektur definiert sind: Konfigurationsmanagement, Fehlermanagement, Leistungsmanagement, Sicherheitsmanagement und Abrechnungsmanagement. Die Ausprägung der einzelnen Teilbereiche ist dabei nahezu beliebig. Sie kann von „nicht vorhanden“ bis zu tiefsten Einblicken in die Dienstrealisierung reichen. Selbst bei gleicher Nutzfunktionalität kann die Managementfunktionalität, die dem Kunden angeboten wird, extrem variieren.

Eine tiefgreifende Behandlung der *management functionality* kann aus zwei Gründen in dieser Arbeit nicht geleistet werden. Zum einen würde es den Rahmen dieser Arbeit bei weitem sprengen, zum anderen liegt bereits eine sehr gute Arbeit für diesen Bereich vor.

Verweis auf Nerb In [Nerb 01] werden explizit die interorganisationalen Aspekte des Customer Service Managements ausführlich untersucht. In der Arbeit werden generische Modelle vorgestellt, die es ermöglichen Dienstmanagementschnittstellen für konkrete Dienste zu entwickeln. Dabei werden neben statischen Aspekten auch die dynamischen Gegebenheiten mittels Interaktionsmodellen dargestellt. Dadurch ist [Nerb 01] eine optimale Ergänzung dieser Arbeit.

Die Arbeit von Nerb hat (wie diese Arbeit auch) einen ihrer Ausgangspunkte im MNM-Dienstmodell und verwendet zur Modellierung ebenfalls UML. Ein Rückgriff auf [Nerb 01] ist nicht zuletzt deswegen möglich und sinnvoll.

4.4.2.3 Das Service Agreement

An dieser Stelle sind, mit einer Ausnahme, alle für diese Arbeit relevanten Teile der Dienstsicht behandelt worden.

Service Agreement
im Rahmen der
Methodik behandelt Die Darstellung des *service agreement* muss an dieser Stelle noch offen bleiben. Bei einer Koppelung zweier Netz-QoS-Dienste ist das *service agreement* das Dokument, in dem unter anderem auch die für die Koppelung zu erledigenden Teilschritte und deren Aufgabenverteilung zwischen Kunden- und Dienstleisterseite festgeschrieben werden. Weil zu diesem Zeitpunkt der Vorgang der Koppelung an sich noch nicht behandelt ist, wird die Behandlung des *service agreement* auf das nächste Kapitel verschoben.

4.5 ZUSAMMENFASSUNG DER MODELLE

Zu Beginn dieses Kapitels wurde durch allgemeine Betrachtungen zur Dienstorientierung die sog. Black-Box-Sicht auf Netz-QoS-Architekturen informell festgelegt. Black-Box-Sicht

Darauf basierend wurde in zwei großen Schritten ein dienstorientiertes Modell für Netz-QoS-Architekturen erstellt, das aus drei Teilmodellen besteht. Im ersten Schritt wurde das **statische Grundmodell** entwickelt, das im zweiten Schritt durch dynamisch Aspekte ergänzt wurde. Sie manifestieren sich in **Modellen der Lebensabschnittskorridore** und in **Ereignisdiagrammen**. Mit allen drei Teilmodellen für Netz-QoS-Architekturen ist die Black-Box-Sicht umgesetzt und konkretisiert worden. Modell für Netz-QoS-Architekturen

Im dritten Teil des Kapitels wurde aus dem Modell für Netz-QoS-Architekturen ein Modell für Netz-QoS-Dienste entwickelt. Dies geschah im ersten Schritt durch Aufzeigen der Anpassungsmöglichkeiten der Modelle von Netz-QoS-Architekturen an die spezifischen Gegebenheiten in einem konkreten Dienst. Der zweite Schritt führte dann eine Einbettung des angepassten Architektur-Modells in das MNM-Dienstmodell durch. Daraus resultierte dann die Modellierung für Netz-QoS-Dienste. Modell für Netz-QoS-Dienste

4.5.1 Zwischenabgleich mit Anforderungen

Zum Ende des ersten großen Teils der Lösung der Koppelungsproblematik wird jetzt ein erster (und noch unvollständiger) Abgleich mit den in [Sektion 2.4](#) aufgestellten Anforderungen durchgeführt.

Der Vergleich der Modelle mit den allgemeinen Anforderungen aus [Abschnitt 2.4.1](#) ergibt:

- Anforderung: Nachhaltige und allgemeine Anwendbarkeit

Sowohl die Modellierung von Netz-QoS-Architekturen als auch das Modell von Netz-QoS-Diensten werden konsequent in einem Verfahren abgeleitet, das nur von allgemeinen Eigenschaften von Netz-QoS-Diensten ausgeht. Bei der Entwicklung der Modelle werden keinerlei Einschränkungen gemacht, die über die Dienstorientierung hinausgehen. Die Anforderung wird für den Bereich der Modelle voll erfüllt. Anforderung:
Nachhaltige und
allgemeine
Anwendbarkeit

- Anforderung: Produktorientierung

Der Anforderung der Produktorientierung wird in der Modellierung explizit Rechnung getragen. Vor der Dienstmodellierung ist eine Anpassung an Spezifika eines bestimmten Dienstes vorgesehen ([Abschn. 4.4.1](#)). Die Anforderung wird für den Bereich der Modelle voll erfüllt. Anforderung:
Produktorientierung

■ Anforderung: szenariospezifische Ergebnisorientierung

Anforderung:
szenariospezifische
Ergebnisorientierung

Die Spezifika eines konkreten Szenarios für eine Koppelung haben bei der Entwicklung der Modelle noch keine Rolle gespielt. Andererseits sind aber keine Einschränkungen gemacht worden, die eine Anwendung der Modellierung auf bestimmte Szenarios beschränken würde. Die Anforderung ist zwar noch nicht erfüllt, durch die Modelle entsteht aber auch kein Hinderungsgrund.

■ Anforderung: Realisierbarkeit

Anforderung:
Realisierbarkeit

Eine Aussage über die Realisierbarkeit des Lösungsansatzes dieser Arbeit kann anhand der Modellierung alleine noch nicht getroffen werden. Die Anforderung ist zwar noch nicht erfüllt, durch die Modelle entsteht aber auch kein Hinderungsgrund.

Nachfolgend erfolgt die Überprüfung der Anforderungen aus Abschnitt 2.4.3. Diese Anforderungen werden speziell an die Modellierung gestellt:

■ Anforderung: Normalisierung

Anforderung:
Normalisierung

Alle für die in der Methodik zu verwendenden Modelle von konkreten Netz-QoS-Architekturen und konkreten Netz-QoS-Diensten gehen auf das gleiche, in diesem Kapitel vorgestellte (Meta-) Modell zurück. Durch die Verwendung von Stereotypen sind die Zusammenhänge zum (Meta-) Modell unmittelbar nachvollziehbar. Damit stellt die Modellierung die geforderte Normalisierung von konkreten Netz-QoS-Architekturen und -Diensten dar. Die Anforderung ist voll erfüllt.

■ Anforderung: Dienstorientierung

Anforderung:
Dienstorientierung

Mit der Entwicklung der Black-Box-Sicht in Sektion 4.2 wird vor Beginn bereits die Grundlage zur Dienstorientierung in der Modellierung gelegt. Da die Modelle die Black-Box-Sicht strikt umsetzen, ist die Anforderung voll erfüllt.

■ Anforderung: Trennung von Dienst und Realisierung

Anforderung:
Trennung von Dienst
und Realisierung

Die strikt eingehaltene Black-Box-Sicht fokussiert nur Dienstbestandteile; realisierungsspezifische Eigenschaften werden ausgeblendet. Zusammen mit dem MNM-Dienstmodell als Ausgangspunkt für die Modellierung von Netz-QoS-Diensten wird damit die Anforderung der Trennung von Dienst und Realisierung voll erfüllt.

■ Anforderung: Management integraler Dienstbestandteil

Anforderung:
Management
integraler
Dienstbestandteil

Mit dem MNM-Dienstmodell als Basis für die Modellierung von Netz-QoS-Diensten wird das Management eines Dienstes automatisch als integraler Bestandteil eines Dienstes erfasst. Die Anforderung ist voll erfüllt.

4.5 ZUSAMMENFASSUNG DER MODELLE

■ Anforderung: Abdeckung des gesamten Lebenszyklus

Mit den Modellen der Lebensabschnittskorridore wird nicht nur eine für die Methodik geeignete Gruppierung der Einzelereignisse einer Netz-QoS-Architektur vorgenommen. Durch die Einbettung der Lebensabschnittskorridore in den Lebenszyklus wird auch die Abdeckung des gesamten Lebenszyklus erreicht. Die Anforderung ist voll erfüllt.

Anforderung:
Abdeckung des
gesamten
Lebenszyklus

Tabelle 4.3 fasst den eben durchgeführten Abgleich mit den Anforderungen nochmals zusammen.

Allgemeine Anforderungen, nur Modelle	
Anforderung	Bewertung
Nachhaltig und allgemein anwendbar	✓
Produktorientierung	✓
szenariospezifische Ergebnisorientierung	noch ?
Realisierbarkeit	noch ?

Anforderungen an Modellierung	
Anforderung	Bewertung
Normalisierung	✓
Dienstorientierung	✓
Trennung von Dienst und Realisierung	✓
Management integraler Dienstbestandteil	✓
Abdeckung des gesamten Lebenszyklus	✓

✓: erfüllt ? : keine Aussage möglich

Tabelle 4.3: Erfüllung der Anforderungen durch Modelle dieser Arbeit

Zusammenfassend lässt sich feststellen, dass alle Anforderungen, die speziell an die Modellierung gestellt werden, auch erfüllt worden sind. Von den allgemeinen Anforderungen werden jene Anforderungen, die auf die Modelle anwendbar sind, ebenfalls erfüllt.

Anforderungen an
Modellierung erfüllt

Mit der Entwicklung der Modelle wurde die Grundlage für den zweiten Teil der Lösung der Koppelungsproblematik gelegt. Der zweite Teil wird sich mit der eigentlichen Methodik der Koppelung beschäftigen und wird im nächsten Kapitel besprochen.

GENERISCHE KOPPELUNGSMETHODIK FÜR NETZ-QoS-DIENSTE

Das vorangegangene Kapitel hat zwei Modellierungen vorgestellt – eine dienstorientierte Modellierung von Netz-QoS-Architekturen und eine Modellierung von Netz-QoS-Diensten – und hat damit bereits eine Reihe der gestellten Anforderung erfüllt. In diesem Kapitel wird jetzt das eigentliche Ziel dieser Arbeit angegangen: Die Entwicklung einer Methodik zur Lösung der Koppelungsproblematik.

Die Koppelung geschieht in fünf Hauptschritten, die in einem Aktivitätsdiagramm in Anlehnung an UML in Abbildung 5.1 dargestellt sind. In dieser Abbildung und allen im Weiteren folgenden Aktivitätsdiagrammen wird die gleiche Darstellungsvariante angewendet, die bereits in [GHHK 02] benutzt worden ist. Die Eingaben und Ausgaben von Aktivitäten sind darin nur informell als sog. Artefakte dargestellt. Diese Variante wurde gewählt, da die vollständig UML-konforme Darstellung mit Objekten ...

Aktivitätsdiagramme
mit Artefakten

- ... einen signifikanten Aufwand zur korrekten Formalisierung der Artefakte bedeuten würde.
- ... die notwendigen Formalismen den Blick auf den eigentlichen Kern – die Methodik – verstellen würde.
- ... erheblich mehr Platz beanspruchen würde, was die Übersichtlichkeit stark beeinträchtigen würde.

In Sektion 5.1 werden im initialen Schritt aus den beiden zu koppelnden Diensten *A* und *B* zwei Modelle nach dem generischen (Meta-) Modell aus Kapitel 4 erstellt. Dieser Schritt geschieht noch völlig unabhängig vom konkreten Anwendungsfall, für den die Koppelung durchgeführt wird.

Modellierung

Die gewonnenen Modelle der Netz-QoS-Dienste *A* und *B* sind zusammen mit dem konkreten Anwendungsfall für alle weiteren vier Schritte die Eingabeartefakte. Zuerst werden allgemeine Grundeigenschaften und generische Methodikteile der Koppelung erarbeitet (Sekt. 5.2). Diese werden in allen weiteren Schritten benötigt. Der nächste Schritt in Sektion 5.3 erarbeitet erste Teillösungen für die Koppelung durch den Vergleich der statischen Teile der Modelle. Im vorletzten Schritt folgt dann eine Gegenüberstellung von dyna-

Grundeigenschaften

Statische Teile

Dynamische Teile

Vervollständigung mischen Modellteilen (Sekt. 5.4), aus der weitere Teillösungen gewonnen werden. Der letzte Schritt, der in Sektion 5.5 erläutert wird, vervollständigt die Koppelung und fügt die Teillösungen zu einer einzigen Koppelungslösung zusammen.

In der letzten Sektion dieses Kapitels wird die gesamte Methodik nochmals kurz zusammengefasst. Abschließend wird ein Resümee durch einen Vergleich mit den gestellten Anforderungen gezogen.

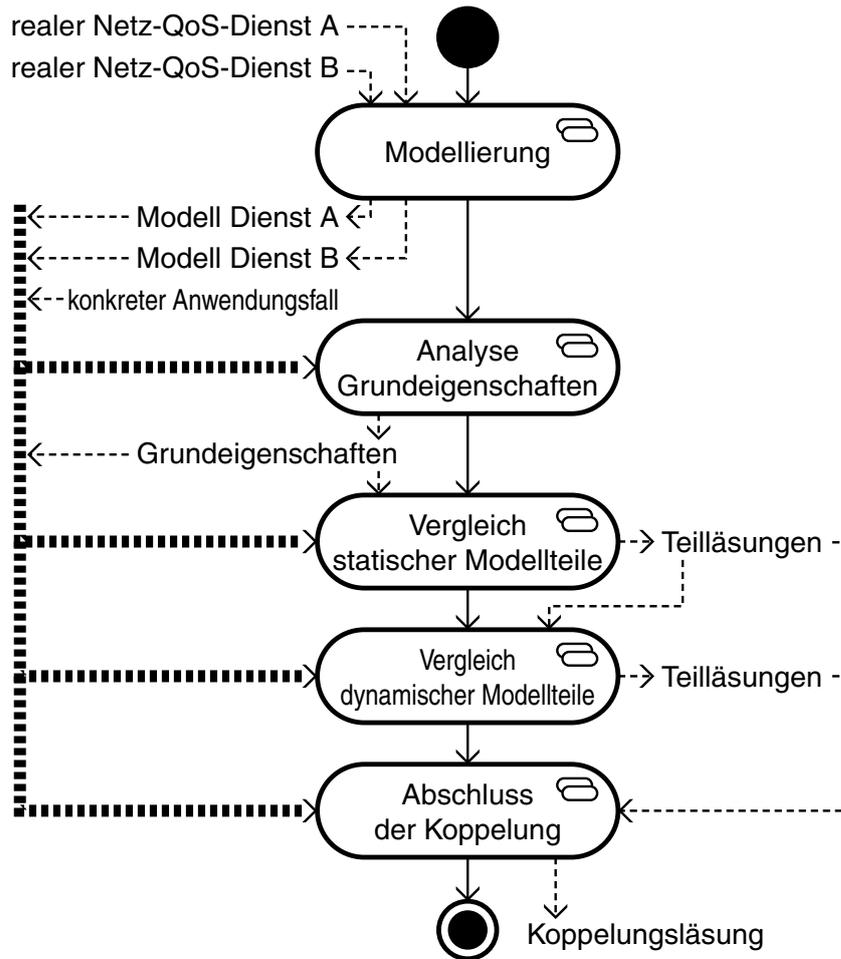


Abbildung 5.1: Aktivitäten zur gesamten Methodik

5.1 METHODIK ZUR MODELLIERUNG

Um die Voraussetzungen für den eigentlichen Vorgang der Koppelung zu schaffen, müssen die zu verbindenden Netz-QoS-Dienste in einer „normierten“ Form dargestellt werden. Deshalb werden im ersten Teil der Methodik die beiden zu koppelnden Netz-QoS-Dienste in Form des Modells aus Kapitel 4 dargestellt.

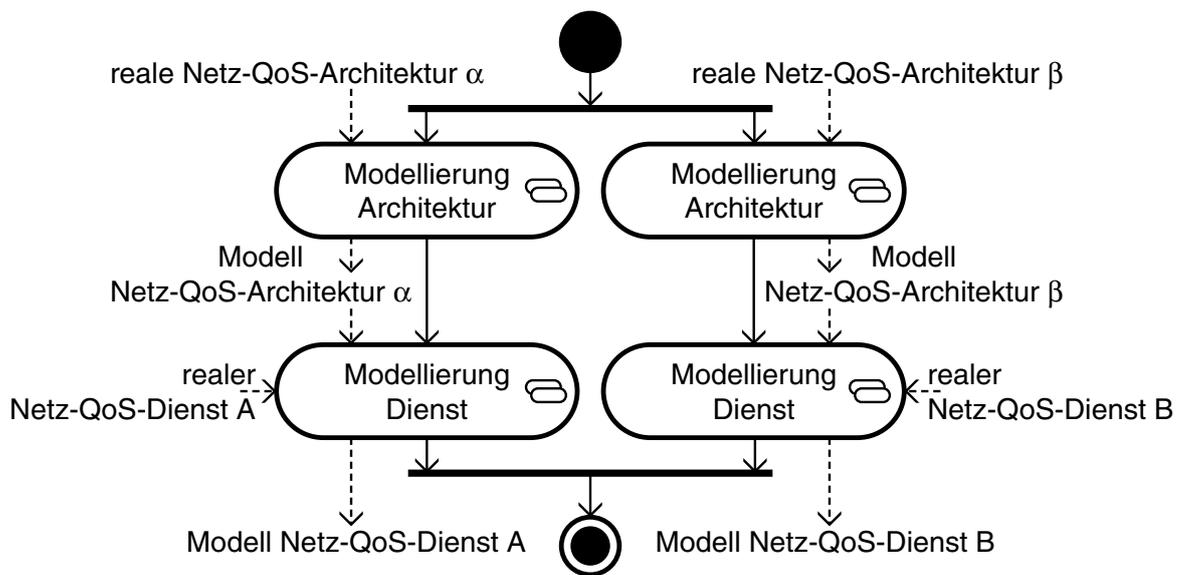


Abbildung 5.2: Aktivitäten zur Modellierung

Der Teilprozess gliedert sich in zwei Schritte und ist in [Abbildung 5.2](#) dargestellt. Zuerst werden die beiden Netz-QoS-Architekturen, auf denen die Dienste aufbauen, modelliert ([Abschn. 5.1.1](#)). Im zweiten Schritt werden diese Modelle zu den Modellen der Netz-QoS-Dienste weiterentwickelt ([Abschn. 5.1.2](#)).

Schritt: Architektur-Modellierung
Schritt: Dienst-Modellierung

Die beiden folgenden Abschnitte stellen die Teilprozesse für jeweils eine Architektur bzw. einen Dienst dar. Wie in [Abbildung 5.2](#) gezeigt, müssen diese Schritte natürlich jeweils für beide zu koppelnden Dienste ausgeführt werden. Dies kann aber parallel geschehen.

Parallele Modellierung beider Dienste

Der Detailgrad der im folgenden präsentierten Schritte ist vergleichsweise niedrig. Außer der Reihenfolge, in der die Modellteile zu erstellen sind, können nur wenige generische Hinweise für die Modellierung gegeben werden. Wie für die meisten Modellierungsaufgaben kann hier nur auf den Einsatz von menschlichem Abstraktions- und Analysevermögen verwiesen werden.

Menschliches Analysevermögen gefragt

5.1.1 Erstellung eines Netz-QoS-Architektur-Modells

Aufgabe diese Teilprozesses ist es, aus dem (Meta-) Modell aus Sektion 4.3 ein konkretes Modell für eine bestimmte Netz-QoS-Architektur zu erstellen (siehe auch Abb. 4.8, S. 84).

Gleiche Struktur wie bei Modellierung Die Schritte zur Instanziierung des allgemeinen Modells gleichen im Wesentlichen der Struktur, die für das allgemeine Modell in Sektion 4.3 besprochen wurde. Die einzelnen Schritte sind als Aktivitätsdiagramm in Abbildung 5.3 dargestellt.

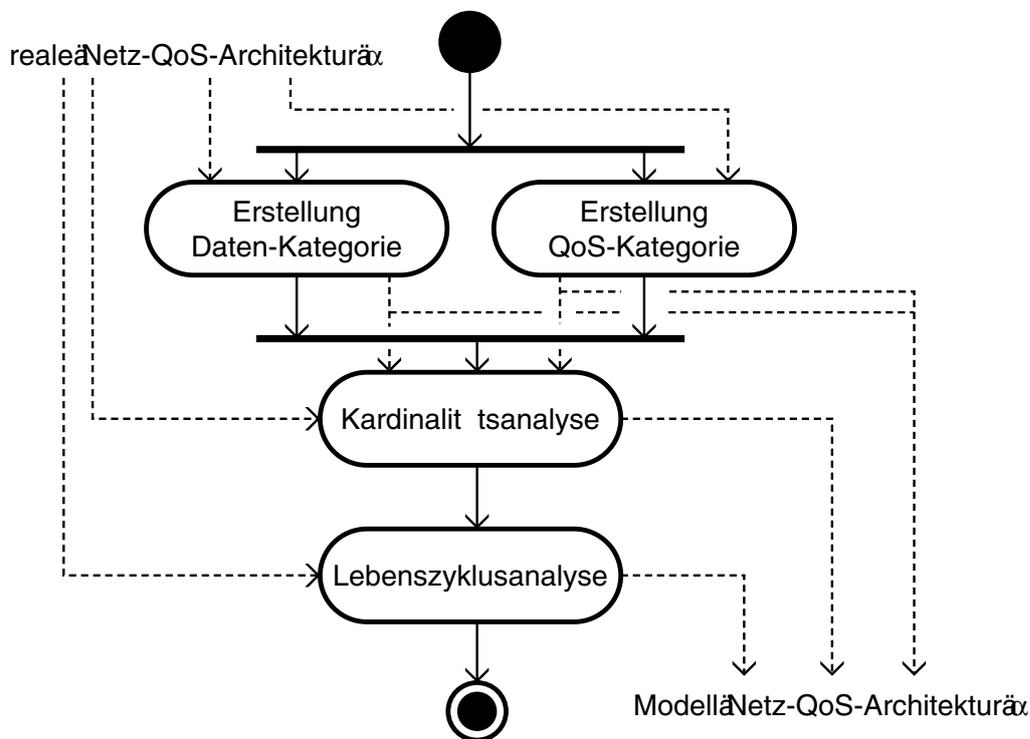


Abbildung 5.3: Aktivitäten zur Architektur-Modellierung

Schritt: Im ersten Teil werden die Bereiche *Daten-Kategorien* und *QoS-Kategorien* entsprechend Abschnitt 4.3.1 erstellt. Zu instanziierten sind beide Klassen selbst sowie alle dazugehörigen Unterklassen. Hierfür werden Standardisierungs- und Spezifikationsdokumente ausgewertet. Die beiden Aktivitäten sollten parallel erfolgen, weil *Daten-Kategorien* und *QoS-Kategorien* in den Dokumenten meist eng miteinander verwoben dargestellt werden.

Schritt: Das sich aus den beiden Schritten ergebende statische Grundmodell wird dann einer Kardinalitätsanalyse nach Abschnitt 4.3.3 unterzogen. In den Fällen von Singleton-Klassen ist besonders darauf zu achten, ob eine Auffaltungstransformation notwendig wird. Diese ist gegebenenfalls nach dem in Unterabschnitt 4.3.3.1 dargestellten Verfahren durchzuführen.

Abschließend werden die dynamischen Aspekte der Architektur-Modellierung abgearbeitet. Zuerst müssen hierzu nach Unterabschnitt 4.3.4.2 die Lebensabschnittskorridore der Grundvorgänge bestimmt werden. Dabei können die Zeitbedingungen aus Tabelle 4.2 benutzt werden, um zu überprüfen, ob die Korridore relativ zueinander und absolut bezüglich des Lebenszyklus korrekt erfasst worden sind. Anschließend werden zu jedem vorhandenen Korridor nach Unterabschnitt 4.3.4.3 die zugehörigen Ereignisse erfasst und modelliert.

Schritt: Lebenszyklusanalyse

5.1.2 Erstellung eines Netz-QoS-Dienst-Modells

Zur Darstellung eines Netz-QoS-Dienstes wird die Modellierung aus Sektion 4.4 verwendet. Die Erstellung eines konkreten Modells teilt sich in die in Abbildung 5.4 dargestellten Schritte ein.

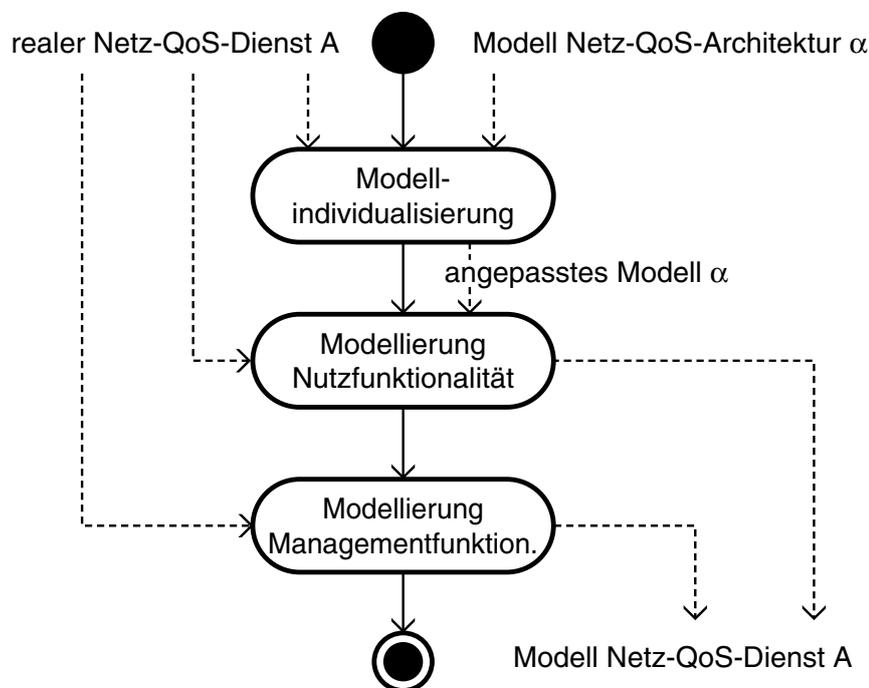


Abbildung 5.4: Aktivitäten zur Dienstmodellierung

Im vorangegangenen Teilprozess wurde bereits die Netz-QoS-Architektur, auf der der Dienst basiert, modelliert. Der erste Schritt in diesem Teilprozess sorgt für eine Anpassung des Architektur-Modells auf die konkreten Ausprägungen im vorliegenden Dienst nach Abschnitt 4.4.1. Die Unterabschnitte 4.4.1.1 und 4.4.1.2 haben bereits exemplarisch gezeigt, welche Modifikationen benötigt werden könnten.

Schritt: Modellindividualisierung

Am Ende des Schritts kann noch eine einfache Überprüfung vorgenommen werden: Eine notwendige Voraussetzung, dass ein Dienst später vollständig spezifiziert sein kann, ist das Fehlen jeglicher Definitionskorridore in den dynamischen Modellen. Wären nämlich noch Definitionskorridore vorhanden,

Keine Definitionskorridore für vollständige Spezifikation

dann würde das bedeuten, dass später weitere Spezifikationen vorgenommen werden müssten.

Umfang der Modifikationen als Maß für Standardkonformität

Der Umfang der Modifikationen ist gleichzeitig auch ein (grobes) Maß dafür, wie exakt ein Dienst einen Architektur-Standard umsetzt:

- Mussten keine Modifikationen vorgenommen werden, dann wird der Standard voll umgesetzt.
- Mussten Teile aus dem Architektur-Modell gestrichen werden, dann wird der Standard nur eingeschränkt umgesetzt.
- Sind Ergänzungen oder andere Modifikationen notwendig gewesen um den Dienst zu modellieren, dann ist der Dienst nicht konform zum Standard der Netz-QoS-Architektur. Dies gilt natürlich nur soweit, wie der Standard die Modifikationen nicht vorsieht (siehe DiffServ).

Schritt: Modellierung Nutzfunktionalität

Mit dem modifizierten Architektur-Modell ist nach Unterabschnitt [4.4.2.1](#) bereits ein Großteil der im nächsten Schritt zu bearbeitenden Nutzfunktionalität modelliert worden. Es verbleibt die Spezifikation des *service access point*. Soweit möglich sollten dabei Standards und offene Spezifikationen zum Einsatz kommen.

Schritt: Modellierung Managementfunktionalität

Der letzte Schritt des Teilprozesses besteht aus der Modellierung der Managementfunktionalität. Wie in Unterabschnitt [4.4.2.2](#) bereits dargelegt, wird hierfür auf die Arbeit [[Nerb 01](#)] von Nerb zurückgegriffen.

Mit dem Abschluss des letzten Schritts im Teilprozess ist auch der gesamte Schritt der Modellierung aus Abbildung [5.1](#) abgeschlossen. Mit der anschließenden Sektion beginnt jetzt der eigentliche Teil der Methodik für den Koppelungsvorgang.

5.2 BASISVERGLEICH UND GRUNDLAGEN

Nachdem mit der vorangegangenen Sektion die zu verbindenden Dienste als Netz-QoS-Dienst-Modelle dargestellt sind, beginnt jetzt der eigentliche Vorgang der Koppelung. Im Verlauf dieser Sektion werden zunächst einige Grundlagen für die Koppelungsmethodik gelegt.

Der erste Abschnitt beschäftigt sich mit der Bestimmung allgemeiner Eigenschaften einer Koppelung. Sie lassen sich rein aus dem konkreten Anwendungsfall und ohne Detailvergleich der Netz-QoS-Dienste ableiten.

Grundeigenschaften

In allen weiteren Abschnitten der Sektion werden generische Bausteine entwickelt, die im Laufe der Methodik für konkrete Teilproblemstellungen angewendet werden:

Ein Abschnitt stellt Aktivitätsmuster vor, die mehrfach verwendbar sind (Abschn. 5.2.3).

generische Aktivitätsmuster

Weiterhin wird eine universelle Notationsstruktur eingeführt (Abschn. 5.2.2). Mit ihr werden die unterschiedlichen Fälle der Teilproblemstellungen sowie die zugehörigen Lösungsmöglichkeiten systematisch aufgeschrieben. Um Teilfälle und Lösungsvarianten eindeutig referenzieren zu können, wird auch ein Namensschema eingeführt.

Notationsstruktur

Namensschema

Abschließend wird noch die Koppelungsproblematik zwischen zustandsbehafteten und zustandslosen Entitäten abstrakt behandelt (Abschn. 5.2.4). Die abstrakten Fälle werden im Laufe der Methodik an mehreren Stellen konkretisiert.

Abstrakte Koppelung
zustandslos /
zustandsbehaftet

5.2.1 Grundeigenschaften des Anwendungsfalls

Wie bereits in Abschnitt 2.3.2 festgestellt, lassen sich zwei **Grundanwendungsfälle** für eine Koppelung unterscheiden (Abb. 5.5):

Zwei Grundanwendungsfälle

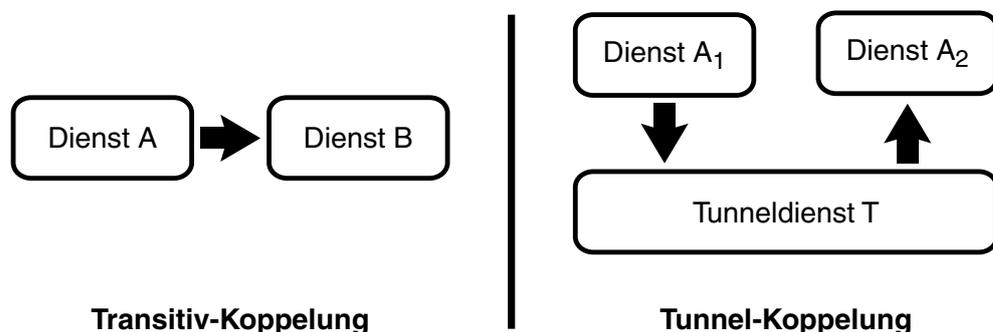


Abbildung 5.5: Grundanwendungsfälle

■ **Grundanwendungsfall Transitiv-Koppelung**

Anwendungsfall
Transitiv-Koppelung

Bei einer Transitiv-Koppelung erfolgt die Kommunikation zwischen den beiden beteiligten Netz-QoS-Diensten A und B ohne Kenntnis und Beachtung anderer Netz-QoS-Dienste (linker Teil in Abb. 5.5). Das Szenario aus Abschnitt 2.1.2 gibt ein Beispiel. Ziel der Koppelung muss es sein, den Übergang soweit wie möglich verlustlos zu gestalten.

Für eine Transitiv-Koppelung von Dienst A mit Dienst B wird die Kurzschreibweise

$$A \rightarrow B$$

verwendet.

■ **Grundanwendungsfall Tunnel-Koppelung**

Anwendungsfall
Tunnel-Koppelung

Eine Tunnel-Koppelung liegt dann vor, wenn zwei gleiche Netz-QoS-Dienste A_1 und A_2 über einen dazwischenliegenden **Tunneldienst** T verbunden werden sollen (rechter Teil in Abb. 5.5), wie z.B. im Szenario von Abschnitt 2.1.1. Die Tunnel-Koppelung ist dabei mehr als nur die Kombination zweier Transitiv-Koppelungen, weil bekannt ist, dass es sich bei A_1 und A_2 um die gleichen Dienste handelt, die beide auf den gleichen Netz-QoS-Architekturen beruhen. Diese zusätzliche Information wird im Laufe der Methodik noch vielfach zur Problemlösung benutzt werden.

Für eine Tunnel-Koppelung von Dienst A_1 mit Dienst A_2 via den Tunneldienst T wird die Kurzschreibweise

$$A \curvearrowright B$$

verwendet.

Koppelungssymmetrie In beiden Fällen muss noch die Richtung unterschieden werden, in der eine Kommunikation initiiert wird. Dies ist die sog. **Koppelungssymmetrie**:

■ **Asymmetrische Koppelung**

Asymmetrische
Koppelung

Hier wird die Kommunikation immer nur von einer Seite begonnen. Für eine Tunnel-Koppelung bedeutet das, dass immer entweder Dienst A_1 oder A_2 die Kommunikation initiiert. Im Fall der Transitiv-Koppelung ist es entsprechend immer Dienst A oder B . Der Dienst, der die Kommunikation initiiert, hat die Rolle **Quelldienst** inne. Die Rolle des anderen Dienstes heißt dann entsprechend **Zieldienst**.

■ **Symmetrische Koppelung**

Symmetrische
Koppelung

Eine Koppelung wird als symmetrisch bezeichnet, wenn die Kommunikation von beiden Seiten begonnen werden kann. Wie im asymmetrischen Fall kann die Unterscheidung analog für Tunnel- und Transitiv-Koppelung gemacht werden. Bei einer symmetrischen Koppelung können die Rollen Quell- und Zieldienst nicht unterschieden werden.

Im späteren Verlauf der Methodik wird die symmetrische Koppelung auf die asymmetrische Koppelung zurückgeführt. Dies geschieht, indem

man den symmetrischen Fall als zwei asymmetrische Fälle begreift, wobei jeweils die Rollen von Quell- und Zieldienst vertauscht sind.

Es sei bemerkt, dass auch bei einer Transitiv-Koppelung die Unterscheidung der Symmetrieeigenschaft sinnvoll ist. Häufig wird hier nur eine symmetrische Koppelung betrachtet. Bei Anwendungen wie z.B. Video-On-Demand, bei denen die QoS-relevante Kommunikation nur von einer Seite (im Beispiel vom Video-Provider) initiiert wird, liegt aber nur ein asymmetrischer Fall vor. Asymmetrisch auch bei Tunnel-Koppelung sinnvoll

Ebenfalls zu beachten ist, dass die Koppelungssymmetrie nicht die **Kommunikationsrichtung** festlegt. Bei einem Dienst wird diese nur durch die Klasse *Kommunikationsrelation* eines Netz-QoS-Dienstes (Abb. 4.18, S. 104) festgelegt: Kommunikationsrichtung

■ **Unidirektionale Kommunikationsrichtung**

Liegt vor bei *Daten-Kategorien* die mit *Unicast unidirektional* oder *Multicast 1:n* assoziiert sind. Unidirektionale Kommunikation

■ **Bidirektionale Kommunikationsrichtung**

Liegt vor bei *Daten-Kategorien*, die mit *Unicast bidirektional* oder *Multicast m:n* assoziiert sind. Bidirektionale Kommunikation

Die insgesamt erzielbare Kommunikationsrichtung im gekoppelten System wird in einem späteren Schritt der Methodik bestimmt.

Die Eigenschaften Grundanwendungsfall und Koppelungssymmetrie werden für die restliche Methodik durchgehend benötigt. Entsprechend sind deren Bestimmung die nächsten Schritte in der Methodik (Abb. 5.6).

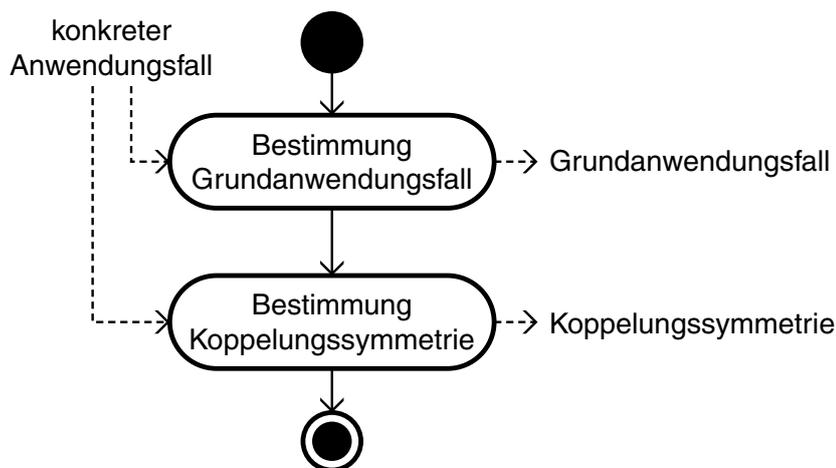


Abbildung 5.6: Aktivitäten zur Feststellung grundlegender Koppelungseigenschaften

5.2.2 Namensschema und Notationsstruktur

Durch verschiedene Anwendungsfälle, Protokollvarianten, Dienstausrüstungen etc. müssen bei der Koppelung von Netz-QoS-Diensten eine große Anzahl von Detailfällen behandelt werden. Zu den einzelnen Detailfällen gibt es weiterhin häufig nicht nur eine, sondern mehrere Lösungsmöglichkeiten. In der Methodik müssen alle diese Detailfälle und ihre Lösungsmöglichkeiten unterschieden werden. Zu diesem Zweck wird jetzt ein Namensschema und eine Notationsstruktur festgelegt.

Große Fallanzahl
macht Systematik
nötig

Im weiteren Verlauf wird in der Methodik eine einheitliche Notation für Teilprobleme bzw. Teilfälle und deren Lösungsmöglichkeiten verwendet. Zur eindeutigen Referenzierung wird ein eigenes Namensschema benutzt. Das Schema gehorcht folgender EBNF¹-Syntax:

```

    <fall> ::= KM- <bereich> - <einzelfall>
    <bereich> ::= g? <part>+
    <einzelfall> ::= <part>+

    <loesung> ::= L <loesPart> ( - <loesPart> )* -( <fall> )
    <loesPart> ::= <part>* <zahl>

    <part> ::= <GBuchst> ( <kBuchst> )* <zahl>?
    <zahl> ::= <ziffer> ( 0 | <ziffer> )*
    <ziffer> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
    <GBuchst> ::= A | B | C | D | E | F | G | H | I | J | K
    | L | M | N | O | P | Q | R | S | T | U
    | V | W | X | Y | Z
    <kBuchst> ::= a | b | c | d | e | f | g | h | i | j | k
    | l | m | n | o | p | q | r | s | t | u
    | v | w | x | y | z

```

Fallbezeichner Der Bezeichner eines Teilfalls <fall> besteht aus einem vorangestelltem „KM“ (für Koppelmuster), einem Bezeichner für den Teilbereich <bereich>, zu dem der Fall gehört, und einem individuellen Bezeichner für den Teilfall <einzelfall>. Die Teile <bereich> und <einzelfall> bestehen jeweils aus mindestens einer Teilkomponente <part>. Ein <part> beginnt immer mit einem großen Buchstaben. Einem <bereich> kann ein kleines „g“ vorangestellt sein, es markiert einen generischen Fall.

Lösungsbezeichner Der Bezeichner einer Lösungsmöglichkeit besteht aus einem vorangestelltem „L“, gefolgt von einer beliebigen Menge von Teilbezeichnern <loesPart>. Ein solcher Teilbezeichner enthält mindestens eine fortlaufende Nummer <zahl>. Nach den Teilbezeichnern folgt in Klammern angehängt der Fallbezeichner <fall> des Teilfalls, zu dem die Lösung gehört.

Das Namensschema und die Notation werden am folgenden Beispiel illustriert:

¹Extended Backus-Naur Form (EBNF)

◆ **KM-bereich-einzelfall:** Teilfall eines Koppelmusters.

Fall KM-bereich-einzelfall

Zu Beginn wird die durch den Teilfall aufgeworfene Problemstellung beschrieben.

★ **L1-(KM-bereich-einzelfall):** Erste Lösungsmöglichkeit zum Teilfall [KM-bereich-einzelfall](#).

Lösung L1-(KM-bereich-einzelfall)

Zu Beginn kann eine allgemeine Beschreibung der Lösungsmöglichkeit stehen.

- ▶ **Voraussetzung:** Voraussetzungen, die erfüllt sein müssen, bevor die Lösung angewendet werden kann.
- ▶ **Lösungskomponenten:** Teilkomponenten, die für die Lösung benötigt werden.
- ▶ **Lösungsworkflow:** Teilworkflows, die für die Realisierung der Lösung auszuführen sind.
- ▶ **Problempotential:** Mögliche Probleme, die durch diese Lösung verursacht werden.
- ▶ **Analoge Lösung:** [L1-\(KM-bereich-einzelfall\)](#). Ist eine Lösungsmöglichkeit analog zu einer anderen, dann wird auf diese direkt verwiesen.
- ▶ **Zusätzliche Aktivitäten:** Bei einem Verweis auf eine analoge Lösung werden hier zusätzliche Aktionen notiert.

Im weiteren Verlauf der Methodik werden konkret die folgenden Komponenten in Bereichs- und Einzelfall-Bezeichnern verwendet werden (Tab. 5.1):

Bereichskomponente	Einzelfallkomponente
g generisch	A Asymmetrisch
Csm Customer Service Manag.	Bd Bidirektional
Dk Daten-Kategorie	M Multicast
Evt Event, Ereignis	Semmod Semantik-Modifikation
G Garantietyp	Sig Signalisierung
I Instanz	S Symmetrisch
Kr Kommunikationsrelation	Trans Transformation
Qk QoS-Kategorie	Tr Transitiv-Koppelung
Qp QoS-Parameter	Tu Tunnel-Koppelung
S Semantik	Typmod Typ-Modifikation
Up Ungepaart	Ud Unidirektional
Usap Usage Service Access Point	U Unicast
Vo Verbindungsorientiert	Vgl Vergleich
W Wertemenge	
Z Zustand	

Tabelle 5.1: Komponenten von Bereichs- und Einzelfallbezeichnern

Am Ende der Arbeit befinde sich getrennte Indexe für Teilfälle (S. 257) und Teillösungen (S. 259).

Explizite Nennung erfüllt Anforderung Durch die explizit getrennte Nennung von Teilworkflows und Teilkomponenten einer Lösungsmöglichkeit wird über die gesamte Methodik hinweg der Anforderung nach Spezifikation von Komponenten und Workflows (Abschn. 2.4.4) Rechnung getragen.

Querverweise im PDF aktiv verlinkt Zur Unterstützung einer interaktiven Anwendung der Methodik ist vom Autor eine elektronische Fassung dieser Arbeit erhältlich. Es handelt sich um ein Dokument im PDF-Format. Darin sind alle Querverweise als Hyperlinks angelegt, insbesondere auch solche auf Fallbezeichnungen und Lösungsmöglichkeiten. In einer „real world“-Anwendung der Methodik können dann alle Querverweise auf einfache Art unmittelbar und schnell nachvollzogen werden.

5.2.3 Generische Aktivitätsmuster der Koppelung

Dieser Abschnitt stellt zwei allgemeine Muster von Aktivitäten vor, die im Rahmen der Methodik mehrfach Verwendung finden.

5.2.3.1 Aktivitätsmuster zur Ermittlung optimaler Paarungen

Mehrfachausprägung gleicher Meta-Modell-Klassen Stellt man zwei Netz-QoS-Dienste gegenüber, kann es vorkommen, dass in jedem Dienst mehrere Ausprägungen des gleichen Modellteils aus dem Meta-Modell vorhanden sind.

Sind beispielsweise zwei DiffServ-basierte Dienste zu koppeln, gibt es in jedem Dienst jeweils zwei Spezialisierungen der Klasse *Daten-Kategorie*.

Optimale Paarungen sind zu ermitteln In einem solchen Fall ist es die Aufgabe der Methodik, die optimalen Koppelpaarungen zu ermitteln. Um gewährleisten zu können, dass wirklich die optimale Kombination gefunden wird, müssen im Allgemeinen zuerst alle möglichen Paarungen untersucht werden. Dies kann mit dem in Abbildung 5.7 dargestellten allgemeinen, iterativen Aktivitätsmuster geschehen:

Auswahl eines Paares ■ Eine Iteration beginnt mit der Auswahl eines Paares. Aus den Diensten *A* und *B* bzw. *T* wird jeweils ein Teilnehmer ausgewählt.

Analyse des Paares ■ Der zweite Iterationsschritt besteht aus der Untersuchung der Koppelung dieses einen Paares. Das Ergebnis der Aktion sind Lösungen zur Durchführung der Koppelung.

Iterationsschleife ■ Die beiden vorhergehenden Schritte werden wiederholt, bis alle Paarkombinationen untersucht worden sind.

Auswählen der besten Paarung ■ Nachdem alle Paarkombinationen untersucht sind, liegen Lösungen für alle Kombinationen vor. Im letzten Schritt müssen nun die tatsächlich für die Koppelung zu verwendenden Paarungen ausgewählt werden. Dies geschieht u.a. durch Bewertung der Aufwändigkeit der Lösungsschritte und deren Erfolg hinsichtlich Erlangung einer vollständi-

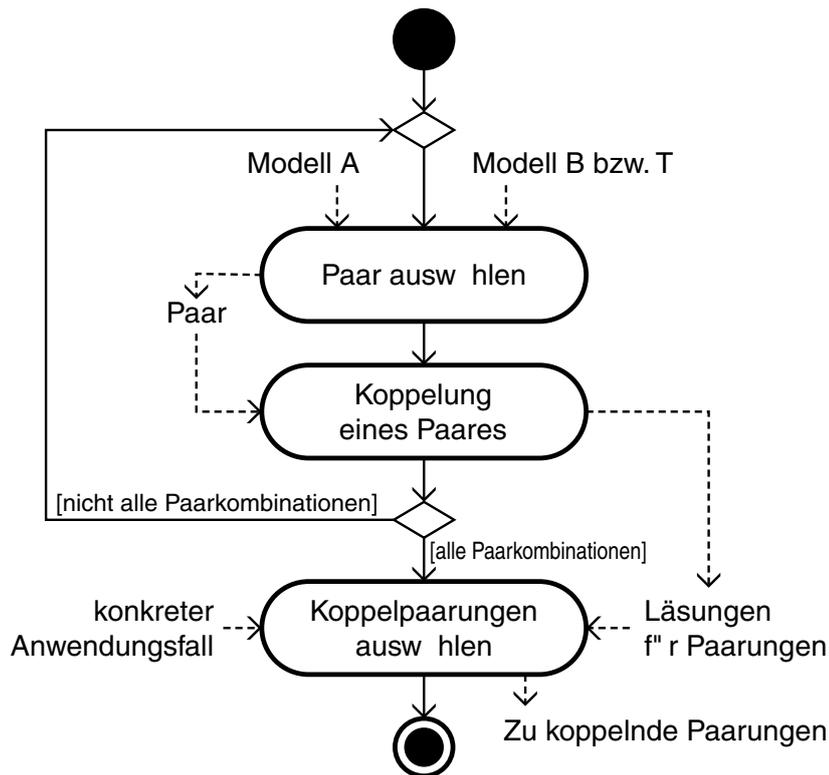


Abbildung 5.7: Allgemeines Aktivitätsmuster für Ermittlung optimaler Paarungen

gen Funktionalität des Gesamtsystems. Dabei müssen alle Bewertungen immer im Kontext des konkreten Anwendungsfalls vorgenommen werden: Bei zwei verschiedenen Anwendungsfällen, bei denen jeweils die gleichen Dienste zum Einsatz kommen, können die beiden Koppelungen durchaus unterschiedlich sein. Der Grund hierfür liegt dann darin, dass Lösungen, die für den einen Anwendungsfall gut geeignet sind, im anderen Fall schlechter bewertet werden müssen.

Die Gegenüberstellung von *Daten-Kategorien* in Abschnitt 5.3.1 gibt ein ausführliches Beispiel der Anwendung dieses Musters an.

5.2.3.2 Aktivitätsmuster für Koppelungssymmetrie

Bei Einführung der Unterscheidung der Koppelungssymmetrie wurde bereits angekündigt, dass der symmetrische Fall auf den asymmetrischen zurückgeführt wird. Teilprozesse der Methodik, in denen dies geschieht, sind nach einem strukturell identischen Muster gestaltet. Dieses Muster ist in Abbildung 5.8 dargestellt und wird im Folgenden erklärt.

Rückführung symmetrischer Fall auf asymmetrischen Fall

Zu Beginn erfolgt eine Verzweigung nach dem Grundanwendungsfall.

- Liegt eine Transitiv-Koppelung zwischen den Netz-QoS-Diensten A und B vor, dann wird zuerst immer der asymmetrische Koppelfall $A \rightarrow B$ untersucht. Unter Einbeziehung des konkreten Anwendungsfalls wird eine Teillösung ausgewählt. Handelt es sich tatsächlich um eine asym-

Asymmetrisch $A \rightarrow B$

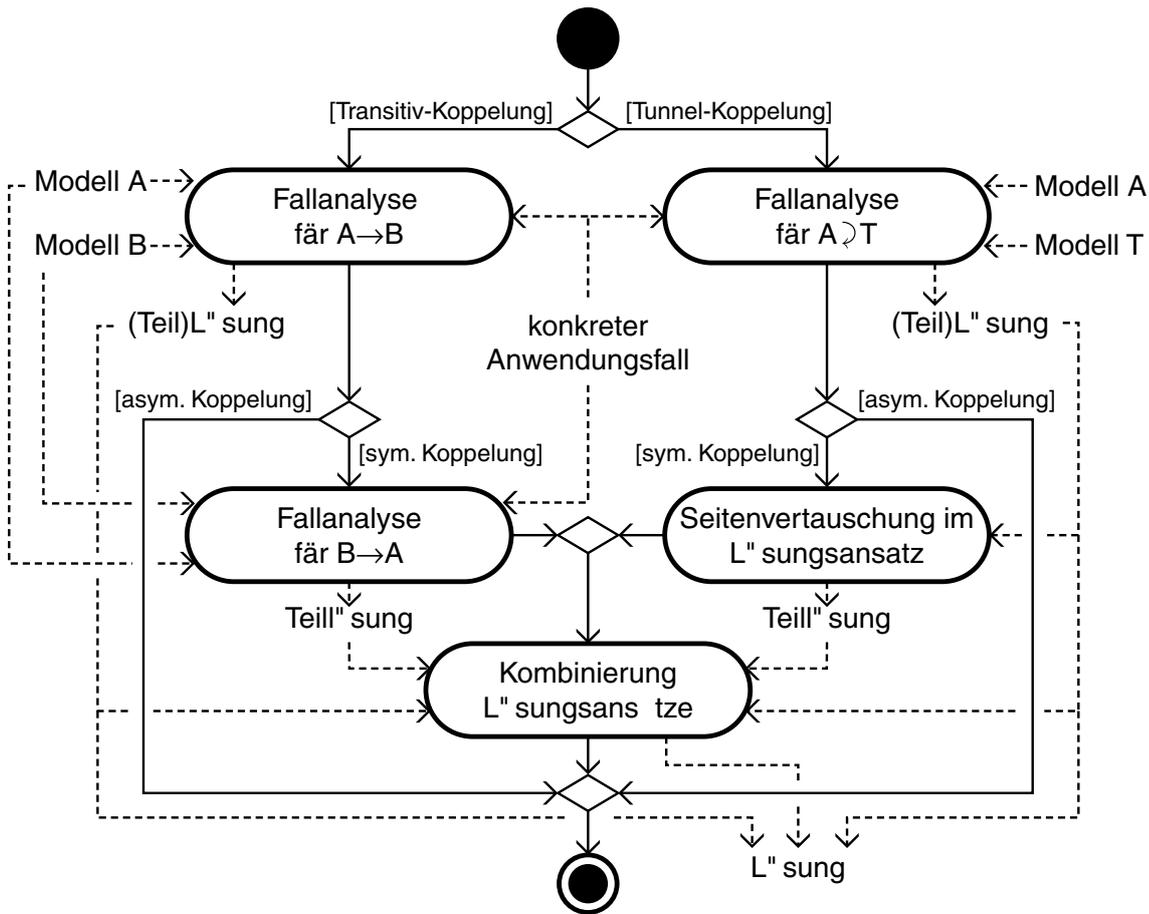


Abbildung 5.8: Allgemeines Aktivitätsmuster für Koppelungssymmetrien

metrische Koppelung, dann ist diese Teillösung auch gleich die Lösung des Teilprozesses.

Rückrichtung für symmetrischen Fall

Im symmetrischen Fall ist bis jetzt nur eine der beiden Teilrichtungen bearbeitet worden. Deshalb wird im nächsten Schritt die gleiche Aktion wie vorher ausgeführt, allerdings in der Rückrichtung. D.h. jetzt wird der asymmetrische Fall $B \rightarrow A$ untersucht, was zu einer weiteren Teillösung führt.

Asymmetrisch $A \rightarrow T$

■ Bei einer Tunnel-Koppelung wird ebenfalls zuerst der asymmetrische Fall analysiert, also von A_1 nach A_2 via T . Die unter Beachtung des konkreten Anwendungsfalls gewonnene Teillösung ist wieder gleichzeitig die komplette Lösung für eine asymmetrische Koppelung.

Gleiche Lösung für Rückrichtung

Für eine symmetrische Koppelung muss bei einer Tunnel-Koppelung aber jetzt nicht der vorangegangene Schritt wiederholt werden, denn schließlich befindet sich an beiden Tunnelenden der gleiche Dienst A . Für die Rückrichtung muss jetzt nur die bereits gefundene Lösung mit vertauschten Rollen von A_1 und A_2 neu interpretiert werden. Hieraus erhält man eine weitere Teillösung.

Zum Schluss müssen die gefundenen Teillösungen noch miteinander kombiniert werden. Handelt es sich um eine asymmetrische Koppelung, dann ist weiter nichts zu tun, da nur genau eine Teillösung erarbeitet wurde. Bei den symmetrischen Koppelungen müssen die beiden gefundenen Teillösungen zu einer gemeinsamen Lösung verarbeitet werden. Bei der Kombination zweier Teillösungen muss neben der Zusammenlegung von Lösungskomponenten und Lösungsworkflows auch analysiert werden, inwieweit Einschränkungen einer Teillösung sich auf die kombinierte Lösung insgesamt durchschlagen.

Kombination der Teillösungen

Das erste Beispiel für die Anwendung des eben vorgestellten allgemeinen Prozesses gibt das Attribut *Verbindungsorientiert* in Unterabschnitt [5.3.1.1](#).

5.2.4 Grundproblem zustandsloser und zustandsbehafteter Entitäten

Ein im Laufe der Methodik immer wiederkehrendes Problem ist der Übergang zwischen zustandslosen und zustandsbehafteten Entitäten. Deshalb werden in diesem Abschnitt alle auftretenden Fälle zuerst allgemein behandelt. Dabei wird jedem Fall eine Fallbezeichnung zugewiesen, die dann die aus dem Fall resultierende Problemstellung erörtert. Anschließend werden mögliche Lösungsmuster und deren Einschränkungen angegeben.

Übergänge
zustandslos /
zustandsbehaftet

In der Arbeit wird später im konkreten Fall über die Fallbezeichnung wieder Bezug auf diese allgemeinen Muster genommen. Um die im vorangegangenen Abschnitt identifizierten grundsätzlichen Anwendungsfälle in der Untersuchung von zustandslosen / -behafteten Modellteilen schnell nachvollziehen zu können, werden diese Anwendungsfälle im Folgenden explizit behandelt.

Allgemeine Muster
später
wiederverwendet

5.2.4.1 Zustandslos / -behaftet bei Transitiv-Koppelung

In diesem Unterabschnitt werden nacheinander alle Fallkombinationen des Übergangs zwischen zustandslosen und zustandsbehafteten Entitäten für eine Transitiv-Koppelung untersucht.

Übergänge bei
Transitiv-Koppelung

◆ **KM-gZ-Tr1:** Zustandslos → zustandslos.

Fall KM-gZ-Tr1

Sind beide in Beziehung stehende Entitäten zustandslos, dann treten keine Probleme auf. Keine der beiden Entitäten hält Zustandsinformationen, entsprechend sind keine zu übergeben.

◆ **KM-gZ-Tr2:** Zustandslos → zustandsbehaftet.

Fall KM-gZ-Tr2

Beim Übergang von einer zustandslosen zu einer zustandsbehafteten Entität entsteht das Problem, dass die abhängige Zielentität vor dem Übergangsbeginn in einen definierten Ausgangszustand gebracht werden muss, dessen Semantik kompatibel zur zustandslosen Quellentität

ist. Ebenso muss nach Ende des Übergangs die Zielentität wiederum in einen passenden Zustand versetzt werden. Hierzu gibt es zwei allgemeine Lösungsansätze:

Lösung
L1-(KM-gZ-Tr2)

- ★ **L1-(KM-gZ-Tr2):** Globale, statische Zielentität.
 - ▶ **Voraussetzung:** Zielentität muss a priori bestimmbar sein und muss sich vor, während und nach dem Übergang im gleichen Zustand befinden dürfen.
 - ▶ **Lösungskomponenten:** Keine.
 - ▶ **Lösungsworkflow:** Manuelle Festlegung der Zielentität und ihres globalen Zustands.

Lösung
L2-(KM-gZ-Tr2)

- ★ **L2-(KM-gZ-Tr2):** Dynamische Anpassung der Zielentität.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungskomponenten:** Wächterkomponente, die kontrolliert, wann der Entitätenübergang beginnt bzw. endet und dynamisch den Zustand der Zielentität zu Beginn und am Ende festlegt.
 - ▶ **Lösungsworkflow:** Erstellen und Installieren der Wächterkomponente.
 - ▶ **Problempotential:** Erkennen des Endes der Übergangsphase zwischen Entitäten. Das Übergangsende kann nicht anhand eines Zustandes der Quellentität erkannt werden.

Fall KM-gZ-Tr3

- ◆ **KM-gZ-Tr3:** Zustandsbehaftet → zustandslos.

Erfolgt ein Übergang von einer zustandsbehafteten Entität auf eine zustandslose, dann treten keine Probleme auf, da die abhängige Zielentität keine Zustandsinformationen benötigt.

Fall KM-gZ-Tr4

- ◆ **KM-gZ-Tr4:** Zustandsbehaftet → zustandsbehaftet.

Liegt ein Übergang vor, bei der Quell- und Zielentität zustandsbehaftet sind, dann müssen Ausgangszustand und jeder Zustandsübergang von der Quell- auf die Zielentität abgebildet werden.

Lösung
L1-(KM-gZ-Tr4)

- ★ **L1-(KM-gZ-Tr4):** Dynamische Umsetzungskomponente.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungskomponenten:** Umsetzungskomponente, die dynamisch den Initialzustand und die Zustandsübergänge von der Quellentität auf die Zielentität überträgt.
 - ▶ **Lösungsworkflow:** Erstellen und Installieren der Umsetzungskomponente.
 - ▶ **Problempotential:** Möglicherweise lassen sich nicht alle Zustände bzw. Zustandsübergänge der Quellentität auf die Zielentität abbilden.

5.2.4.2 Zustandslos / -behaftet bei Tunnel-Koppelung

Nachdem im vorangegangenen Unterabschnitt die Fallkombinationen für die Transitiv-Koppelung untersucht wurden, widmet sich dieser Unterabschnitt dem Übergang zwischen zustandslosen und zustandsbehafteten Entitäten im Fall einer Tunnel-Koppelung. Übergänge bei Tunnel-Koppelung

◆ **KM-gZ-Tu1:** Zustandslos \curvearrowright zustandslos. Fall KM-gZ-Tu1

Analog zu [KM-gZ-Tr1](#) tritt hier kein Problem auf, da keine der beteiligten Entitäten zustandsbehaftet ist. Eine Zustandsübertragung zwischen Quellentität und Zielentität ist nicht notwendig.

◆ **KM-gZ-Tu2:** Zustandslos \curvearrowright zustandsbehaftet. Fall KM-gZ-Tu2

Am Tunneleingang liegt der Fall [KM-gZ-Tr2](#) vor. Der Tunnelausgang entspricht Fall [KM-gZ-Tr3](#). Da zwischen Quell- und Zielentität keine Zustandsübermittlung stattfinden muss, ist das zu lösende Problem nur analog zu [KM-gZ-Tr2](#), da [KM-gZ-Tr3](#) problemlos ist. Entsprechend die Lösungsmöglichkeiten:

★ **L1-(KM-gZ-Tu2):** Globale, statische Zielentität. Lösung
L1-(KM-gZ-Tu2)

- ▶ **Analoge Lösung:** [L1-\(KM-gZ-Tr2\)](#), zwischen Quell- und Tunnelentität.

★ **L2-(KM-gZ-Tu2):** Dynamisch Anpassung der Zielentität. Lösung
L2-(KM-gZ-Tu2)

- ▶ **Analoge Lösung:** [L2-\(KM-gZ-Tr2\)](#), zwischen Quell- und Tunnelentität.

◆ **KM-gZ-Tu3:** Zustandsbehaftet \curvearrowright zustandslos. Fall KM-gZ-Tu3

Das Problem in diesem Fall besteht in der Zustandsübermittlung zwischen Quell- und Zielentität über eine zustandslose Tunnelentität. Ein Rückgriff auf den Fall [KM-gZ-Tr3](#) am Tunneleingang ist nicht sinnvoll, da dabei die Zustandsinformationen für die Zielentität verloren gehen.

★ **L1-(KM-gZ-Tu3):** Zusätzliche tunnelinterne Zustandssignalisierung. Lösung
L1-(KM-gZ-Tu3)

- ▶ **Voraussetzung:** Tunnelinterne Übertragungsmöglichkeit für Signalisierung.
- ▶ **Lösungskomponenten:** Zustandssender am Tunneleingang, Zustandsempfänger am Tunnelausgang und zusätzlicher Mechanismus (z.B. Protokoll) zur Zustandsübermittlung zwischen Sender und Empfänger. Es wird die Tunnelentität selbst zur Übermittlung benutzt.
- ▶ **Lösungsworkflow:** Erstellen und Installieren von Sender, Empfänger und Übermittlungsmechanismus.
- ▶ **Problempotential:** Wechselwirkung zwischen regulärer Tunnelnutzung und zusätzlicher Signalisierung.

- Lösung L2-(KM-gZ-Tu3) ★ **L2-(KM-gZ-Tu3):** Zusätzliche tunnelexterne Zustandssignalisierung.
- ▶ **Voraussetzung:** Tunnelexterne Übertragungsmöglichkeit für Signalisierung.
 - ▶ **Lösungskomponenten:** Zustandssender am Tunneleingang, Zustandsempfänger am Tunnelausgang und zusätzlicher Mechanismus (z.B. Protokoll) zur Zustandsübermittlung zwischen Sender und Empfänger. Es wird eine zusätzliche Tunnelentität zur Übermittlung benutzt.
 - ▶ **Lösungsworkflow:** Erstellen und Installieren von Sender, Empfänger und Übermittlungsmechanismus und zusätzlicher Tunnelentität.
- Fall KM-gZ-Tu4 ♦ **KM-gZ-Tu4:** Zustandsbehaftet ↷ zustandsbehaftet.
- Die Problemstellung ist identisch zu [KM-gZ-Tu3](#). Allerdings ergibt sich dadurch, dass die Tunnelentität selbst zustandsbehaftet ist, eine weitere Lösungsmöglichkeit.
- Lösung L1-(KM-gZ-Tu4) ★ **L1-(KM-gZ-Tu4):** Zustandssignalisierung indirekt durch Zustand der Tunnelentität.
- ▶ **Voraussetzung:** Bijektive Abbildung von Zuständen zwischen Quell- / Zielentität einerseits und Tunnelentität andererseits.
 - ▶ **Lösungskomponenten:** Zwei Umsetzungskomponenten nach [L1-\(KM-gZ-Tr4\)](#) an den Übergängen Quellentität → Tunnelentität und Tunnelentität → Zielentität.
 - ▶ **Lösungsworkflow:** Analog zu [L1-\(KM-gZ-Tr4\)](#).
- Lösung L2-(KM-gZ-Tu4) ★ **L2-(KM-gZ-Tu4):** Zusätzliche tunnelinterne Zustandssignalisierung.
- ▶ **Analoge Lösung:** [L1-\(KM-gZ-Tu3\)](#).
- Lösung L3-(KM-gZ-Tu4) ★ **L3-(KM-gZ-Tu4):** Zusätzliche tunnelexterne Zustandssignalisierung.
- ▶ **Analoge Lösung:** [L2-\(KM-gZ-Tu3\)](#).

Für ein konkreteres Beispiel sei auf Unterabschnitt [5.3.1.1](#) verwiesen. Dort wird das allgemeine Muster auf das Attribut *Verbindungsorientiert* der Klasse *Daten-Kategorie* angewendet.

5.3 VERGLEICH DER STATISCHEN MODELLTEILE

Nachdem in Sektion 5.2 erste allgemeine Eigenschaften einer Koppelung festgestellt wurden, beginnt nun der detaillierte Vergleich der zu koppeln- den Netz-QoS-Dienste. Wie bei der Entwicklung der Modellierung, werden im ersten Teil die statischen Eigenschaften untersucht und miteinander verglichen. Auf die in dieser Sektion gefundenen Lösungsmöglichkeiten können später im nächsten Teil die dynamischen Analysen angewendet werden.

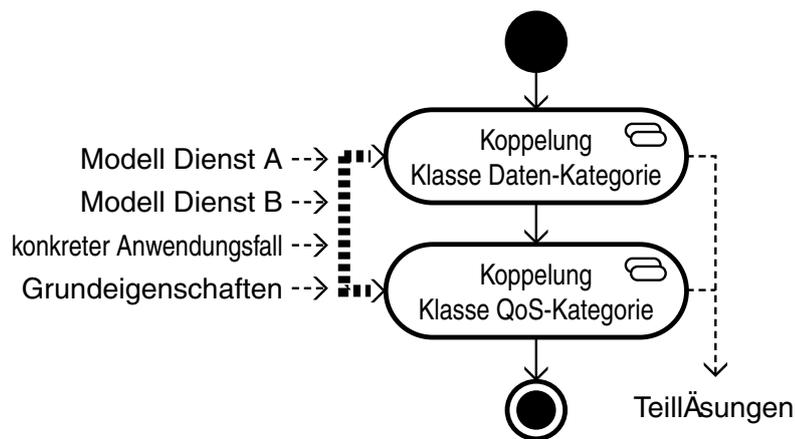


Abbildung 5.9: Aktivitäten zum Vergleich statischer Modellteile

In Abbildung 5.9 sind die beiden Hauptschritte abgebildet, die in dieser Sektion behandelt werden. In Abschnitt 5.3.1 wird zuerst die Koppelung bezüglich der Klasse *Daten-Kategorie* untersucht. Die zweite Teilaktivität widmet sich dann in Abschnitt 5.3.2 der Klasse *QoS-Kategorie*. Das Ergebnis beider Aktivitäten sind Teillösungen, die zur Gesamtlösung beitragen.

5.3.1 Gegenüberstellung statischer Teile von Daten-Kategorien

Der erste große Teilprozess der Koppelungsmethodik beschäftigt sich mit der Klasse *Daten-Kategorie*, ihren Attributen und der assoziierten Klasse *Kommunikationsrelation*. Wie bei der Modellentwicklung, werden zuerst rein statische Eigenschaften betrachtet. Die Kardinalitäten der Instanzen, sowie die dynamischen Eigenschaften fließen in spätere Schritte ein.

Hauptaufgabe des Teilprozesses aus Abbildung 5.10 ist es, zwischen den zu koppelnden Netz-QoS-Diensten die beste Kombination von zu verbindenden *Daten-Kategorien* zu finden. Hierzu werden in einem iterativen Prozess, der eine Spezialisierung des Musters aus Unterabschnitt 5.2.3.1 ist, alle Kombinationen von *Daten-Kategorien* miteinander verglichen.

Iteration über alle Paarungen von Daten-Kategorien

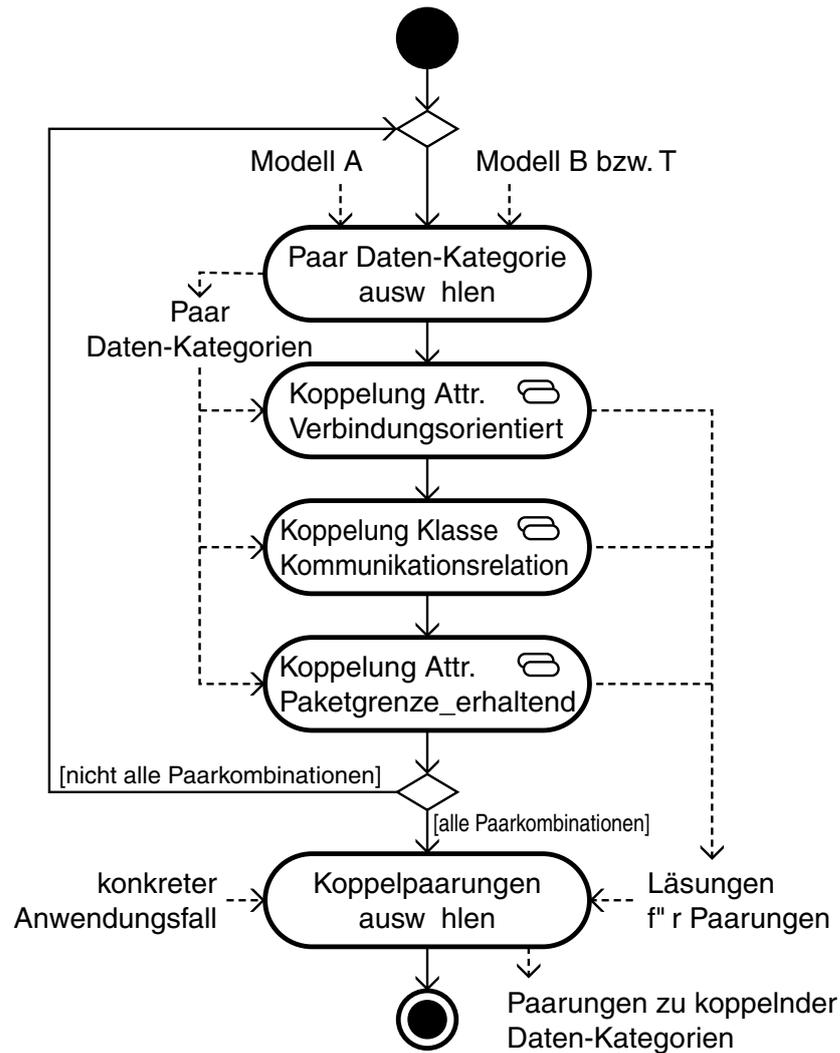


Abbildung 5.10: Aktivitäten zur Koppelung der Klasse *Daten-Kategorie*

- Teilschritte einer Iteration

Ein Iterationsschritt beginnt mit der Auswahl von jeweils einer *Daten-Kategorie* aus den Modellen der Dienste *A* und *B* bzw. *T*. Dieses Paar von *Daten-Kategorien* bildet die Eingabeartefakte für die anschließenden drei Schritte. Dort werden die beiden Attribute *Verbindungsorientiert* und *Paketgrenze_erhaltend* und die assoziierte Klasse *Kommunikationsrelation* im Detail gegenübergestellt. Die hierfür notwendigen Teilprozesse werden in den kommenden Unterabschnitten erläutert. Das Ergebnis aller drei Teilprozesse sind Lösungsmuster. Sie beschreiben die für eine Koppelung notwendigen Aktivitäten und Komponenten sowie den erzielbaren Funktionalitätsumfang im gekoppelten System.
- Schritt Koppelpaarungen auswählen

Im letzten Schritt des Teilprozesses werden dann die tatsächlich zu koppelnden Paarungen von *Daten-Kategorien* ausgewählt. Eine Einflussgröße dafür ist die Menge der analysierten Paarungen mit den für deren Koppelung notwendigen Lösungsschritte. Daran kann bemessen werden, wie aufwändig und wie erfolgversprechend die jeweiligen Paarungen sind. Der konkrete

Anwendungsfall muss dabei ebenfalls in die Entscheidung mit einbezogen werden. Wenn beispielsweise keine Variante die volle Funktionalität der Einzeldienste im gekoppelten System realisieren kann, kann anhand des konkreten Anwendungsfalls entschieden werden, welche der eingeschränkten Varianten am ehesten akzeptabel ist. Betriebswirtschaftliche Überlegungen (z.B. hinsichtlich der Kosten) können hier ebenfalls einfließen. Detaillierte Anweisungen lassen sich hier nicht geben, vielmehr sind individuelle Entscheidungen zu treffen. Das Ergebnis dieses Schritts kann auch sein, dass eine Koppelung nicht sinnvoll ist. Die Methodik endet dann bereits hier.

Individuelle
Entscheidung

Abbruch möglich

5.3.1.1 Attribut Verbindungsorientiert

Die Koppelung des Attributs *Verbindungsorientiert* stellt einen Spezialfall der in Abschnitt 5.2.4 vorgestellten allgemeinen Fallmuster KM-gZ-* dar. Die zustandslose Entität ist dabei eine *Daten-Kategorie* mit *Verbindungsorientiert==false*, eine zustandsbehaftete Entität ist durch eine *Daten-Kategorie* mit *Verbindungsorientiert==true* gegeben. Nachfolgend geschieht die Untersuchung zunächst nur für die asymmetrische Koppelung. Eine symmetrische Koppelung wird durch den im Anschluss geschilderten Teilprozess der Methodik abgedeckt. Die Interpretation von KM-gZ-* ergibt:

- ◆ **KM-Vo-Tr1:** Transitiv-Koppelung mit verbindungslos → verbindungslos. Fall KM-Vo-Tr1

Keine Probleme, da beide Dienste verbindungslos sind.

- ◆ **KM-Vo-Tr2:** Transitiv-Koppelung mit verbindungslos → verbindungsorientiert. Fall KM-Vo-Tr2

Beim Übergang von einer verbindungslosen zu einer verbindungsorientierten *Daten-Kategorie* entsteht das Problem, dass im Zieldienst vor Übertragungsbeginn eine Verbindung aufgebaut werden muss.

- ★ **L1-(KM-Vo-Tr2):** Globale, statische Verbindung. Lösung L1-(KM-Vo-Tr2)

- ▶ **Voraussetzung:** Zielverbindung muss a priori bestimmbar sein.

- ▶ **Lösungskomponenten:** Keine.

- ▶ **Lösungsworkflow:**

- Bereitstellungsphase: Aufbau der Verbindung im Zieldienst.

- Beendigungsphase: Löschen der Verbindung im Zieldienst.

- ★ **L2-(KM-Vo-Tr2):** Dynamischer Verbindungsaufbau. Lösung L2-(KM-Vo-Tr2)

- ▶ **Voraussetzung:** Keine.

- ▶ **Lösungskomponenten:** Wächterkomponente, die kontrolliert, wann die Datenübertragung beginnt bzw. endet und dynamisch im Zieldienst eine Verbindung auf- bzw. abbaut.

- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren der Wächterkomponente.
- ▶ **Problempotential:** Erkennen des Endes der Datenübertragung. Kann beispielsweise durch Timeout oder höhere Protokollschichten erfolgen.

Fall KM-Vo-Tr3 ◆ **KM-Vo-Tr3:** Transitiv-Koppelung mit verbindungsorientiert → verbindungslos.

Problemlos, weil keine Verbindung im Zieldienst aufgebaut werden muss.

Fall KM-Vo-Tr4 ◆ **KM-Vo-Tr4:** Transitiv-Koppelung mit verbindungsorientiert → verbindungsorientiert.

Liegt ein Übergang vor, bei dem die *Daten-Kategorien* in Quell- und Zieldienst verbindungsorientiert sind, dann müssen im Zieldienst Verbindungen aufgebaut werden, nachdem dies im Quelldienst erfolgt ist.

Lösung
L1-(KM-Vo-Tr4)

★ **L1-(KM-Vo-Tr4):** Dynamischer Verbindungsaufbau.

- ▶ **Voraussetzung:** Keine.
- ▶ **Lösungskomponenten:** Umsetzungskomponente, die dynamisch den im Quelldienst stattfindenden Verbindungsaufbau auch im Zieldienst initiiert.
- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren der Umsetzungskomponente.
- ▶ **Problempotential:** Signalisierung eines erfolglosen Verbindungsaufbaus im Zieldienst an den Quelldienst.

Fall KM-Vo-Tu1 ◆ **KM-Vo-Tu1:** Tunnel-Koppelung mit verbindungslos ↷ verbindungslos.

Analog zu [KM-Vo-Tr1](#) tritt hier kein Problem auf, da alle Dienste verbindungslos sind.

Fall KM-Vo-Tu2 ◆ **KM-Vo-Tu2:** Tunnel-Koppelung mit verbindungslos ↷ verbindungsorientiert.

Am Tunnelleingang liegt der Fall [KM-Vo-Tr2](#) vor. Der Tunnelausgang entspricht Fall [KM-Vo-Tr3](#), bei dem nichts zu tun ist. Da zwischen Quell- und Zieldienst kein Verbindungsaufbau stattfinden muss, ist das zu lösende Problem nur analog zu [KM-Vo-Tr2](#), da [KM-gZ-Tr3](#) problemlos ist.

Lösung
L1-(KM-Vo-Tu2)

★ **L1-(KM-Vo-Tu2):** Globale, statische Verbindung im Tunneldienst.

- ▶ **Analoge Lösung:** [L1-\(KM-Vo-Tr2\)](#), zwischen Quell- und Tunneldienst.

Lösung
L2-(KM-Vo-Tu2)

★ **L2-(KM-Vo-Tu2):** Dynamischer Verbindungsaufbau im Tunneldienst

- ▶ **Analoge Lösung:** [L2-\(KM-Vo-Tr2\)](#), zwischen Quell- und Tunneldienst.

- ◆ **KM-Vo-Tu3:** Tunnel-Koppelung mit verbindungsorientiert ↷ verbindungslos. Fall KM-Vo-Tu3

Das Problem in diesem Fall besteht im Verbindungsaufbau zwischen Quell- und Zieldienst über einen verbindungslosen Tunneldienst hinweg. Ein Rückgriff auf den Fall [KM-Vo-Tr3](#) am Tunneleingang ist nicht sinnvoll, da dabei die Signalisierung für den Verbindungsaufbau im Zieldienst verloren geht.

- ★ **L1-(KM-Vo-Tu3):** Zusätzliche tunnelinterne Signalisierung für Verbindungsaufbau. Lösung
L1-(KM-Vo-Tu3)

- ▶ **Voraussetzung:** Tunnelinterne Übertragungsmöglichkeit für Signalisierung.
- ▶ **Lösungskomponenten:**
 - Sender am Tunneleingang, der Verbindungsaufbau signalisiert .
 - Empfänger am Tunnelausgang, der Verbindungsaufbau im Zieldienst vornimmt.
 - Signalisierungsprotokoll zwischen Sender und Empfänger, das den gleichen Tunnel benutzt, wie die Datenübertragung selbst.
- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger und Signalisierungsprotokoll.
- ▶ **Problempotential:** Beeinflussung des QoS der regulären Datenübertragung durch zusätzliches Signalisierungsprotokoll, da die Übertragung im gleichen Tunnel erfolgt.

- ★ **L2-(KM-Vo-Tu3):** Zusätzliche tunnelexterne Signalisierung für Verbindungsaufbau. Lösung
L2-(KM-Vo-Tu3)

- ▶ **Voraussetzung:** Tunnelexterne Übertragungsmöglichkeit für Signalisierung.
- ▶ **Lösungskomponenten:**
 - Sender am Tunneleingang, der Verbindungsaufbau signalisiert.
 - Empfänger am Tunnelausgang, der Verbindungsaufbau im Zieldienst vornimmt.
 - Signalisierungsprotokoll zwischen Sender und Empfänger, das nicht den gleichen Tunnel benutzt.
 - Zusätzlicher Übertragungsweg für Verbindungssignalisierung.
- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger, Signalisierungsprotokoll und Übertragungsweg für Verbindungssignalisierung.

Fall KM-Vo-Tu4 ♦ **KM-Vo-Tu4:** Tunnel-Koppelung mit verbindungsorientiert \supset verbindungsorientiert.

Die Problemstellung ist identisch zu [KM-Vo-Tu3](#). Allerdings ergibt sich dadurch, dass der Tunneldienst selbst verbindungsorientiert ist, eine weitere Lösungsmöglichkeit.

Lösung
L1-(KM-Vo-Tu4)

★ **L1-(KM-Vo-Tu4):** Verbindungssignalisierung indirekt durch Zustand des Tunneldienstes.

► **Voraussetzung:** Wechselseitig kompatible Verbindungssignalisierung zwischen Quell- / Zieldienst und Tunneldienst.

► **Lösungskomponenten:** Zwei Umsetzungskomponenten nach [L1-\(KM-Vo-Tr4\)](#) an den Übergängen Quelldienst → Tunneldienst und Tunneldienst → Zieldienst.

► **Lösungsworkflow:** Analog zu [L1-\(KM-Vo-Tr4\)](#).

Lösung
L2-(KM-Vo-Tu4)

★ **L2-(KM-Vo-Tu4):** Zusätzliche tunnelinterne Signalisierung für Verbindungsaufbau.

► **Analoge Lösung:** [L1-\(KM-Vo-Tu3\)](#).

Lösung
L3-(KM-Vo-Tu4)

★ **L3-(KM-Vo-Tu4):** Zusätzliche tunnelexterne Signalisierung für Verbindungsaufbau.

► **Analoge Lösung:** [L2-\(KM-Vo-Tu3\)](#).

Fallübersicht

Tabelle 5.2 fasst alle eben besprochenen Fälle nochmals zusammen:

Transitiv-Koppelung Quelldienst → Zieldienst					
Fallbezeichnung	A:Verbor.	B:Verbor.	Aktion notwendig	Lösungen	potent. vollst.
KM-Vo-Tr1	false	false	-	-	✓
KM-Vo-Tr2	false	true	ja	L1-(KM-Vo-Tr2) L2-(KM-Vo-Tr2)	✓
KM-Vo-Tr3	true	false	-	-	✓
KM-Vo-Tr4	true	true	ja	L1-(KM-Vo-Tr4)	✓
Tunnel-Koppelung Quell- / Zieldienst \supset Tunneldienst					
Fallbezeichnung	A:Verbor.	T:Verbor.	Aktion notwendig	Lösungen	potent. vollst.
KM-Vo-Tu1	false	false	-	-	✓
KM-Vo-Tu2	false	true	ja	L1-(KM-Vo-Tu2) L2-(KM-Vo-Tu2)	✓
KM-Vo-Tu3	true	false	ja	L1-(KM-Vo-Tu3) L2-(KM-Vo-Tu3)	✓
Fortsetzung auf der nächsten Seite					

5.3 VERGLEICH DER STATISCHEN MODELLEILE

Fortsetzung von der vorhergehenden Seite					
KM-Vo-Tu4	<i>true</i>	<i>true</i>	ja	L1-(KM-Vo-Tu4) L2-(KM-Vo-Tu4) L3-(KM-Vo-Tu4)	✓

Tabelle 5.2: Koppelungsmuster für Attribut *Verbindungsorientiert*

Teilprozess der Methodik

Den Teilprozess zur Koppelung des Attributs *Verbindungsorientiert* erhält man durch Anwendung des allgemeinen Musters aus Unterabschnitt 5.2.3.2. Durch den in Abbildung 5.11 dargestellten Teilprozess wird jetzt auch der Fall einer symmetrischen Koppelung abgedeckt. Für die drei Aktivitäten, in denen Fallanalysen vorgenommen werden müssen, kann Tabelle 5.2 benutzt werden.

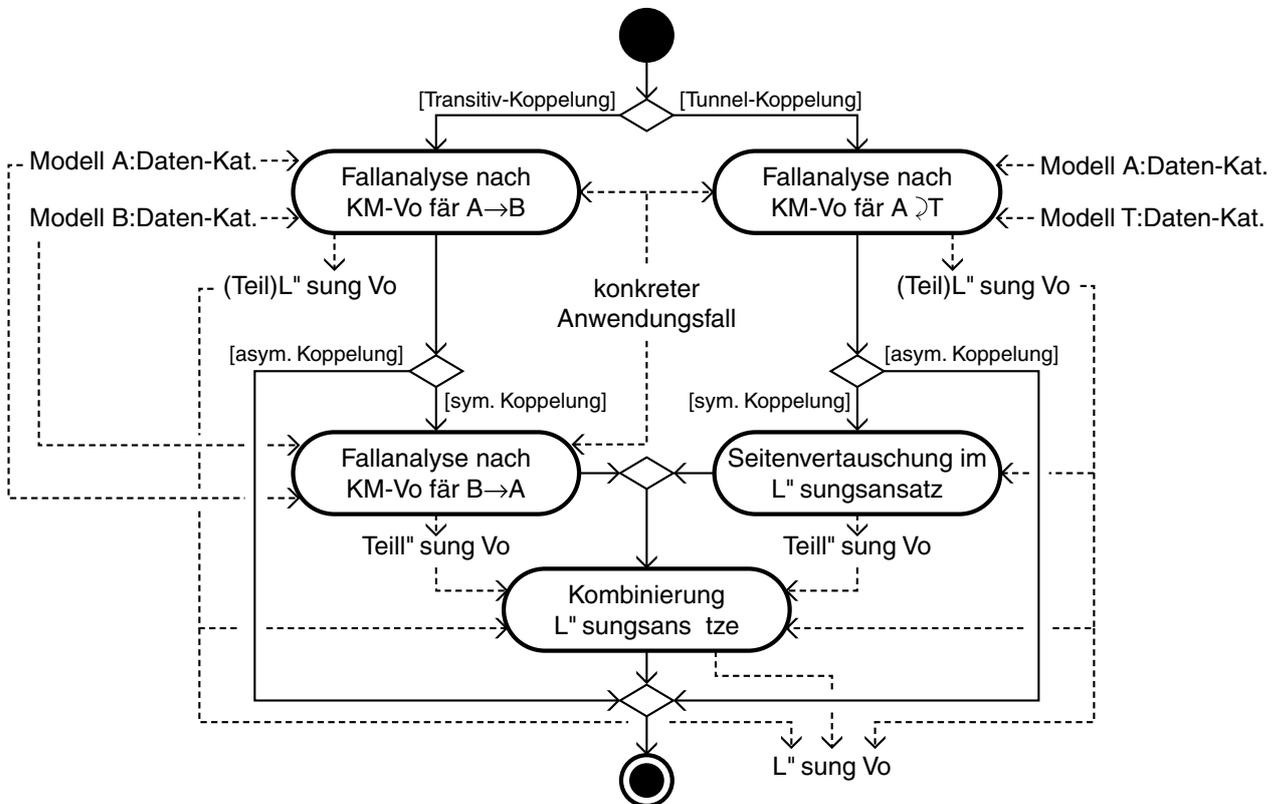


Abbildung 5.11: Aktivitäten zur Koppelung des Attributs *Verbindungsorientiert*

5.3.1.2 Klasse Kommunikationsrelation

Zur Analyse der Fälle bei der Koppelung verschiedener *Kommunikationsrelationen* werden zunächst alle Fälle für eine asymmetrische Koppelung betrachtet. Für eine symmetrische Koppelung wird gleich im Anschluss ein entsprechender Teilworkflow vorgestellt, mit dem dieser Koppelfall bewältigt wird.

Untersuchung nach Mächtigkeit Für die Betrachtung der *Kommunikationsrelationen* kann eine Untersuchung nach Mächtigkeit angewendet werden: Ist ein nachgeschalteter Dienst mindestens gleich mächtig (bei asymmetrischer Transitiv-Koppelung) bzw. ein Tunneldienst mindestens gleich mächtig zum Quell- / Zieldienst (bei asymmetrischer Tunnel-Koppelung), dann besteht im Prinzip kein Problemfall. Hinsichtlich der *Kommunikationsrelation* sind bidirektionale Kommunikationsrelationen mächtiger als unidirektionale.

Unterschieden werden muss aber noch zwischen *Unicast*- und *Multicast*-Varianten, sowie den Grundanwendungsfällen. Die Anwendung des Mächtigkeitsprinzips ist nur bei homogenen Uni- / Multicast-Varianten Transitiv-Koppelungen sinnvoll.

Außerdem ergeben sich bei allen Tunnelkoppelungen Optimierungsmöglichkeiten, die gesondert betrachtet werden müssen. Ebenso müssen alle heterogenen Uni- / Multicast-Varianten eigens untersucht werden. Daraus ergibt sich insgesamt:

Homogenen *Kommunikationsrelationen* sind Transitiv-Koppelungen problemlos, wenn der nachgeschaltete Dienst bzw. der Tunneldienst bidirektional ist oder beide Dienste unidirektional sind. Alle anderen Fälle müssen eingehender untersucht werden. Nachfolgend geschieht diese Untersuchung zunächst nur für die asymmetrische Koppelung. Eine symmetrische Koppelung wird durch den im Anschluss geschilderten Teilprozess der Methodik abgedeckt.

Spezielle homogene Uni- / Multicast-Varianten

Fall KM-Kr-UUTr₃ ♦ **KM-Kr-UUTr₃**: Transitiv-Koppelung *Unicast bidirektional* → *Unicast unidirektional*.

Die Problematik besteht darin, dass für die Nachbildung des bidirektionalen Quelldienstes auf einem unidirektionalen Zieldienst zwei Kanäle (einer für jede Richtung) benötigt werden.

Lösung L₁-(KM-Kr-UUTr₃) ★ **L₁-(KM-Kr-UUTr₃)**: Splitten der bidirektionalen Kommunikation beim Dienstübergang.

Die Idee besteht darin, die eine bidirektionale *Daten-Kategorie* durch zwei unidirektionale *Daten-Kategorien* (für jede Richtung eine) zu substituieren.

- ▶ **Voraussetzung:** Anwendung, die den gekoppelten Dienst benutzt, muss gesplittete Kommunikationskanäle unterstützen.
- ▶ **Lösungskomponenten:** Splittingkomponente, die die beiden Richtungen des Quelldienstes in zwei getrennte Instanzen von *Daten-Kategorien* aufteilt.
- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren der Splittingkomponente.

5.3 VERGLEICH DER STATISCHEN MODELLETEILE

- ▶ **Problempotential:** Wenn die Anwendung keine getrennten Kanäle unterstützt, dann scheitert die Koppelung komplett.

◆ **KM-Kr-UUTu₃:** Tunnel-Koppelung *Unicast bidirektional* \supset *Unicast unidirektional*. Fall KM-Kr-UUTu₃

Dadurch, dass der schwächere Dienst als ein Tunneldienst auftritt, kann die Problematik ohne Einschränkung überwunden werden.

- ★ **L₁-(KM-Kr-UUTu₃):** Splitten / Zusammensetzen der bidirektionalen Kommunikation an den Tunnelenden. Lösung L₁-(KM-Kr-UUTu₃)

Die Idee besteht darin, die bidirektionale *Daten-Kategorie* des Quell- / Zieldienstes im Tunneldienst durch zwei unidirektionale *Daten-Kategorien* (für jede Richtung eine) zu substituieren.

- ▶ **Voraussetzung:** Keine.
- ▶ **Lösungskomponenten:** Splittingkomponenten in Quell- und Zieldienst, die die beiden Richtungen aus Quell- / Zieldienst in zwei getrennte Instanzen von *Daten-Kategorien* aufteilen bzw. wieder zusammenführen.
- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren der Splittingkomponenten.

◆ **KM-Kr-MMT₃:** Transitiv-Koppelung *Multicast m:n* \rightarrow *Multicast 1:n*. Fall KM-Kr-MMT₃

Das Problem hier ist, dass die Rückrichtungen der Multicastkommunikation des Quelldienstes im Zieldienst nicht verfügbar sind.

- ★ **L₁-(KM-Kr-MMT₃):** Nachbildung der Rückrichtung durch zusätzliche Multicastgruppen im Zieldienst. Lösung L₁-(KM-Kr-MMT₃)

Zur Nachbildung der m:n Kommunikation im Zieldienst werden die Rückrichtungen durch zusätzliche Multicastgruppen gebildet, die für jeden Teilnehmer im Zieldienst einzeln eingerichtet werden müssen.

- ▶ **Problempotential:** Die Lösung ist eher akademischer Natur, da im Extremfall bis zu m*n Multicastgruppen in der Zielarchitektur erstellt werden müssten. Der dafür erforderliche Managementaufwand ist enorm.

◆ **KM-Kr-MMTu_{1/2/4}:** Tunnel-Koppelung *Multicast* \supset *Multicast* mit mindestens gleich mächtigem Tunneldienst. Fall KM-Kr-MMTu_{1/2/4}

Hier besteht kein echtes Problem, allerdings ist der Tunneldienst stärker als notwendig. Da kein Teilnehmer der Multicastkommunikation im Tunneldienst sitzt, ist die Verwendung von Multicastkommunikation letztlich Verschwendung.

- ★ **L₁-(KM-Kr-MMTu_{1/2/4}):** Benutzung des Multicast-Dienstes als Unicast-Dienst und eigenes Tunnel-Protokoll. Lösung L₁-(KM-Kr-MMTu_{1/2/4})

- ▶ **Analoge Lösung:** L₁-(KM-Kr-MUTu_{1/2/4}).

- **Zusätzliche Aktivitäten:** Zusätzlich zur analogen Lösung sind keine weiteren speziellen Komponenten und Workflows notwendig. Eine Multicastgruppe mit nur zwei Teilnehmern wird anstatt des Unicast-Dienstes benutzt.

Fall KM-Kr-MMTu3

◆ **KM-Kr-MMTu3:** Tunnel-Koppelung *Multicast m:n* ↷ *Multicast 1:n*.

Die Problemstellung ist analog zu [KM-Kr-MMTu1/2/4](#), da ein Multicast-Dienst hier als Tunneldienst auftritt.

Lösung
L1-(KM-Kr-MMTu3)

★ **L1-(KM-Kr-MMTu3):** Benutzung des Multicast-Dienstes als Unicast-Dienst und eigenes Tunnelprotokoll mit zwei Datenkanälen.

► **Analoge Lösung:** [L1-\(KM-Kr-MUTu3\)](#).

Fallübersicht

Transitiv-Koppelung <i>Unicast</i> → <i>Unicast</i>					
Fall-bezeichnung	Quell-dienst A	Ziel-dienst B	Akt. notw.	Lösungen	potent. vollst.
KM-Kr-UUTr1	<i>unidir</i>	<i>unidir</i>	-	-	✓
KM-Kr-UUTr2	<i>unidir</i>	<i>bidir</i>	-	-	✓
KM-Kr-UUTr3	<i>bidir</i>	<i>unidir</i>	ja	L1-(KM-Kr-UUTr3)	✗
KM-Kr-UUTr4	<i>bidir</i>	<i>bidir</i>	-	-	✓
Tunnel-Koppelung <i>Unicast</i> ↷ <i>Unicast</i>					
Fall-bezeichnung	Quell / Ziel-dienst A	Tunnel-dienst T	Akt. notw.	Lösungen	potent. vollst.
KM-Kr-UUTu1	<i>unidir</i>	<i>unidir</i>	-	-	✓
KM-Kr-UUTu2	<i>unidir</i>	<i>bidir</i>	-	-	✓
KM-Kr-UUTu3	<i>bidir</i>	<i>unidir</i>	ja	L1-(KM-Kr-UUTu3)	✓
KM-Kr-UUTu4	<i>bidir</i>	<i>bidir</i>	-	-	✓

Tabelle 5.3: Koppelungsmuster für Kommunikationsrelationen *Unicast* / *Unicast*

Transitiv-Koppelung <i>Multicast</i> → <i>Multicast</i>					
Fall-bezeichnung	Quell-dienst A	Ziel-dienst B	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-MMTr1	<i>1:n</i>	<i>1:n</i>	-	-	✓
KM-Kr-MMTr2	<i>1:n</i>	<i>m:n</i>	-	-	✓
KM-Kr-MMTr3	<i>m:n</i>	<i>1:n</i>	ja	L1-(KM-Kr-MMTr3)	✗
KM-Kr-MMTr4	<i>m:n</i>	<i>m:n</i>	-	-	✓
Fortsetzung auf der nächsten Seite					

Fortsetzung von der vorhergehenden Seite					
Tunnel-Koppelung <i>Multicast</i> \curvearrowright <i>Multicast</i>					
Fall-bezeichnung	Quell/Zield. <i>A</i>	Tunnel-dienst <i>T</i>	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-MMTu1	1:n	1:n	ja	L1-(KM-Kr-MMTu1/2/4)	✓
KM-Kr-MMTu2	1:n	m:n	ja	L1-(KM-Kr-MMTu1/2/4)	✓
KM-Kr-MMTu3	m:n	1:n	ja	L1-(KM-Kr-MMTu3)	✓
KM-Kr-MMTu4	m:n	m:n	ja	L1-(KM-Kr-MMTu1/2/4)	✓

Tabelle 5.4: Koppelungsmuster für Kommunikationsrelationen *Multicast* / *Multicast*

Spezielle heterogene Uni- / Multicast-Varianten

Allgemein ist anzumerken, dass alle heterogenen Uni- / Multicast-Fälle mit Ausnahme der Fälle [KM-Kr-MUTu1/2/4](#), [KM-Kr-MUTu3](#) wenn irgendmöglich zu vermeiden sind. Zwar sind Koppelungen in einigen Fällen verlustfrei möglich, erzeugen aber einen beträchtlichen Managementaufwand, der nicht gerechtfertigt ist:

- ◆ **KM-Kr-UMTr1/2/4:** Transitiv-Koppelung *Unicast* \rightarrow *Multicast* mit mindestens gleich mächtigem nachgeschalteten Dienst. Fall KM-Kr-UMTr1/2/4

Letztlich liegt kein echtes Problem vor, allerdings ist diese Variante sehr verschwenderisch, da in der Lösung die Multicastkommunikation vielfach zur Unicastkommunikation „missbraucht“ wird.

- ★ **L1-(KM-Kr-UMTr1/2/4):** Benutzung des Multicast-Dienstes als Unicast-Dienst. Lösung L1-(KM-Kr-UMTr1/2/4)

Faktisch sind hier keine speziellen Komponenten und Workflows notwendig. Jede Multicastgruppe hat einfach nur zwei Teilnehmer.

- ◆ **KM-Kr-UMTr3:** Transitiv-Koppelung *Unicast bidirektional* \rightarrow *Multicast* Fall KM-Kr-UMTr3
1:n.

Auch dies ist analog zu [KM-Kr-UMTr1/2/4](#) eine der extrem verschwenderischen Varianten. Der Aufwand verdoppelt sich hier sogar noch:

- ★ **L1-(KM-Kr-UMTr3):** Splitten der bidirektionalen Kommunikation beim Dienstübergang. Lösung L1-(KM-Kr-UMTr3)

- ▶ **Analoge Lösung:** [L1-\(KM-Kr-UUTr3\)](#).

- ▶ **Zusätzliche Aktivitäten:** Benutzung des Multicast-Dienstes als Unicast-Dienst, analog zu [L1-\(KM-Kr-UMTr1/2/4\)](#).

- ◆ **KM-Kr-UMTu1/2/4:** Tunnel-Koppelung *Unicast* \curvearrowright *Multicast* mit mindestens gleich mächtigem nachgeschalteten Dienst. Fall KM-Kr-UMTu1/2/4

Ebenfalls eine der Verschwendungsvarianten, bei der eine Unicastkommunikation im Tunnel-Dienst eigentlich genügen würde.

- Lösung L1-(KM-Kr-UMTu1/2/4) ★ **L1-(KM-Kr-UMTu1/2/4)**: Benutzung des Multicast-Dienstes als Unicast-Dienst.
- ▶ **Analoge Lösung:** L1-(KM-Kr-UMTr1/2/4).
 - ▶ **Zusätzliche Aktivitäten:** Lösungsmöglichkeit reduziert die Fälle auf die problemlosen Fälle KM-Kr-UUTr1/2/4.
- Fall KM-Kr-UMTu3 ♦ **KM-Kr-UMTu3**: Tunnel-Koppelung *Unicast bidirektional* \supset *Multicast 1:n*.
Wiederum eine der Verschwendungsvarianten, bei der eine Unicastkommunikation im Tunnel-Dienst eigentlich genügen würde. Hier sogar mit doppeltem Aufwand.
- Lösung L1-(KM-Kr-UMTu3) ★ **L1-(KM-Kr-UMTu3)**: Splitten / Zusammensetzen der bidirektionalen Kommunikation an den Tunnelenden mit Benutzung des Multicast-Dienstes als Unicast-Dienst.
- ▶ **Analoge Lösung:** L1-(KM-Kr-UUTu3), L1-(KM-Kr-UMTr1/2/4).
- Fall KM-Kr-MUTr1/2/4 ♦ **KM-Kr-MUTr1/2/4**: Transitiv-Koppelung *Multicast* \rightarrow *Unicast* mit mindestens gleich mächtigem nachgeschalteten Dienst.
Die Problematik dieses Falls besteht darin, dass eine Multicastkommunikation durch eine Menge von Unicastkommunikationen substituiert werden muss. Hierdurch entsteht ein enormer Managementaufwand, um die Menge der Unicastkommunikationen kohärent zur Multicastgruppe zu halten.
- Lösung L1-(KM-Kr-MUTr1/2/4) ★ **L1-(KM-Kr-MUTr1/2/4)**: Nachbildung des Multicast-Mechanismus durch Gruppe einzelner Unicast-Kommunikation.
In [SchmJB 01] werden diverse Varianten vorgestellt.
- ▶ **Problempotential:** Die Lösung ist eher akademischer Natur, da im Extremfall bis zu $m \cdot n$ Multicastgruppen in der Zielarchitektur erstellt werden werden müssten. Der dafür erforderliche Managementaufwand ist enorm.
- Fall KM-Kr-MUTr3 ♦ **KM-Kr-MUTr3**: Transitiv-Koppelung *Multicast m:n* \rightarrow *Unicast unidirektional*.
Dieser Fall ist der ungünstigste aller Fälle bei der Koppelung der *Kommunikationsrelationen*. Zu den Schwierigkeiten aus Fall **KM-Kr-MUTr1/2/4** kommt hier noch die Tatsache, dass die ungerichtete Gruppenkommunikation auch im Zieldienst durch unidirektionale Unicastkommunikation nachgebildet werden muss. Im Endeffekt bedeutet dies, dass ein vollvermaschtes Netz aller Teilnehmer im Zieldienst aufgebaut werden muss. Möglich ist dies aber nur dann (analog zu Fall **KM-Kr-UUTr3**), wenn die Anwendung, die den gekoppelten Dienst benutzt, mit gesplitteten Kommunikationskanälen umgehen kann.
Der Managementaufwand zur Realisierung einer solchen Koppelung ist selbst bei einer einstelligen Teilnehmeranzahl so enorm, dass sie prak-

5.3 VERGLEICH DER STATISCHEN MODELLETEILE

tisch nicht mehr handhabbar ist. Deshalb wird auf die Angabe einer Lösungsvariante hier verzichtet.

- ◆ **KM-Kr-MUTu1/2/4:** Tunnel-Koppelung *Multicast* \supset *Unicast* mit mindestens gleich mächtigem nachgeschalteten Dienst. Fall KM-Kr-MUTu1/2/4

Dies ist die bevorzugte Variante bei der Tunnel-Koppelung von *Multicast-Daten-Kategorien*.

- ★ **L1-(KM-Kr-MUTu1/2/4):** Eigenes Tunnelprotokoll mit einem Datenkanal. Lösung L1-(KM-Kr-MUTu1/2/4)

Die Idee dieser Lösung besteht darin, zur Übertragung der *Multicast*-Kommunikation im Tunneldienst ein eigenes Tunnelprotokoll zu verwenden, das selbst nur *Unicast*-Kommunikation benötigt.

► **Lösungskomponenten:**

- Sender am Tunneleingang, der *Multicast*-Kommunikation in eigenes Protokoll einpackt.
- Empfänger am Tunnelausgang, der *Multicast*-Kommunikation aus eigenem Protokoll wiederherstellt.
- Tunnelprotokoll zwischen Sender und Empfänger.

► **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger und Tunnelprotokoll.

- ◆ **KM-Kr-MUTu3:** Tunnel-Koppelung *Multicast m:n* \supset *Unicast unidirektional*. Fall KM-Kr-MUTu3

Ebenfalls eine der bevorzugten Varianten bei der Tunnel-Koppelung von *Multicast-Daten-Kategorien*.

- ★ **L1-(KM-Kr-MUTu3):** Eigenes Tunnelprotokoll mit zwei getrennten Datenkanälen. Lösung L1-(KM-Kr-MUTu3)

Bei dieser Lösung wird das Splitting im Tunneldienst aus [L1-\(KM-Kr-UUTu3\)](#) kombiniert mit dem Tunnelprotokoll aus [L1-\(KM-Kr-MUTu1/2/4\)](#).

► **Analoge Lösung:** [L1-\(KM-Kr-UUTu3\)](#) kombiniert mit [L1-\(KM-Kr-MUTu1/2/4\)](#).

Fallübersicht

Transitiv-Koppelung <i>Unicast</i> \rightarrow <i>Multicast</i>					
Fallbezeichnung	Quelldienst A	Zieldienst B	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-UMTr1	<i>unidir</i>	<i>1:n</i>	ja	L1-(KM-Kr-UMTr1/2/4)	✓
KM-Kr-UMTr2	<i>unidir</i>	<i>m:n</i>	ja	L1-(KM-Kr-UMTr1/2/4)	✓
Fortsetzung auf der nächsten Seite					

KAPITEL 5: GENERISCHE KOPPELUNGSMETHODIK FÜR NETZ-QoS-DIENSTE

Fortsetzung von der vorhergehenden Seite					
KM-Kr-UMTr3	<i>bidir</i>	<i>1:n</i>	ja	L1-(KM-Kr-UMTr3)	✗
KM-Kr-UMTr4	<i>bidir</i>	<i>m:n</i>	ja	L1-(KM-Kr-UMTr1/2/4)	✓
Tunnel-Koppelung <i>Unicast</i> ↷ <i>Multicast</i>					
Fall-bezeichnung	Quell/Zield. A	Tunnel-dienst T	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-UMTu1	<i>unidir</i>	<i>1:n</i>	ja	L1-(KM-Kr-UMTu1/2/4)	✓
KM-Kr-UMTu2	<i>unidir</i>	<i>m:n</i>	ja	L1-(KM-Kr-UMTu1/2/4)	✓
KM-Kr-UMTu3	<i>bidir</i>	<i>1:n</i>	ja	L1-(KM-Kr-UMTu3)	✓
KM-Kr-UMTu4	<i>bidir</i>	<i>m:n</i>	ja	L1-(KM-Kr-UMTu1/2/4)	✓

Tabelle 5.5: Koppelungsmuster für Kommunikationsrelationen *Unicast* / *Multicast*

Transitiv-Koppelung <i>Multicast</i> → <i>Unicast</i>					
Fall-bezeichnung	Quell-dienst A	Ziel-dienst B	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-MUTr1	<i>1:n</i>	<i>unidir</i>	ja	L1-(KM-Kr-MUTr1/2/4)	✓
KM-Kr-MUTr2	<i>1:n</i>	<i>bidir</i>	ja	L1-(KM-Kr-MUTr1/2/4)	✓
KM-Kr-MUTr3	<i>m:n</i>	<i>unidir</i>	ja	–	✗
KM-Kr-MUTr4	<i>m:n</i>	<i>bidir</i>	ja	L1-(KM-Kr-MUTr1/2/4)	✓
Tunnel-Koppelung <i>Multicast</i> ↷ <i>Unicast</i>					
Fall-bezeichnung	Quell/Zield. A	Tunnel-dienst T	Akt. notw.	Lösungen	pot. vollst.
KM-Kr-MUTu1	<i>1:n</i>	<i>unidir</i>	ja	L1-(KM-Kr-MUTu1/2/4)	✓
KM-Kr-MUTu2	<i>1:n</i>	<i>bidir</i>	ja	L1-(KM-Kr-MUTu1/2/4)	✓
KM-Kr-MUTu3	<i>m:n</i>	<i>unidir</i>	ja	L1-(KM-Kr-MUTu3)	✓
KM-Kr-MUTu4	<i>m:n</i>	<i>bidir</i>	ja	L1-(KM-Kr-MUTu1/2/4)	✓

Tabelle 5.6: Koppelungsmuster für Kommunikationsrelationen *Multicast* / *Unicast*

Teilprozess der Methodik

Der Teilprozess der Methodik für die Klasse *Kommunikationsrelation* ist in Abbildung 5.12 dargestellt. Er ist vom Muster aus Unterabschnitt 5.2.3.2 abgeleitet. Die drei Fallanalysen sind anhand der Tabellen 5.3, 5.4, 5.5 und 5.6 durchzuführen. Der Schritt der Kombinierung der Lösungsansätze liefert in diesem Fall nicht nur eine Lösung zur Koppelung der Klasse. Anhand der tatsächlich ausgewählten Lösung kann jetzt auch die Kommunikationsrichtung (uni- / bidirektional, siehe Abschn. 5.2.1) des Gesamtsystems bestimmt werden.

5.3 VERGLEICH DER STATISCHEN MODELLEILE

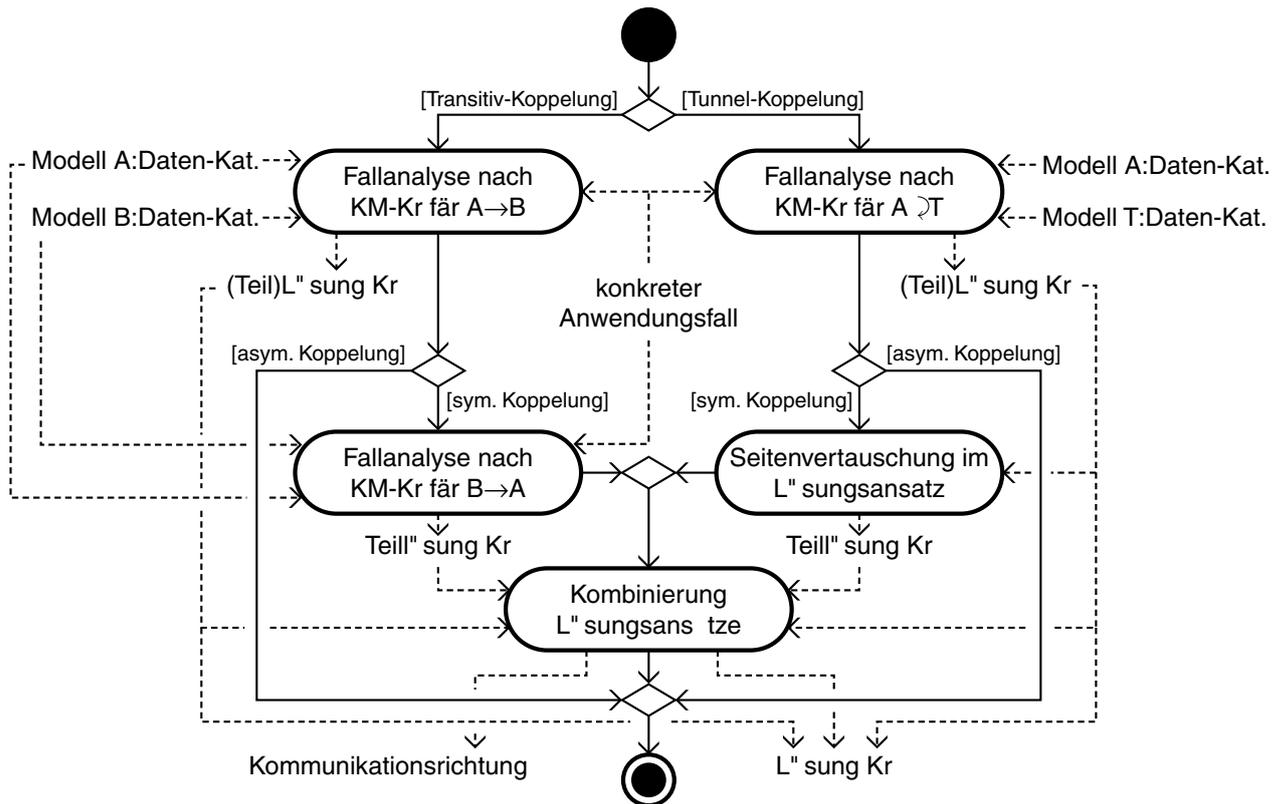


Abbildung 5.12: Aktivitäten zur Koppelung der Klasse *Kommunikationsrelation*

5.3.1.3 Attribut *Paketgrenze_erhaltend*

Für die Koppelung des booleschen Attributs *Paketgrenze_erhaltend* ist neben der Unterscheidung des Grundanwendungsfalls auch die Kommunikationsrichtung entscheidend. Dabei ist es unerheblich, von welcher Seite die Kommunikation initiiert wurde, d.h. ob symmetrische Koppelung vorliegt oder nicht. Die Kommunikationsrichtung ist ein Ergebnis des vorherigen Schritts.

Es ergibt sich kein Handlungsbedarf bei beidseitig gleicher Attributbelegung: Entweder es müssen auf keiner Seite die Paketgrenzen erhalten werden oder beide Seite können die Paketgrenzen erhalten. Dabei spielt es auch keine Rolle, ob die Kommunikation uni- oder bidirektional ist.

Auch in den unidirektionalen Fällen, bei denen sich die Attribute unterscheiden und der mächtigere Dienst (also mit *Paketgrenze_erhaltend==true*) der nachgeschaltete bzw. der Tunneldienst ist, muss nichts getan werden.

Bei allen verbleibenden Fällen liegt die Ursache in der Beteiligung eines Übergangs des Datenstroms von *Paketgrenze_erhaltend==true* → *Paketgrenze_erhaltend==false* :

- ◆ **KM-Pe-UdTr3**: Unidirektionale Transitiv-Koppelung mit *A:Paketgrenze_erhaltend==true* → *B:Paketgrenze_erhaltend==false* .

Die Eigenschaft kann nicht auf Dienst *B* abgebildet werden. Es kommt immer zu einem Informationsverlust beim Übergang.

- Lösung L1-(KM-Pe-UdTr3) ★ **L1-(KM-Pe-UdTr3)**: Reduktion der Eigenschaft des Gesamtsystems.
- ▶ **Voraussetzung:** *Paketgrenze_erhaltend* ist keine wesentliche Eigenschaft des Gesamtsystems.
 - ▶ **Lösungsworkflow:** Planungsphase: Neuspezifikation des Dienstes *A* auf *Paketgrenze_erhaltend=false* .
 - ▶ **Problempotential:** *Paketgrenze_erhaltend==true* könnte in späteren Anwendungsfällen benötigt werden.
- Lösung L2-(KM-Pe-UdTr3) ★ **L2-(KM-Pe-UdTr3)**: Abbruch des Koppelungsprozesses.
- ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Abbrechen des Koppelungsprozesses.
 - ▶ **Problempotential:** Koppelung kommt nicht zustande.
- Fall KM-Pe-UdTu3 ◆ **KM-Pe-UdTu3**: Unidirektionale Tunnel-Koppelung mit *Paketgrenze_erhaltend==true* \supset *Paketgrenze_erhaltend==false* .
- Die Problematik ist analog zu **KM-Pe-UdTr3**. Analog zu den Lösungsmöglichkeiten des allgemeinen Musters **KM-gZ-UdTu3** ergeben sich zwei weitere Lösungsvarianten:
- Lösung L1-(KM-Pe-UdTu3) ★ **L1-(KM-Pe-UdTu3)**: Zusätzliches tunnelinternes Protokoll zur Übermittlung der Paketgrenzen.
- ▶ **Voraussetzung:** Tunnelinterne Übertragungsmöglichkeit für zusätzliches Protokoll.
 - ▶ **Lösungskomponenten:** Sender für Paketgrenzen am Tunneleingang, Empfänger und Paketierer am Tunnelausgang und zusätzliches Protokoll zur Übermittlung der Paketgrenzen zwischen Dienst A_1 und A_2 . Es wird der Tunnel selbst zur Übermittlung benutzt.
 - ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger / Paketierer und Übermittlungsprotokoll.
 - ▶ **Problempotential:** Wechselwirkung Beeinflussung des QoS der Datenübertragung und der zusätzlichen Grenzenübermittlung.
- Lösung L2-(KM-Pe-UdTu3) ★ **L2-(KM-Pe-UdTu3)**: Zusätzliches tunnelexternes Protokoll zur Übermittlung der Paketgrenzen.
- ▶ **Voraussetzung:** Tunnelexterne Übertragungsmöglichkeit für zusätzliches Protokoll.
 - ▶ **Lösungskomponenten:** Sender für Paketgrenzen am Tunneleingang, Empfänger und Paketierer am Tunnelausgang und zusätzliches Protokoll zur Übermittlung der Paketgrenzen zwischen Dienst A_1 und A_2 . Es wird ein zusätzlicher Kanal zur Übermittlung benutzt.

5.3 VERGLEICH DER STATISCHEN MODELLETEILE

- ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger / Paketierer und Übermittlungsmechanismus.
- ★ **L3-(KM-Pe-UdTu3):** Reduktion der Eigenschaft des Gesamtsystems. Lösung L3-(KM-Pe-UdTu3)
 - ▶ **Voraussetzung:** *Paketgrenze_erhaltend* ist keine wesentliche Eigenschaft des Gesamtsystems.
 - ▶ **Lösungsworkflow:** Planungsphase: Neuspezifikation des Dienstes *A* auf *Paketgrenze_erhaltend=false*.
 - ▶ **Problempotential:** *Paketgrenze_erhaltend==true* könnte in späteren Anwendungsfällen benötigt werden.
- ★ **L4-(KM-Pe-UdTu3):** Abbruch des Koppelungsprozesses. Lösung L4-(KM-Pe-UdTu3)
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Abbrechen des Koppelungsprozesses.
 - ▶ **Problempotential:** Koppelung kommt nicht zustande.
- ◆ **KM-Pe-BdTr2:** Bidirektionale Transitiv-Koppelung mit unterschiedlichen Paketgrenzenerhaltungseigenschaften. Fall KM-Pe-BdTr2

Paketgrenze_erhaltend kann nur in der Richtung vom „schwächeren“ auf den „stärkeren“ Dienst abgebildet werden. In der anderen Richtung kommt es immer zu einem Informationsverlust beim Übergang.

 - ★ **L1-(KM-Pe-BdTr2):** Reduktion der Eigenschaft des Gesamtsystems. Lösung L1-(KM-Pe-BdTr2)
 - ▶ **Voraussetzung:** *Paketgrenze_erhaltend* ist keine wesentliche Eigenschaft des Gesamtsystems.
 - ▶ **Lösungsworkflow:** Planungsphase: Neuspezifikation des „stärkeren“ Dienstes auf *Paketgrenze_erhaltend=false*.
 - ▶ **Problempotential:** *Paketgrenze_erhaltend==true* könnte in späteren Anwendungsfällen benötigt werden.
 - ★ **L2-(KM-Pe-BdTr2):** Abbruch des Koppelungsprozesses. Lösung L2-(KM-Pe-BdTr2)
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Abbrechen des Koppelungsprozesses.
 - ▶ **Problempotential:** Koppelung kommt nicht zustande.
- ◆ **KM-Pe-BdTu2:** Bidirektionale Tunnel-Koppelung mit unterschiedlichen Paketgrenzenerhaltungseigenschaften. Fall KM-Pe-BdTu2

Die Problematik ist identisch zu [KM-Pe-UdTu3](#). Allerdings tritt sie an beiden Tunnelenden auf. Entsprechend sind die Lösungen symmetrisch an beiden Tunnelenden anzuwenden.

 - ★ **L1-(KM-Pe-BdTu2):** Zusätzliches tunnelinternes Protokoll zur Übermittlung der Paketgrenzen. Lösung L1-(KM-Pe-BdTu2)

Lösung
L2-(KM-Pe-BdTu2)

- ▶ **Analoge Lösung:** L1-(KM-Pe-UdTu3).
- ▶ **Zusätzliche Aktivitäten:** Anzuwenden an beiden Tunnelenden.
- ★ **L2-(KM-Pe-BdTu2):** Zusätzliches tunnelexternes Protokoll zur Übermittlung der Paketgrenzen.

Lösung
L3-(KM-Pe-BdTu2)

- ▶ **Analoge Lösung:** L2-(KM-Pe-UdTu3).
- ▶ **Zusätzliche Aktivitäten:** Anzuwenden an beiden Tunnelenden.
- ★ **L3-(KM-Pe-BdTu2):** Reduktion der Eigenschaft des Gesamtsystems.

Lösung
L4-(KM-Pe-BdTu2)

- ▶ **Analoge Lösung:** L3-(KM-Pe-UdTu3).
- ★ **L4-(KM-Pe-BdTu2):** Abbruch des Koppelungsprozesses.
- ▶ **Analoge Lösung:** L4-(KM-Pe-UdTu3).

Fallübersicht

Tabelle 5.7 fasst alle Koppelmuster für das Attribut *Paketgrenze_erhaltend* zusammen und weist jeder Variante eine Fallbezeichnung zu. Die letzte Spalte gibt zusammenfassend Auskunft darüber, ob eine Lösungsmöglichkeit verfügbar ist, die im gekoppelten Gesamtsystem die gewünschten Eigenschaften erhält.

Unidirektional: Quelldienst A → Zieldienst B					
Fallbezeichnung	A: P_e	B: P_e	Aktion notwendig	Lösungen	potent. vollst.
KM-Pe-UdTr1	false	false	-	-	✓
KM-Pe-UdTr2	false	true	-	-	✓
KM-Pe-UdTr3	true	false	ja	L1-(KM-Pe-UdTr3) L2-(KM-Pe-UdTr3)	✗
KM-Pe-UdTr4	true	true	-	-	✓
Unidirektional: Quell- / Zieldienst A ↷ Tunneldienst T					
Fallbezeichnung	A: P_e	T: P_e	Aktion notwendig	Lösungen	potent. vollst.
KM-Pe-UdTu1	false	false	-	-	✓
KM-Pe-UdTu2	false	true	-	-	✓
KM-Pe-UdTu3	true	false	ja	L1-(KM-Pe-UdTu3) L2-(KM-Pe-UdTu3) L3-(KM-Pe-UdTu3) L4-(KM-Pe-UdTu3)	✓
KM-Pe-UdTu4	true	true	-	-	✓
Fortsetzung auf der nächsten Seite					

5.3 VERGLEICH DER STATISCHEN MODELLEILE

Fortsetzung von der vorhergehenden Seite					
Bidirektional: Quelldienst $A \rightarrow$ Zieldienst B					
Fall-bezeichnung	$A: P_e$	$B: P_e$	Aktion notwendig	Lösungen	potent. vollst.
KM-Pe-BdTr1	<i>false</i>	<i>false</i>	-	-	✓
KM-Pe-BdTr2	<i>false</i> <i>true</i>	<i>true</i> <i>false</i>	ja	L1-(KM-Pe-BdTr2) L2-(KM-Pe-BdTr2)	✗
KM-Pe-BdTr3	<i>true</i>	<i>true</i>	-	-	✓
Bidirektional: Quell- / Zieldienst $A \curvearrowright$ Tunneldienst T					
Fall-bezeichnung	$A: P_e$	$T: P_e$	Aktion notwendig	Lösungen	potent. vollst.
KM-Pe-BdTu1	<i>false</i>	<i>false</i>	-	-	✓
KM-Pe-BdTu2	<i>false</i> <i>true</i>	<i>true</i> <i>false</i>	ja	L1-(KM-Pe-BdTu2) L2-(KM-Pe-BdTu2) L3-(KM-Pe-BdTu2) L4-(KM-Pe-BdTu2)	✓
KM-Pe-BdTu3	<i>true</i>	<i>true</i>	-	-	✓

Tabelle 5.7: Koppelungsmuster für Attribut *Paketgrenze_erhaltend*

Teilprozess der Methodik

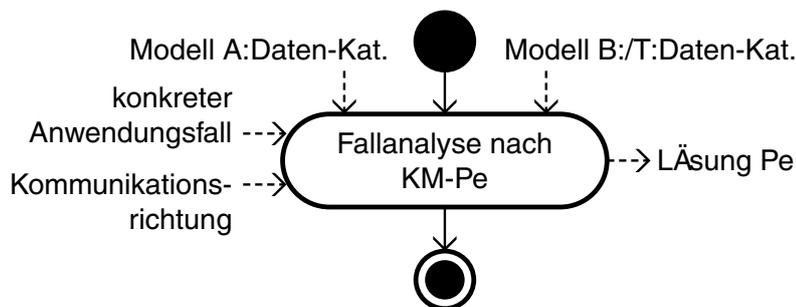


Abbildung 5.13: Aktivitäten zur Koppelung des Attributs *Paketgrenze_erhaltend*

Die für die Koppelung hinsichtlich des Attributs *Paketgrenze_erhaltend* notwendige Aktivität ist in Abbildung 5.13 gezeigt. Sie besteht aus der Fallanalyse durch Tabelle 5.7. Die notwendigen Eingabeartefakte sind zunächst die Paare der Klassen *Daten-Kategorie* und die Kommunikationsrichtung. Bestehen mehrere Lösungsmöglichkeiten, muss noch anhand des konkreten Anwendungsfalls eine individuelle Auswahl der anzuwendenden Lösung getroffen werden.

- I) ... die eine Paarung zu einer Reduzierung der Qualitätseigenschaften am Dienstübergang führt. Diese sei aber mit geringen (technischen) Aufwand zu realisieren und im Einkauf vergleichsweise billig.
- II) ... die andere Paarung gravierend bessere Qualitätssicherungen am Dienstübergang ermöglicht als für die Koppelung notwendig wäre. Dafür sei aber auch ein technisch höherer Aufwand zur Realisierung zu betreiben und der Preis sei deutlich höher als bei der ersten Paarung.

Erst am konkreten Anwendungsfall kann entschieden werden, welche der Paarungen ausgewählt wird. Das zeigen die folgenden Beispiele: Angenommen ...

- ... die Anwendung, für die die Koppelung durchgeführt wird, nutzt den von der Koppelung geforderten Qualitätsrahmen erfahrungsgemäß überhaupt nicht aus. Dann ist Variante I völlig ausreichend.
- ... der Kunde, für dessen Anwendung die Koppelung durchgeführt wird, ist ein Premium-Kunde. In seinem Vertrag sind hohe Strafen für Verletzungen der zugesicherten Qualität vereinbart. Dann ist Variante II zu bevorzugen.
- ... die Anwendung benötigt die bessere Qualität am Dienstübergang, aber der finanzielle Rahmen kann die Kosten von Variante II nicht abdecken. Dann muss die Koppelung scheitern und es muss nach alternativen Diensten gesucht werden.

Wie schon an anderen Stellen der Methodik lassen sich hier keine generischen Vorgehensweisen entwickeln. Die konkreten Entscheidungen müssen individuell von den Gegebenheiten des Anwendungsfalls abhängig gemacht werden.

5.3.2.1 Klasse QoS-Parameter

Weil gemäß der Modellierung zu einer *QoS-Kategorie* mehrerer *QoS-Parameter* gehören können, besteht die Problemstellung zur Koppelung der *QoS-Parameter* wiederum darin, optimale Koppelpaare auszuwählen. Deshalb wird zunächst erneut auf das bereits bekannte allgemeine Aktivitätsmuster aus Unterabschnitt 5.2.3.1 zurückgegriffen. Damit werden iterativ die möglichen Paarungen der *QoS-Parameter* bestimmt.

Der Iterationsschritt ist hier dreigeteilt. Jeder der Schritte beschäftigt sich mit einer der Eigenschaften, die nach dem Modell einen *QoS-Parameter* definieren. Die drei Schritte werden in den folgenden Abschnitten en detail erklärt. Entscheidend ist dabei der erste Schritt, nämlich der Vergleich der *Semantiken*. Nur wenn die *Semantiken* zweier *QoS-Parameter* als gleichwertig eingestuft werden können, dann ist die Ausführung der weiteren Schritte sinnvoll.

Durch die unterschiedliche Mächtigkeit von Netz-QoS-Diensten hinsichtlich der Parameter einer *QoS-Kategorie* ist es in realen Anwendungen durchaus

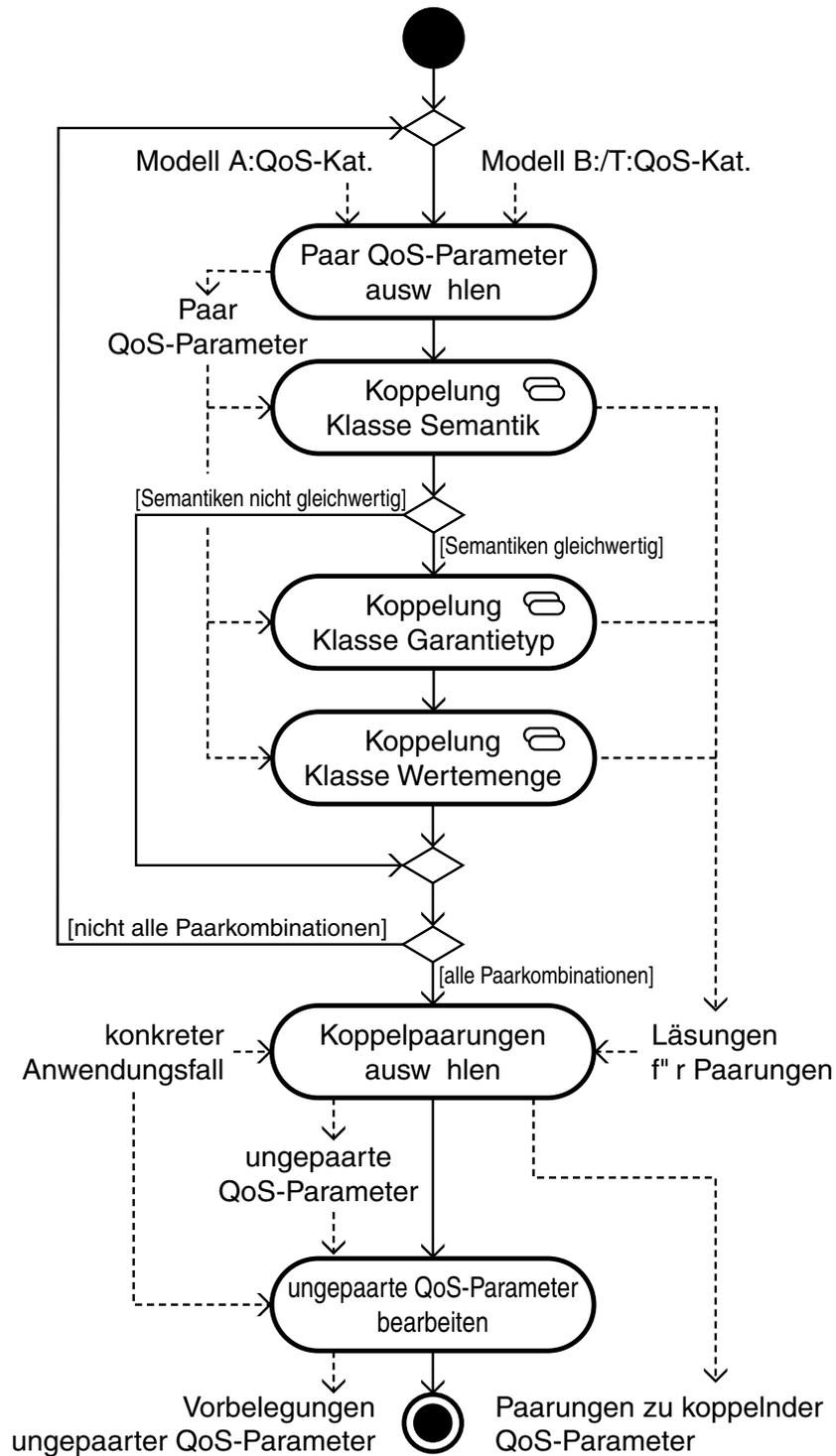


Abbildung 5.15: Aktivitäten zur Koppelung der Klasse *QoS-Parameter*

wahrscheinlich, dass nicht für alle Parameter im zu koppelnden Dienst ein Gegenpart gefunden werden kann. Solche *QoS-Parameter* werden durch den letzten Schritt gesondert betrachtet. Dieser wird in Unterabschnitt 5.3.2.5 besprochen.

5.3.2.2 Klasse Semantik

Für die Gegenüberstellung der *Semantiken* zweier *QoS-Parameter* lassen sich keine allgemeingültigen Muster angeben. Der Grund hierfür liegt in der enormen Spannweite der Techniken, mit der die *Semantik* eines Parameters spezifiziert wird. Häufig werden dafür nur Fließtexte benutzt. Zu deren Vergleich lassen sich überhaupt keine allgemeinen Vorgehensweisen definieren. Wenn die *Semantiken* nach einer formalen Methode spezifiziert werden (siehe auch Abschnitt 3.4.2), wie z.B. [Gars 04], dann ist ein systematischer Vergleich zwar möglich, aber auch wiederum individuell für die eingesetzte Methode. Deshalb werden als Lösungsmöglichkeiten hier nur drei unterschiedliche Ergebnisse des Semantikvergleichs benannt.

- ◆ **KM-QpS-Vgl:** Vergleich zweier *Semantiken* der Klasse *QoS-Parameter*. Fall KM-QpS-Vgl
- ★ **L1-(KM-QpS-Vgl):** Zwei *Semantiken* sind gleichwertig. Lösung
L1-(KM-QpS-Vgl)
 - Ergibt die Analyse, dass zwei *Semantiken* unmittelbar gleichwertig sind, dann wird diese Lösung zugewiesen.
- ★ **L2-(KM-QpS-Vgl):** Zwei *Semantiken* sind nach einfacher Umformung gleichwertig. Lösung
L2-(KM-QpS-Vgl)
 - In vielen Fällen sind die *Semantiken* von *QoS-Parametern* zwar nicht unmittelbar gleichwertig, sie können aber durch eine einfache Umformung in Beziehung zueinander gesetzt werden.
 - Beispielsweise könnten sich zwei *QoS-Parameter* namens „Delay“ gegenüber stehen, von denen der eine ein „Roundtrip Delay“ und der andere ein „Oneway Delay“ ist. Die beiden können zumindest näherungsweise durch die Formel

$$\text{„Oneway Delay“} = \frac{1}{2} \text{„Roundtrip Delay“}$$
 in Beziehung gesetzt werden.
- ★ **L3-(KM-QpS-Vgl):** Zwei *Semantiken* sind nicht vergleichbar. Lösung
L3-(KM-QpS-Vgl)
 - Können zwischen zwei *Semantiken* keine unmittelbaren Beziehungen hergestellt werden, dann bleiben die *QoS-Parameter*, zu denen sie gehören, ungepaart. Um dies für die späteren Methodikschritte zu kennzeichnen, werden beiden Parametern diese virtuelle Lösung zugewiesen.

Fallübersicht

Tabelle 5.8 zeigt eine Übersicht der virtuellen Lösungen. Für die Lösung L3-(KM-QpS-Vgl) kann noch keine Aussage über die potentielle Vollständigkeit der Lösung getroffen werden. Dies entscheidet sich erst im letzten Schritt von Abbildung 5.15, der in Unterabschnitt 5.3.2.5 beschrieben wird.

Klasse <i>Semantik</i> von Klasse <i>QoS-Parameter</i>				
Fallbezeichnung	Relation zw. <i>A:Sem.</i> und <i>B/T:Sem</i>	Akt. notw.	Lösungen	pot. vollst.
KM-QpS-Vgl	gleichwertig	ja	L1-(KM-QpS-Vgl)	✓
KM-QpS-Vgl	gleichwertig nach Umformung	ja	L2-(KM-QpS-Vgl)	✓
KM-QpS-Vgl	nicht vergleichbar	ja	L3-(KM-QpS-Vgl)	-

Tabelle 5.8: Koppelungsmuster für Klasse *Semantik*

Teilprozess der Methodik

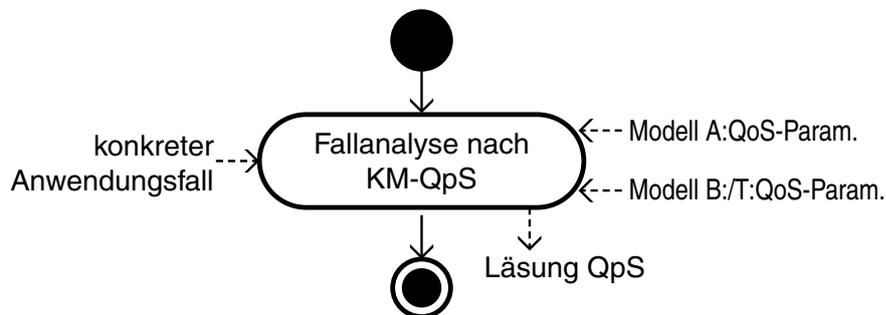


Abbildung 5.16: Aktivitäten zur Koppelung der Klasse *Semantik*

Hinter dem einzigen Schritt im Aktivitätsdiagramm aus Abbildung 5.16 verbirgt sich die individuelle Analyse (wie oben beschrieben) und die Zuweisung der virtuellen Lösungen.

5.3.2.3 Klasse Garantietyp

Beim Vergleich für die Klasse *Garantietyp* sind nur alle nicht-abstrakten Spezialisierungen der Klasse zu betrachten, weil nur diese in einem konkreten Modell für Netz-QoS-Dienste vorkommen können (siehe Unterabschn. 4.3.1.3). Zwischen diesen Klassen ist eine Ordnung nach steigender Garantiegüte gegeben. Somit kann eine Untersuchung nach Mächtigkeit angewendet werden. Nachfolgend geschieht die Untersuchung zunächst nur für die asymmetrische Koppelung. Eine symmetrische Koppelung wird durch den im Anschluss geschilderten Teilprozess der Methodik abgedeckt.

In den Fällen einer Transitiv-Koppelung, in denen der *Garantietyp* stärker oder gleich stark ist, treten keine unlösbaren Probleme auf, da die Ansprüche des vorgeschalteten Dienstes befriedigt werden können. Gleiches gilt für Tunnel-Koppelungen, bei denen der Tunneleinstrom mindestens gleich stark zum Quell- / Zieldienst ist.

Die verbleibenden problematischen Fälle sind:

- ◆ **KM-QpG-Tr3:** Transitiv-Koppelung mit stärkerem *Garantiety* im Quelledienst. Fall KM-QpG-Tr3

Die Problematik in diesem Fall liegt darin, dass der geforderte *Garantiety* vom Zieldienst nicht erbracht werden kann. Es existiert kein Lösungsmuster um dieses Problem verlustfrei zu lösen. Das Gesamtsystem wird immer nur höchstens den *Garantiety* des (schwächeren) Zieldienstes haben.

- ★ **L1-(KM-QpG-Tr3):** Akzeptieren der verminderten Garantiegüte im Gesamtsystem. Lösung L1-(KM-QpG-Tr3)

- ▶ **Voraussetzung:** Keine.
- ▶ **Problempotential:** Dies ist keine Problemlösung im eigentlichen Sinn. Leider ist sie in einigen Fällen der einzige Ausweg um überhaupt eine Koppelung zu realisieren.

- ◆ **KM-QpG-Tu3:** Tunnel-Koppelung mit stärkerem *Garantiety* im Quell- / Zieldienst. Fall KM-QpG-Tu3

Die Problematik ist analog zu [KM-QpG-Tr3](#), da die Garantiegüte im Gesamtsystem durch den Übergang vom Quelledienst zum Tunneldienst reduziert wird.

- ★ **L1-(KM-QpG-Tu3):** Akzeptieren der verminderten Garantiegüte im Gesamtsystem. Lösung L1-(KM-QpG-Tu3)

- ▶ **Analoge Lösung:** [L1-\(KM-QpG-Tr3\)](#).

Fallübersicht

Tabelle 5.9 zeigt einen Überblick über alle Fälle der Koppelung von *Garantiety* eines *QoS-Parameters*.

Transitiv-Koppelung Quelledienst $A \rightarrow$ Zieldienst B				
Fallbezeichnung	Relation	Akt. notw.	Lösungen	potent. vollst.
KM-QpG-Tr1	$A:Gtyp$ schwächer $B:Gtyp$	-	-	✓
KM-QpG-Tr2	$A:Gtyp$ gleich $B:Gtyp$	-	-	✓
KM-QpG-Tr3	$A:Gtyp$ stärker $B:Gtyp$	ja	L1-(KM-QpG-Tr3)	✗
Tunnel-Koppelung Quell- / Zieldienst $A \curvearrowright$ Tunneldienst T				
Fallbezeichnung	Relation	Akt. notw.	Lösungen	potent. vollst.
KM-QpG-Tu1	$A:Gtyp$ schwächer $T:Gtyp$	-	-	✓
KM-QpG-Tu2	$A:Gtyp$ gleich $T:Gtyp$	-	-	✓
KM-QpG-Tu3	$A:Gtyp$ stärker $T:Gtyp$	ja	L1-(KM-QpG-Tu3)	✗

Tabelle 5.9: Koppelungsmuster für Klasse *Garantiety*

Teilprozess der Methodik

Die Aktivitäten zur Koppelung der Klasse *Garantietyt* sind in Abbildung 5.17

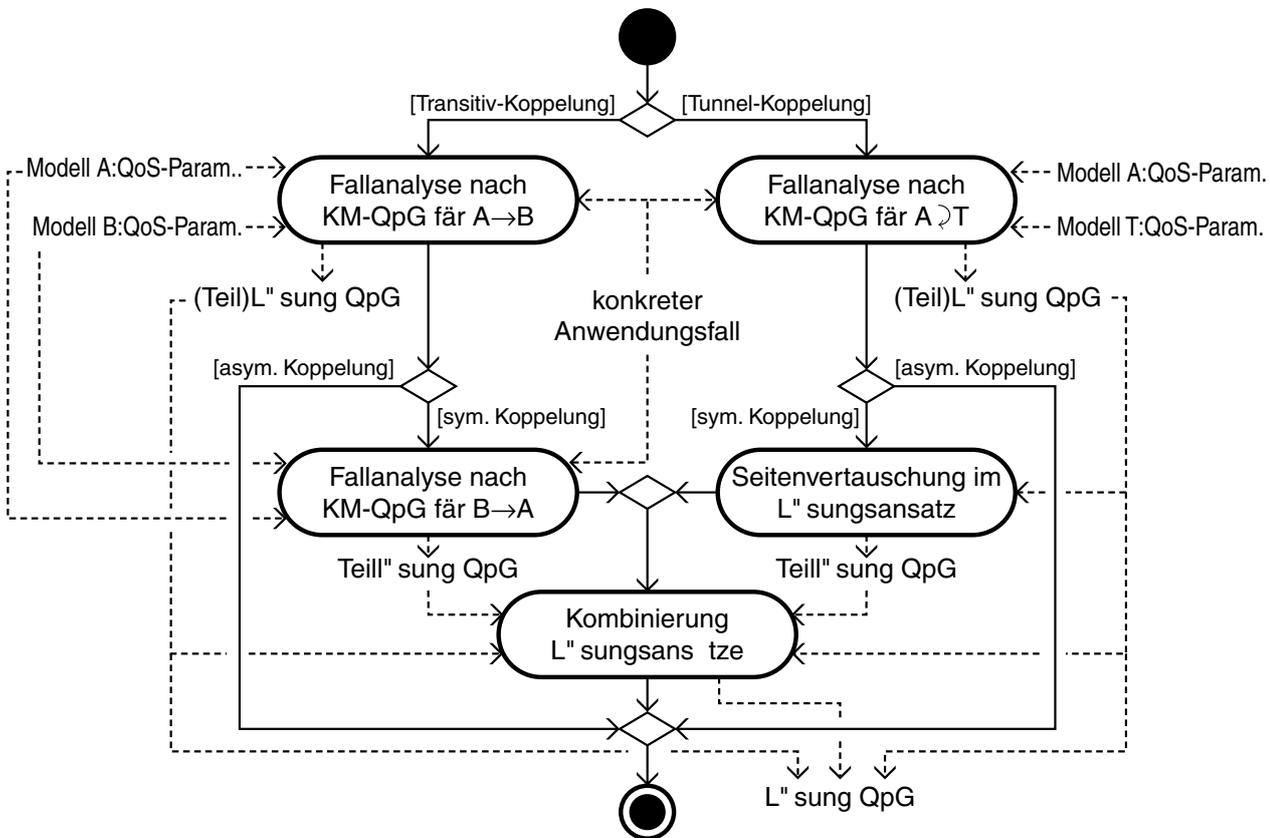


Abbildung 5.17: Aktivitäten zur Koppelung der Klasse *Garantietyt*

abgebildet. Der Teilprozess ist eine Spezialisierung des Musters aus Unterabschnitt 5.2.3.2. Der symmetrische Koppelungsfall wird dadurch nun ebenfalls abgedeckt.

5.3.2.4 Klasse Wertemenge

Wenn die *Semantiken* zweier *QoS-Parameter* als wenigstens „gleichwertig durch Umformung“ klassifiziert wurden, sprich die Lösung *L1-(KM-QpS-Vgl)* oder *L2-(KM-QpS-Vgl)* gefunden wurde, dann können auch deren *Wertemengen* sinnvoll miteinander verglichen werden.

Teilprozess der Methodik

Bevor der eigentliche Vergleich der *Wertemengen* erfolgen kann, sind zuerst noch zwei wichtige Schritte zu tun, wie das Aktivitätsdiagramm aus Abbildung 5.18 zeigt.

Im ersten Schritt werden die Lösungen der beiden vorhergehenden Teilprozesse benötigt (Lösungen QpS und Lösungen QpG). Es gibt zwei mögliche Gründe warum Modifikationen vorgenommen werden müssen:

5.3 VERGLEICH DER STATISCHEN MODELLETEILE

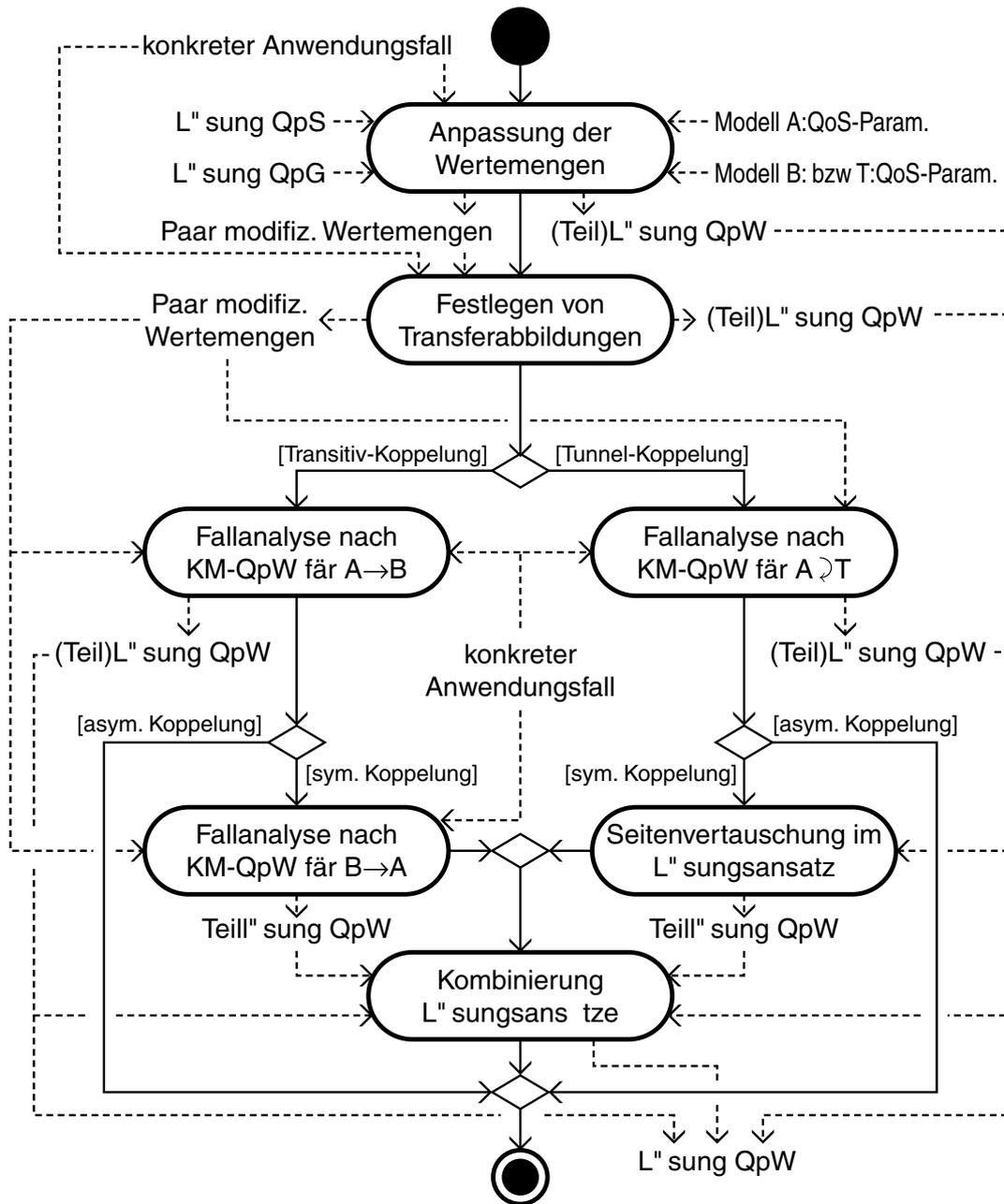


Abbildung 5.18: Aktivitäten zur Koppelung der Klasse *Wertemenge*

- ◆ **KM-QpW-Semmod:** Herstellung der semantischen Gleichwertigkeit durch Umformung. Fall KM-QpW-Semmod

Für Paarungen, für die die Lösung [L2-\(KM-QpS-Vgl\)](#) gewählt wurde, also die Herstellung der semantischen Gleichwertigkeit durch Umformung, muss die gefundene Umformung für die *Wertemenge* analog nachvollzogen werden.

Für das Beispiel der beiden Varianten „Roundtrip Delay“ und „Oneway Delay“ aus Unterabschnitt 5.3.2.2 würde das bedeuten, dass entweder die Elemente der Wertemenge von

„Roundtrip Delay“ halbiert oder die Elemente der Wertemenge von „Oneway Delay“ verdoppelt werden müssen.

Lösung L1-(KM-QpW-Semmod)

- ★ **L1-(KM-QpW-Semmod):** Anpassung der Klasse *Wertemenge* I.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Problempotential:** Dies ist keine Problemlösung im eigentlichen Sinn. Leider in einigen Fällen der einzige Ausweg um überhaupt eine Koppelung zu realisieren.

Fall
KM-QpW-Typmod

- ◆ **KM-QpW-Typmod:** Paarung mit unterschiedlich starken *Garantietypen*.
Für Paarungen mit unterschiedlich starken *Garantietypen* (Fall KM-QpG-Tr1, KM-QpG-Tr3, KM-QpG-Tu1 oder KM-QpG-Tu3) kann in manchen Fällen die Wertemenge des „starken“ Parameters so eingeschränkt werden, dass sich eine bessere Entsprechung zur Wertemenge des „schwachen“ Parameters ergibt.

Bei Koppelung der Klasse *Loss* aus DiffServ AF PHB (Abb. 4.14, S. 95) mit dem ATM-Parameter *CLR* (Abb. 4.17, S. 101) kann die Wertemenge des „stärkeren“ *CLR* eingeschränkt werden. Hierfür könnten im DiffServ-basierten Dienst numerische Werte für *low*, *medium* und *high* empirisch ermittelt werden. Anschließend würde man die Werte von *CLR* auf drei enge Intervalle einschränken, die jeweils einen der ermittelten DiffServ-Werte einschließen.

Lösung L1-(KM-QpW-Typmod)

- ★ **L1-(KM-QpW-Typmod):** Anpassung der Klasse *Wertemenge* II.
 - ▶ **Analoge Lösung:** L1-(KM-QpW-Semmod).

Das Ergebnis der Aktivität ist zum einen ein Paar modifizierter *Wertemengen*. Zum anderen müssen die individuell gemachten Anpassungen noch in einer Lösung mit der Bezeichnung L1-(KM-QpW-Semmod) bzw. L1-(KM-QpW-Typmod) dokumentiert werden (Tab. 5.10).

Anpassungen der Wertemengen				
Fallbezeichnung	Anlass	Akt. notw.	Lösungen	pot. vollst.
KM-QpW-Semmod	L2-(KM-QpS-Vgl)	ja	L1-(KM-QpW-Semmod)	✓
KM-QpW-Typmod	KM-QpG-Tr1 KM-QpG-Tr3 KM-QpG-Tu1 KM-QpG-Tu3	ja	L1-(KM-QpW-Typmod)	✓

Tabelle 5.10: Koppelungsmuster für Anpassungen der Klasse *Wertemenge*

Der zweite Schritt im Teilprozess für *Wertemengen* beschäftigt sich mit der Abbildung der Werte zwischen den Diensten. In einigen Fällen genügt es

nämlich nicht, dass ein Netz-QoS-Dienst, der einen bestimmten Wert eines *QoS-Parameters* benötigt, den gleichen Wert bei einem nachgeschalteten Netz-QoS-Dienst anfordert. Das zeigt das folgende Beispiel:

Ein Unternehmen betreibt einen Netz-QoS-Dienst *A*. Dieser wird von der Anwendung „Voice-over-IP“ genutzt. Der «Geschwindigkeitsparameter» *Verzögerung* des Netz-QoS-Dienstes *A* ist dafür auf eine obere Schranke von *200ms* eingestellt. Ein zweites Unternehmen hat seine Telefonie auch auf Voice-over-IP umgestellt und nutzt dafür den Netz-QoS-Dienst *B*. Beide Unternehmen kooperieren nun und koppeln deshalb ihre Telefonie-Netze.

Wenn nun beim Übergang von *A* nach *B* der Wert für die *Verzögerung* ohne Abbildung im Dienst *B* angefordert wird, dann steigt die Gesamtverzögerung im gekoppelten System im schlechtesten Fall auf *400ms* an. Aus Sicht der VoIP-Anwendung wird damit der «Geschwindigkeitsparameter» *Verzögerung* verletzt. Stattdessen muss Dienst *A* die *Verzögerung* intern auf *100ms* limitieren und darf von Dienst *B* ebenfalls höchstens *100ms* anfordern. Dann kann im Gesamtsystem die Schranke von *200ms* eingehalten werden.

Neben der Verzögerung wird auch eine minimale *Bandbreite* garantiert. Bei diesem «Volumenparameter» wäre eine Aufteilung zwischen den Diensten grundfalsch. Hier muss zwischen *A* und *B* natürlich der benötigte Wert unverändert angefordert werden.

Das Beispiel zeigt, dass ...

- ... nicht alle Parameter beim Dienstübergang abgebildet werden müssen bzw. dürfen.
- ... Abbildungen nicht nur zwischen den Diensten erforderlich sein können, sondern dass möglicherweise auch die interne Behandlung von *Wertemengen* eines *QoS-Parameters* angepasst werden muss.

Durch das Beispiel wird auch klar, dass die Fragen, welche Parameter eine Abbildung durchlaufen müssen und wie diese Abbildung dann konkret auszusehen hat, nur individuell am konkreten Anwendungsfall beantwortet werden kann. Allgemein lässt sich nur sagen: Wenn ein Parameter das Merkmal aufweist, dass sich seine Werte „kumulativ“ verhalten, dann muss überprüft werden, ob Abbildungen notwendig sind.

Zwar kann kein detailliertes Vorgehen im Sinne einer Schritt-für-Schritt-Anleitung gegeben werden, aber aus den in Unterabschnitt 4.3.1.3 eingeführten Unterarten von *QoS-Parametern* lassen sich zumindest Hinweise ablesen:

- *Volumenparameter*

Volumenparameter verhalten sich nicht kumulativ. Bei ihnen muss nur sichergestellt werden, dass die geforderten Werte durchgängig erreicht werden.

KAPITEL 5: GENERISCHE KOPPELUNGSMETHODIK FÜR NETZ-QoS-DIENSTE

■ *Geschwindigkeitsparameter*

Geschwindigkeitsparameter sind typische Kandidaten, bei denen sowohl interne Anpassungen und Transferabbildungen notwendig werden. Die klassischen Vertreter *Verzögerung* und *Verzögerungsvarianz* verhalten sich beide kumulativ.

■ *Korrektheitsparameter*

Für die hier häufig vorkommenden booleschen *Wertemengen* sind keine Abbildungen notwendig. Bei statistischen Werten muss eine Abbildung individuell überprüft werden.

Weitere Einflussfaktoren auf die Notwendigkeit und insbesondere die Art der Abbildung sind beispielsweise:

- Konkreter Anwendungsfall
- *Semantik des QoS-Parameters*
- *Garantiertyp des QoS-Parameters* (Auf Abbildungsvorschrift ist insbesondere bei *statistischen Garantien* zu achten.)
- Koppelsymmetrie
- Bisher erarbeitete Teillösungen aus anderen Teilprozessen

Um bei Abarbeitung der Methodik zu dokumentieren, dass Abbildungen notwendig sind, sind folgende Fall- / Lösungsbezeichnungen zu verwenden:

Fall KM-QpW-Trans

◆ **KM-QpW-Trans:** Abbildung der *Wertemengen*.

Lösung
L1-(KM-QpW-Trans)

★ **L1-(KM-QpW-Trans):** Interne Anpassung der *Wertemenge*.

▶ **Voraussetzung:** Keine.

Lösung
L2-(KM-QpW-Trans)

★ **L2-(KM-QpW-Trans):** Abbildung der *Wertemenge* am Dienstübergang.

▶ **Voraussetzung:** Keine.

Festlegen von Transferabbildungen				
Fall-bezeichnung	Anwendung	Aktion notwendig	Lösungen	potent. vollst.
KM-QpW-Trans	intern	ja	L1-(KM-QpW-Trans)	✓
	Dienstübergang	ja	L2-(KM-QpW-Trans)	✓

Tabelle 5.11: Koppelungsmuster für Abbildungen der Klasse *Wertemenge*

Nachdem in den beiden vorangegangenen Schritten die *Wertemengen* den notwendigen Transformationen unterzogen worden sind, können nun die modifizierten *Wertemengen* miteinander verglichen werden um die bei der Koppelung weiteren notwendigen Aktionen zu bestimmen. Nachfolgend ge-

schiebt die Untersuchung nur für die asymmetrische Koppelung. Eine symmetrische Koppelung wird durch das bereits bekannte Muster aus Unterabschnitt 5.2.3.2 abgedeckt, das im Teilprozess in Abbildung 5.18 bereits integriert ist.

- ◆ **KM-QpW-Tr1:** Transitiv-Koppelung: *Wertemenge* des Quelldienstes ist echte Teilmenge der *Wertemenge* des Zieldienstes. Fall KM-QpW-Tr1

Es gibt kein Problem, da der abhängige Dienst mächtiger ist als der Quelldienst.

- ◆ **KM-QpW-Tr2:** Transitiv-Koppelung: *Wertemenge* des Quelldienstes ist gleich der *Wertemenge* des Zieldienstes. Fall KM-QpW-Tr2

Es gibt kein Problem, da beide *Wertemengen* identisch sind.

- ◆ **KM-QpW-Tr3:** Transitiv-Koppelung: *Wertemenge* des Zieldienstes ist echte Teilmenge der *Wertemenge* des Quelldienstes. Fall KM-QpW-Tr3

In diesem Fall entsteht das Problem, dass der abhängige Dienst nur einen Teil der Werte des Quelldienstes abbilden kann. Die Folge ist immer eine eingeschränkte Wertemenge des Gesamtsystems im Vergleich zum Quelldienst.

- ★ **L1-(KM-QpW-Tr3):** Akzeptieren der verminderten Garantiegüte im Gesamtsystem. Lösung
L1-(KM-QpW-Tr3)

- ▶ **Voraussetzung:** Keine.

- ▶ **Problempotential:** Dies ist keine Problemlösung im eigentlichen Sinn. Leider ist sie in einigen Fällen der einzige Ausweg um überhaupt eine Koppelung zu realisieren.

- ◆ **KM-QpW-Tr4:** Transitiv-Koppelung: *Wertemengen* von Quell- und Zieldienst sind disjunkt. Fall KM-QpW-Tr4

Kann kein einziges gemeinsames Element zwischen den *Wertemengen* gefunden werden, dann scheitert die Koppelung der beiden *QoS-Parameter*.

- ★ **L1-(KM-QpW-Tr4):** Abbruch des Teilprozesses. Lösung
L1-(KM-QpW-Tr4)

- ▶ **Voraussetzung:** Keine.

- ▶ **Lösungsworkflow:** Abbrechen des Teilprozesses.

- ▶ **Problempotential:** Koppelung des Parameter-Paares kommt nicht zustande.

Die einzelnen Fälle einer Tunnel-Koppelung verhalten sich analog zu denen einer Transitiv-Koppelung. Der einzige Unterschied besteht darin, dass bei einer Tunnel-Koppelung eventuell noch die aktuelle Wertebelegung aus dem Quelldienst hin zum Zieldienst signalisiert werden kann. Notwendig ist dies nur, wenn die ursprüngliche Wertebelegung des Quelldienstes durch die Wertebelegung im Tunneldienst nicht nachvollzogen werden kann (analog z.B. zu den Fällen KM-Pe-*).

- Fall KM-QpW-TuSig ♦ **KM-QpW-TuSig:** Signalisieren der tatsächlichen Wertebelegung im Quelldienst hin zum Zieldienst.
Die prinzipiellen Signalisierungsmöglichkeiten sind analog zu denen vom [KM-Vo-Tu4](#).
- Lösung L1-(KM-QpW-TuSig) ★ **L1-(KM-QpW-TuSig):** Wertesignalisierung indirekt durch Werte des Tunneldienstes.
▶ **Analoge Lösung:** [L1-\(KM-Vo-Tu4\)](#).
- Lösung L2-(KM-QpW-TuSig) ★ **L2-(KM-QpW-TuSig):** Zusätzliche tunnelinterne Signalisierung für Werteübermittlung.
▶ **Analoge Lösung:** [L2-\(KM-Vo-Tu4\)](#).
- Lösung L3-(KM-QpW-TuSig) ★ **L3-(KM-QpW-TuSig):** Zusätzliche tunnelexterne Signalisierung für Werteübermittlung.
▶ **Analoge Lösung:** [L3-\(KM-Vo-Tu4\)](#).
- Fall KM-QpW-Tu1 ♦ **KM-QpW-Tu1:** Tunnel-Koppelung: *Wertemenge* des Quelldienstes ist echte Teilmenge der *Wertemenge* des Tunneldienstes.
Lösung L1-(KM-QpW-Tu1) ★ **L1-(KM-QpW-Tu1):** Wertesignalisierung vom Quell- zum Zieldienst.
▶ **Analoge Lösung:** [L1-\(KM-QpW-TuSig\)](#) oder [L2-\(KM-QpW-TuSig\)](#) oder [L2-\(KM-QpW-TuSig\)](#).
- Fall KM-QpW-Tu2 ♦ **KM-QpW-Tu2:** Tunnel-Koppelung: *Wertemenge* des Quelldienstes ist gleich der *Wertemenge* des Tunneldienstes.
Lösung L1-(KM-QpW-Tu2) ★ **L1-(KM-QpW-Tu2):** Wertesignalisierung vom Quell- zum Zieldienst.
▶ **Analoge Lösung:** [L1-\(KM-QpW-TuSig\)](#) oder [L2-\(KM-QpW-TuSig\)](#) oder [L2-\(KM-QpW-TuSig\)](#).
- Fall KM-QpW-Tu3 ♦ **KM-QpW-Tu3:** Tunnel-Koppelung: *Wertemenge* des Tunneldienstes ist echte Teilmenge der *Wertemenge* des Quelldienstes.
Lösung L1-(KM-QpW-Tu3) ★ **L1-(KM-QpW-Tu3):** Akzeptieren der verminderten Garantiegüte im Gesamtsystem und Wertesignalisierung vom Quell- zum Zieldienst.
▶ **Analoge Lösung:** [L1-\(KM-QpW-Tr3\)](#).
▶ **Zusätzliche Aktivitäten:** Muss mit einer der Lösungen [L1-\(KM-QpW-TuSig\)](#), [L2-\(KM-QpW-TuSig\)](#) oder [L3-\(KM-QpW-TuSig\)](#) kombiniert werden.
- Fall KM-QpW-Tu4 ♦ **KM-QpW-Tu4:** Tunnel-Koppelung: *Wertemengen* von Quell- und Tunneldienst haben keine gemeinsamen Elemente.
Lösung L1-(KM-QpW-Tu4) ★ **L1-(KM-QpW-Tu4):** Abbruch des Teilprozesses.
▶ **Analoge Lösung:** [L1-\(KM-QpW-Tr3\)](#).

Die nachfolgende Tabelle [5.12](#) gibt einen Überblick über die eben diskutierten Fälle:

5.3 VERGLEICH DER STATISCHEN MODELLETEILE

Transitiv-Koppelung Quelldienst $A \rightarrow$ Zieldienst B				
Fallbezeichnung	Mengenrelation	Akt. notw.	Lösungen	pot. vollst.
KM-QpW-Tr1	$A:Wertem. \subset B:Wertem.$	-	-	✓
KM-QpW-Tr2	$A:Wertem. = B:Wertem.$	-	-	✓
KM-QpW-Tr3	$A:Wertem. \supset B:Wertem.$	ja	L1-(KM-QpW-Tr3)	✗
KM-QpW-Tr4	$A:Wertem. \cap B:Wertem. = \emptyset$	ja	L1-(KM-QpW-Tr4)	✗
Tunnel-Koppelung Quell- / Zieldienst $A \supset$ Tunneldienst T				
Fallbezeichnung	Relation	Akt. notw.	Lösungen	pot. vollst.
KM-QpW-Tu1	$A:Wertem. \subset T:Wertem.$	ja	L1-(KM-QpW-Tu1)	✓
KM-QpW-Tu2	$A:Wertem. = T:Wertem.$	ja	L1-(KM-QpW-Tu2)	✓
KM-QpW-Tu3	$A:Wertem. \supset T:Wertem.$	ja	L1-(KM-QpW-Tu3)	✗
KM-QpW-Tu4	$A:Wertem. \cap T:Wertem. = \emptyset$	ja	L1-(KM-QpW-Tu4)	✗

Tabelle 5.12: Koppelungsmuster für Klasse *Garantiertyp*

Damit ist der letzten Block des Teilprozesses aus Abbildung 5.18 abgeschlossen.

5.3.2.5 Behandlung ungepaarter QoS-Parameter

Bei der Koppelung der *QoS-Parameter* einer *QoS-Kategorie* kann es vorkommen, dass für einen *QoS-Parameter* des einen Dienstes keine Entsprechung im anderen Dienst gefunden werden konnte. Dieser Umstand bedeutet aber nicht sofort, dass die Koppelung scheitert oder dass das Gesamtsystem nur mit reduzierten Eigenschaften realisiert werden kann.

Stammt ein ungepaarter *QoS-Parameter* nämlich aus einem nachgeschalteten Dienst, dann bedeutet dies, dass der Parameter für das Gesamtsystem (eigentlich) nicht benötigt wird. Da sich der Parameter aber nicht einfach „wegdefinieren“ lässt, kommt als Problemlösung nur eine „sinnvolle“ Vorbelegung des Parameters in Frage. „Sinnvoll“ lässt sich dabei nur anhand des konkreten Anwendungsfalls definieren.

Bei der Abarbeitung der Methodik für eine Tunnel-Koppelung zur Verbindung zweier VoIP-Standorte ist der «Geschwindigkeitsparameter» *Verzögerung* im Tunneldienst ungepaart geblieben. Im Quell- / Zieldienst wird nur der «Volumenparameter» *Bandbreite* spezifiziert. Anhand der Anwendung VoIP kann im Tunneldienst der Parameter *Verzögerung* auf einen für Telefonie sinnvollen Wert von maximal *200ms* vorbelegt werden.

Ist der ungepaarte *QoS-Parameter* aber Teil des Quelldienstes, so heißt das, dass eine Eigenschaft des Quelldienstes durch den nachgeschalteten Dienst nicht befriedigt werden kann. Wird die Koppelung zwischen den Diensten durchgeführt, dann verliert das Gesamtsystem die durch den *QoS-Parameter* eigentlich zugesicherte Eigenschaft.

Unterscheidet man explizit die Grundanwendungsfälle, ergibt sich die folgende Fallunterscheidung für eine asymmetrische Koppelung:

Fall KM-QpUp-ATr1 ♦ **KM-QpUp-ATr1:** Asymmetrische Transitiv-Koppelung, ungepaarter *QoS-Parameter* aus dem Quelldienst.

Lösung
L1-(KM-QpUp-ATr1)

- ★ **L1-(KM-QpUp-ATr1):** Abbruch des Teilprozesses.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Abbrechen des Teilprozesses.
 - ▶ **Problempotential:** Koppelung des Parameters kommt nicht zustande.

Fall KM-QpUp-ATr2 ♦ **KM-QpUp-ATr2:** Asymmetrische Transitiv-Koppelung, ungepaarter *QoS-Parameter* aus dem Zieldienst.

Lösung
L1-(KM-QpUp-ATr2)

- ★ **L1-(KM-QpUp-ATr2):** Vorbelegen des *QoS-Parameters*.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Bereitstellungsphase: Ermitteln eines Vorbelegungswertes.
 - ▶ **Problempotential:** Vorbelegungswert könnte schwierig abzuschätzen sein.

Fall KM-QpUp-ATu1 ♦ **KM-QpUp-ATu1:** Asymmetrische Tunnel-Koppelung, ungepaarter *QoS-Parameter* aus dem Quell- / Zieldienst.

Lösung
L1-(KM-QpUp-ATu1)

- ★ **L1-(KM-QpUp-ATu1):** Abbruch des Teilprozesses.
 - ▶ **Voraussetzung:** Keine.
 - ▶ **Lösungsworkflow:** Abbrechen des Teilprozesses.
 - ▶ **Problempotential:** Koppelung des Parameters kommt nicht zustande.

Fall KM-QpUp-ATu2 ♦ **KM-QpUp-ATu2:** Asymmetrische Tunnel-Koppelung, ungepaarter *QoS-Parameter* aus dem Tunneldienst.

Lösung
L1-(KM-QpUp-ATu2)

- ★ **L1-(KM-QpUp-ATu2):** Vorbelegen des *QoS-Parameters*.
 - ▶ **Analoge Lösung:** L1-(KM-QpUp-ATr2).

Fall KM-QpUp-ATr1/2/Tu1/2 ♦ **KM-QpUp-ATr1/2/Tu1/2:** Symmetrische Koppelung, Grundanwendungsfall und Seite des ungepaarten *QoS-Parameters* beliebig.

Bei einer symmetrischen Koppelung tritt der ungepaarte *QoS-Parameter* immer einmal auf der Seite des Quelldienstes auf. Eine Vorbelegung ist daher nicht möglich:

★ **L1-(KM-QpUp-ATr1/2/Tu1/2):** Abbruch des Teilprozesses.

Lösung L1-(KM-QpUp-ATr1/2/Tu1/2)

▶ **Analoge Lösung:** L1-(KM-QpUp-ATr1).

Fallübersicht

Tabelle 5.13 stellt alle Fälle für ungepaarte *QoS-Parameter* nochmals zusammen.

Asymmetrische Koppelung: Dienst des ungepaarten <i>QoS-Parameters</i>					
Fallbezeichnung	Anw.-Fall	Dienst	Akt. notw.	Lösungen	pot. vollst.
KM-QpUp-ATr1	Trans.	Quelld.	ja	L1-(KM-QpUp-ATr1)	✗
KM-QpUp-ATr2	Trans.	Zield.	ja	L1-(KM-QpUp-ATr2)	✓
KM-QpUp-ATu1	Tunnel	Quell/ Zield.	ja	L1-(KM-QpUp-ATu1)	✗
KM-QpUp-ATu2	Tunnel	Tunneld.	ja	L1-(KM-QpUp-ATu2)	✓
Symmetrische Koppelung: Dienst des ungepaarten <i>QoS-Parameters</i>					
Fallbezeichnung	Anw.-Fall	Dienst	Akt. notw.	Lösungen	pot. vollst.
KM-QpUp-STr1	Trans.	Quelld.	ja	L1-(KM-QpUp-ATr1/2/Tu1/2)	✗
KM-QpUp-STr2	Trans.	Zield.	ja	L1-(KM-QpUp-ATr1/2/Tu1/2)	✗
KM-QpUp-STu1	Tunnel	Quell/ Zield.	ja	L1-(KM-QpUp-ATr1/2/Tu1/2)	✗
KM-QpUp-STu2	Tunnel	Tunneld.	ja	L1-(KM-QpUp-ATr1/2/Tu1/2)	✗

Tabelle 5.13: Koppelungsmuster für ungepaarte *QoS-Parameter*

Teilprozess der Methodik

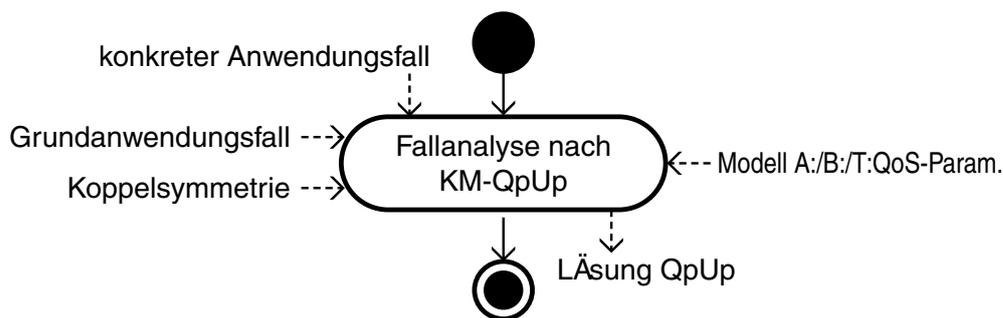


Abbildung 5.19: Aktivitäten zur Behandlung ungepaarter *QoS-Parameter*

Durch die Zusammenfassung der beiden Eingabeartefakte Grundanwendungsfall und Koppelsymmetrie in einer Falluntersuchung ist im Teilprozess von Abbildung 5.19 nur eine Aktion notwendig.

5.4 VERGLEICH DER DYNAMISCHEN MODELLTEILE

Mit dem Vergleich der statischen Modellteile wurden in der vorangegangenen Sektion bereits eine große Anzahl von Teilproblemen analysiert und verschiedene Möglichkeiten zu deren Lösung aufgezeigt. In dieser Sektion wird der Ablauf aus Abbildung 5.1, Seite 136 mit dem Vergleich der dynamischen Modellteile fortgesetzt.

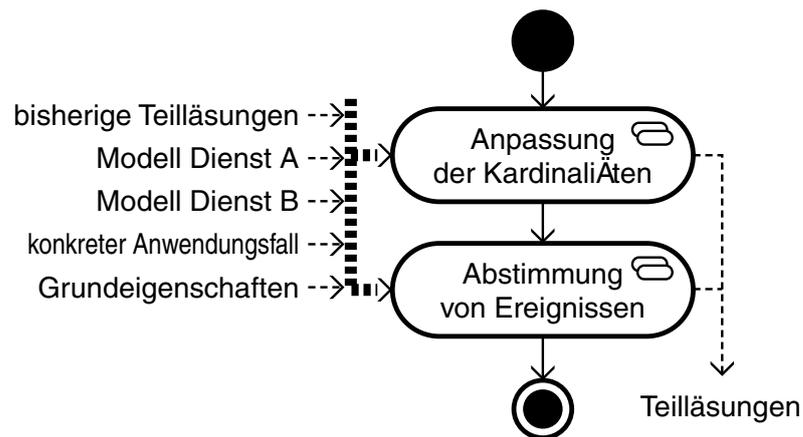


Abbildung 5.20: Aktivitäten zum Vergleich dynamischer Modellteile

Das Aktivitätsdiagramm in Abbildung 5.20 zeigt die beiden Hauptschritte zum Vergleich der dynamischen Modellteile. Die erste Teilaktivität wird sich mit der Anpassung unterschiedlicher Kardinalitäten der Klassen *Daten-Kategorie* und *QoS-Kategorie* beschäftigen. In der zweiten Teilaktivität werden dann Maßnahmen zum Abgleich einzelner Ereignisabläufe in den einzelnen Klassen besprochen.

5.4.1 Anpassung der Kardinalitäten

Wie im Rahmen der Erstellung der Modellierung in Abschnitt 4.3.3 festgestellt, unterscheiden sich Netz-QoS-Dienste auch in der Anzahl der Instanzen der Klassen *Daten-Kategorie* und *QoS-Kategorie*, die zur Betriebsphase erzeugt werden.

In diesem Teilprozess, der in Abbildung 5.21 dargestellt ist, wird eine Anpassung der Kardinalitäten aus den Modellen – sprich eine Anpassung der Instanzenanzahl – für *Daten-Kategorien* und *QoS-Kategorien* vorgenommen.

Andere Mengenunterschiede zwischen anderen Klassen von konkreten Netz-QoS-Diensten werden bereits durch die iterative Paarsuche in den diversen Teilprozessen behandelt:

- Unterschiedlich viele Ausprägungen von *Daten-Kategorien* in konkreten Modellen: Teilprozess in Abbildung 5.10, Seite 154

- Unterschiedlich viele Ausprägungen von *QoS-Kategorien* in konkreten Modellen: Teilprozess in Abbildung 5.14, Seite 172
- Unterschiedlich viele *QoS-Parameter* zweier *QoS-Kategorien* in verschiedenen Modellen: Teilprozess in Abbildung 5.15, Seite 174

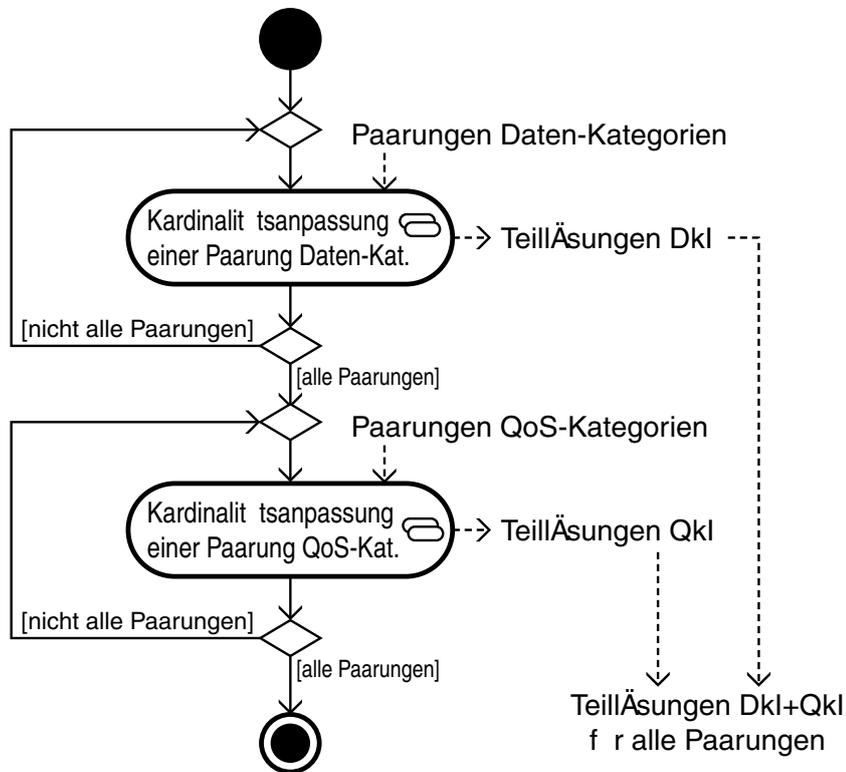


Abbildung 5.21: Aktivitäten zur Anpassung der Kardinalitäten

Im Prozess aus Abbildung 5.21 werden nacheinander die Anpassungen nur noch für jene Paarungen vorgenommen, für die in den vorangegangenen Teilprozessen festgestellt wurde, dass sie sich zur Koppelung eignen. Im Folgenden werden die Aktivitäten beschrieben, die für eine einzelne Paarung auszuführen sind.

In diesem Abschnitt konzentriert sich die Analyse auf die in Unterabschnitt 4.3.3.2 aufgelisteten Varianten. Analysiert man dabei nur die Anzahl der Instanzen von jeweils *Daten-Kategorien* und *QoS-Kategorien*, dann erkennt man, dass es sich jeweils um genau eine Instanz (d.h. ein Singleton) oder um beliebig viele Instanzen handelt.

Singleton und beliebig viele Instanzen verhalten sich analog zu zustandslos und zustandsbehaftet. Für die Koppelung wurden alle dabei auftretenden Fälle bereits allgemein in Abschnitt 5.2.4 behandelt. Zu erledigen verbleibt deren Interpretation für *Daten-Kategorien* und *QoS-Kategorien*.

Für *Daten-Kategorien* wurden die Fälle bereits interpretiert, nämlich bei der Bearbeitung des Attributs *Verbindungsorientiert*. Das bedeutet: Wenn wenigstens eine der *Daten-Kategorien* aus einem Paar mit *Verbindungsorien-*

tiert==true vorliegt, dann ist für die Kardinalitätsanpassung nichts mehr zu tun, da die Teillösungen aus Tabelle 5.2 dies bereits implizit erledigt haben.

Liegen dagegen *Daten-Kategorien* mit *Verbindungsorientiert==false* vor, dann müssen neue Teillösungen bearbeitet werden. Diese Variante deckt dann noch den seltenen Fall ab, dass zwar verbindungslose *Daten-Kategorien* vorliegen, von diesen aber trotzdem Instanzen erzeugt werden müssen. Diese verhalten sich aber völlig analog zum Fall mit *Verbindungsorientiert==true*. Nachfolgend geschieht die Untersuchung zunächst nur für die asymmetrische Koppelung. Eine symmetrische Koppelung wird durch den im Anschluss geschilderten Teilprozess der Methodik abgedeckt.

Fall KM-Dkl-Tr1 ♦ **KM-Dkl-Tr1:** Transitiv-Koppelung mit Singleton-*Daten-Kategorie* → Singleton-*Daten-Kategorie*.

Keine Probleme, da beide *Daten-Kategorien* nur genau einmal instanziiert werden.

Fall KM-Dkl-Tr2 ♦ **KM-Dkl-Tr2:** Transitiv-Koppelung mit Singleton-*Daten-Kategorie* → Mehrfachinstanz-*Daten-Kategorie*.

Beim Übergang von einer Singleton- zu einer Mehrfachinstanz-*Daten-Kategorie* entsteht das Problem, dass im Zieldienst vor Übertragungsbeginn eine Instanz erzeugt werden muss.

Lösung
L1-(KM-Dkl-Tr2)

★ **L1-(KM-Dkl-Tr2):** Globale, statische Instanz.

▶ **Analoge Lösung:** L1-(KM-Vo-Tr2).

Lösung
L2-(KM-Dkl-Tr2)

★ **L2-(KM-Dkl-Tr2):** Dynamische Instanzerzeugung.

▶ **Analoge Lösung:** L2-(KM-Vo-Tr2).

Fall KM-Dkl-Tr3 ♦ **KM-Dkl-Tr3:** Transitiv-Koppelung mit Mehrfachinstanz-*Daten-Kategorie* → verbindungslos.

Problemlos, weil keine Instanzen im Zieldienst erzeugt werden müssen.

Fall KM-Dkl-Tr4 ♦ **KM-Dkl-Tr4:** Transitiv-Koppelung mit Mehrfachinstanz-*Daten-Kategorie* → Mehrfachinstanz-*Daten-Kategorie*.

Liegt eine Koppelung vor, bei der in Quell- und Zieldienst jeweils mehrere Instanzen von *Daten-Kategorien* verwendet werden, dann müssen die Instanzen im Zieldienst zu dem Zeitpunkt erzeugt werden, zu dem im Quelldienst die Instanzen erzeugt werden.

Lösung
L1-(KM-Dkl-Tr4)

★ **L1-(KM-Dkl-Tr4):** Dynamische Instanziiierung.

▶ **Analoge Lösung:** L1-(KM-Vo-Tr4).

Fall KM-Dkl-Tu1 ♦ **KM-Dkl-Tu1:** Tunnel-Koppelung mit Singleton-*Daten-Kategorie* ↷ Singleton-*Daten-Kategorie*.

Analog zu **KM-Dkl-Tr1** tritt hier kein Problem auf, da alle Dienste Singleton-*Daten-Kategorien* verwenden.

- ◆ **KM-Dkl-Tu2:** Tunnel-Koppelung mit Singleton-*Daten-Kategorie* ↷ Mehrfachinstanz-*Daten-Kategorie*. Fall KM-Dkl-Tu2

Am Tunneleingang liegt der Fall **KM-Dkl-Tr2** vor. Der Tunnelausgang entspricht Fall **KM-Dkl-Tr3**, bei dem nichts zu tun ist. Da zwischen Quell- und Zieldienst keine Relation zwischen Instanzen von *Daten-Kategorien* hergestellt werden muss, ist das zu lösende Problem nur analog zu **KM-Dkl-Tr2**, da **KM-Dkl-Tr3** problemlos ist.

- ★ **L1-(KM-Dkl-Tu2):** Globale, statische Instanz im Tunneldienst. Lösung L1-(KM-Dkl-Tu2)
- ▶ **Analoge Lösung:** **L1-(KM-Dkl-Tr2)**, zwischen Quell- und Tunneldienst.
- ★ **L2-(KM-Dkl-Tu2):** Dynamische Instanziierung im Tunneldienst. Lösung L2-(KM-Dkl-Tu2)
- ▶ **Analoge Lösung:** **L2-(KM-Dkl-Tr2)**, zwischen Quell- und Tunneldienst.

- ◆ **KM-Dkl-Tu3:** Tunnel-Koppelung mit Mehrfachinstanz-*Daten-Kategorie* ↷ Singleton-*Daten-Kategorie*. Fall KM-Dkl-Tu3

Das Problem in diesem Fall besteht in der gleichzeitigen Instanzerzeugung in Quell- und Zieldienst über einen Tunneldienst hinweg, der nur genau eine Instanz benutzt. Ein Rückgriff auf den Fall **KM-Dkl-Tr3** am Tunneleingang ist nicht sinnvoll, da dabei die Signalisierung für die Instanziierung im Zieldienst fehlt.

- ★ **L1-(KM-Dkl-Tu3):** Zusätzliche tunnelinterne Signalisierung für Instanziierung. Lösung L1-(KM-Dkl-Tu3)
- ▶ **Voraussetzung:** Tunnelinterne Übertragungsmöglichkeit für Signalisierung.
 - ▶ **Lösungskomponenten:**
 - Sender am Tunneleingang, der Instanziierung signalisiert.
 - Empfänger am Tunnelausgang, der Instanziierung im Zieldienst vornimmt.
 - Signalisierungsprotokoll zwischen Sender und Empfänger, das den gleichen Tunnel benutzt, wie die Datenübertragung selbst.
 - ▶ **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger und Signalisierungsprotokoll.
 - ▶ **Problempotential:** Beeinflussung des QoS der regulären Datenübertragung durch zusätzliches Signalisierungsprotokoll, da die Übertragung im gleichen Tunnel erfolgt.
- ★ **L2-(KM-Dkl-Tu3):** Zusätzliche tunnelexterne Signalisierung für Instanziierung. Lösung L2-(KM-Dkl-Tu3)
- ▶ **Voraussetzung:** Tunnelexterne Übertragungsmöglichkeit für Signalisierung.

► **Lösungskomponenten:**

- Sender am Tunneleingang, der Instanziierung signalisiert.
- Empfänger am Tunnelausgang, der Instanziierung im Zieldienst vornimmt.
- Signalisierungsprotokoll zwischen Sender und Empfänger, das nicht den gleichen Tunnel benutzt.
- Zusätzlicher Übertragungsweg für Verbindungssignalisierung.

► **Lösungsworkflow:** Bereitstellungsphase: Erstellen und Installieren von Sender, Empfänger, Signalisierungsprotokoll und Übertragungsweg für Instanziierungssignalisierung.

Fall KM-Dkl-Tu4 ◆ **KM-Dkl-Tu4:** Tunnel-Koppelung mit Mehrfachinstanz-*Daten-Kategorie* ↷ Mehrfachinstanz-*Daten-Kategorie*.

Die Problemstellung ist identisch zu [KM-Dkl-Tu3](#). Allerdings ergibt sich dadurch dass der Tunneldienst selbst mehrfache Instanzen von *Daten-Kategorien* benutzt eine weitere Lösungsmöglichkeit.

Lösung
L1-(KM-Dkl-Tu4)

★ **L1-(KM-Dkl-Tu4):** Instanziierungssignalisierung indirekt durch Instanz des Tunneldienstes.

► **Voraussetzung:** Wechselseitig kompatible Attributierung der *Daten-Kategorien* zwischen Quell- / Zieldienst und Tunneldienst.

► **Lösungskomponenten:** Zwei Umsetzungskomponenten nach [L1-\(KM-Dkl-Tr4\)](#) an den Übergängen Quelledienst → Tunneldienst und Tunneldienst → Zieldienst.

► **Lösungsworkflow:** Analog zu [L1-\(KM-Dkl-Tr4\)](#).

Lösung
L2-(KM-Dkl-Tu4)

★ **L2-(KM-Dkl-Tu4):** Zusätzliche tunnelinterne Signalisierung für Instanziierung.

► **Analoge Lösung:** [L1-\(KM-Dkl-Tu3\)](#).

Lösung
L3-(KM-Dkl-Tu4)

★ **L3-(KM-Dkl-Tu4):** Zusätzliche tunnelexterne Signalisierung für Instanziierung.

► **Analoge Lösung:** [L2-\(KM-Dkl-Tu3\)](#).

Für die Anpassung der Klasse *QoS-Kategorie* sind die Teilfälle und deren Lösungen völlig analog, weil es sich um ein strukturell absolut analoges Problem handelt. Auf eine explizite Aufzählung der Teilfälle und Lösungsmöglichkeiten wird deshalb verzichtet. Im Namensschema sind statt KM-Dkl-* bei den *QoS-Kategorie* die Bezeichnungen KM-Qkl-* zu verwenden.

Die Tabellen [5.14](#) und [5.15](#) fassen alle Fälle zusammen:

5.4 VERGLEICH DER DYNAMISCHEN MODELLEILE

Transitiv-Koppelung Quelldienst → Zieldienst					
Fall- bezeichnung	Kardinalität		Aktion notwendig	Lösungen	pot. vollst.
	A:D.-Kat.	B:D.-Kat.			
KM-Dkl-Tr1	Singl.	Singl.	-	-	✓
KM-Dkl-Tr2	Singl.	Mult.	ja	L1-(KM-Dkl-Tr2) L2-(KM-Dkl-Tr2)	✓
KM-Dkl-Tr3	Mult.	Singl.	-	-	✓
KM-Dkl-Tr4	Mult.	Mult.	ja	L1-(KM-Dkl-Tr4)	✓
Tunnel-Koppelung Quell- / Zieldienst ↷ Tunneldienst					
Fall- bezeichnung	Kardinalität		Aktion notwendig	Lösungen	pot. vollst.
	A:D.-Kat.	T:D.-Kat.			
KM-Dkl-Tu1	Singl.	Sing.	-	-	✓
KM-Dkl-Tu2	Singl.	Mult.	ja	L1-(KM-Dkl-Tu2) L2-(KM-Dkl-Tu2)	✓
KM-Dkl-Tu3	Mult.	Singl.	ja	L1-(KM-Dkl-Tu3) L2-(KM-Dkl-Tu3)	✓
KM-Dkl-Tu4	Mult.	Mult.	ja	L1-(KM-Dkl-Tu4) L2-(KM-Dkl-Tu4) L3-(KM-Dkl-Tu4)	✓

Tabelle 5.14: Koppelungsmuster für Kardinalitäten der Klasse *Daten-Kategorie*

Transitiv-Koppelung Quelldienst → Zieldienst					
Fall- bezeichnung	Kardinalität		Aktion notwendig	Lösungen	pot. vollst.
	A:QoS-Kat.	B:QoS-Kat.			
KM-Qkl-Tr1	Singl.	Singl.	-	-	✓
KM-Qkl-Tr2	Singl.	Mult.	ja	L1-(KM-Qkl-Tr2) L2-(KM-Qkl-Tr2)	✓
KM-Qkl-Tr3	Mult.	Singl.	-	-	✓
KM-Qkl-Tr4	Mult.	Mult.	ja	L1-(KM-Qkl-Tr4)	✓
Tunnel-Koppelung Quell- / Zieldienst ↷ Tunneldienst					
Fall- bezeichnung	Kardinalität		Aktion notwendig	Lösungen	pot. vollst.
	A:QoS-Kat.	T:QoS-Kat.			
KM-Qkl-Tu1	Singl.	Sing.	-	-	✓
KM-Qkl-Tu2	Singl.	Mult.	ja	L1-(KM-Qkl-Tu2) L2-(KM-Qkl-Tu2)	✓
KM-Qkl-Tu3	Mult.	Singl.	ja	L1-(KM-Qkl-Tu3) L2-(KM-Qkl-Tu3)	✓

Fortsetzung auf der nächsten Seite

Fortsetzung von der vorhergehenden Seite					
KM-Qkl-Tu4	Mult.	Mult.	ja	L1-(KM-Qkl-Tu4) L2-(KM-Qkl-Tu4) L3-(KM-Qkl-Tu4)	✓

Tabelle 5.15: Koppelungsmuster für Kardinalitäten der Klasse *QoS-Kategorie*

Teilprozess der Methodik

In Abbildung 5.22 ist der Teilprozess zur Anpassung der Kardinalitäten von *Daten-Kategorien* dargestellt. Zur Behandlung einer symmetrischen Koppelung wird wieder auf das bereits bekannte Muster aus Unterabschnitt 5.2.3.2 zurückgegriffen. Der analoge Prozess für *QoS-Kategorien* wird nicht explizit dargestellt.

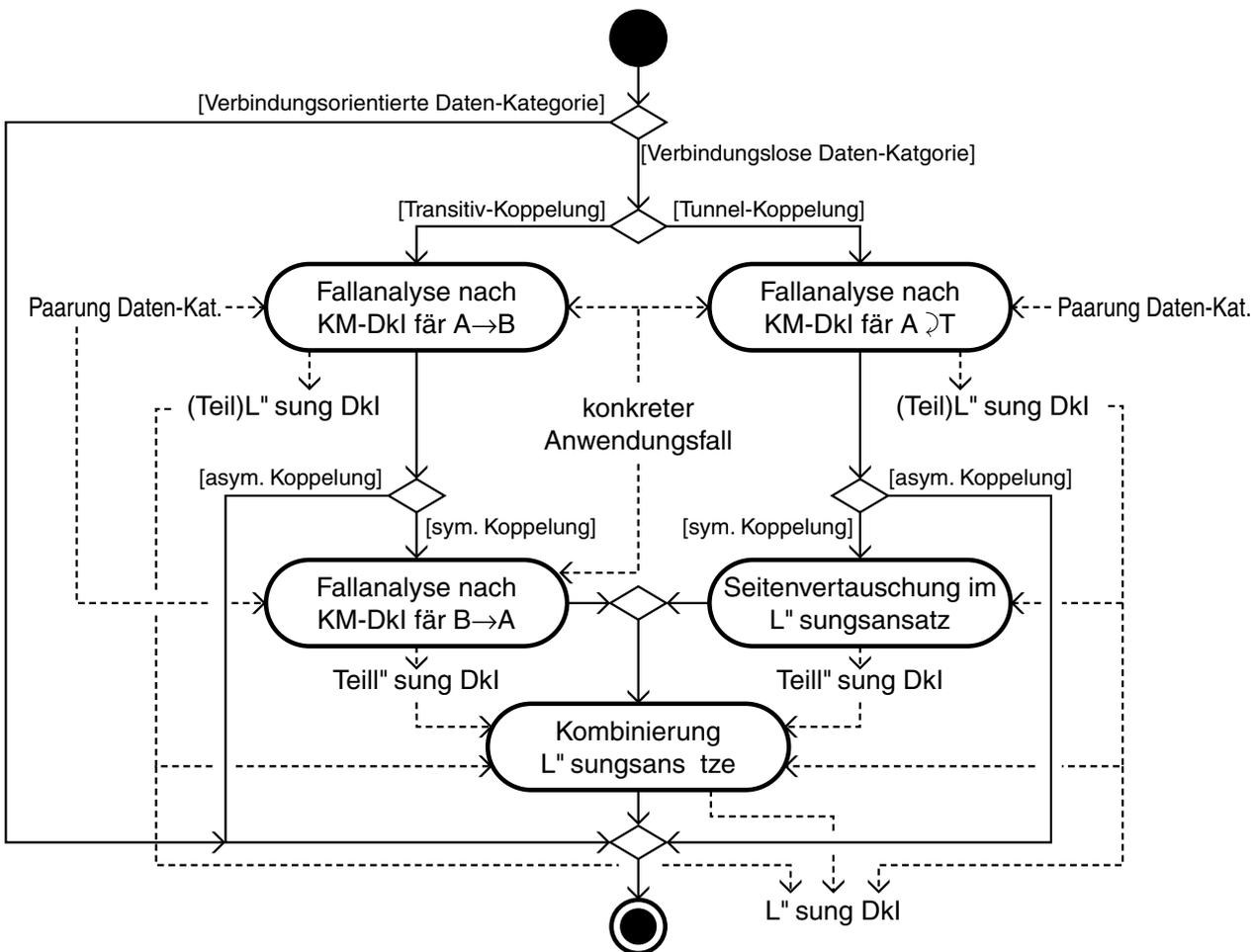


Abbildung 5.22: Aktivitäten zur Anpassung der Kardinalitäten von *Daten-Kategorien*

Jetzt sind alle Mengenunterschiede, sowohl solche auf Ebene der konkreten Modellteile als auch auf der Ebene der Instanzen in der Betriebsphase, durch die Methodik behandelt worden.

5.4.2 Abstimmung von Ereignissen

Zum Abschluss der Koppelung der dynamischen Teile müssen noch die Ereignisse aller Klassen der jeweiligen Dienste aufeinander abgestimmt werden.

Zu betrachten sind hier nicht alle Kombinationen, sondern nur jene Paarungen, die bereits zur Koppelung in den vorangegangenen Teilprozessen ausgewählt worden sind. Konkret besteht der Teilprozess (Abb. 5.23) aus folgenden Teilen:

- Ereigniskoppelung der Paarungen von *Daten-Kategorien* und alle assoziierten Klassen
- Ereigniskoppelung der Paarungen von *QoS-Kategorien*
- Ereigniskoppelung der Paarungen von *QoS-Parametern* und alle assoziierten Klassen

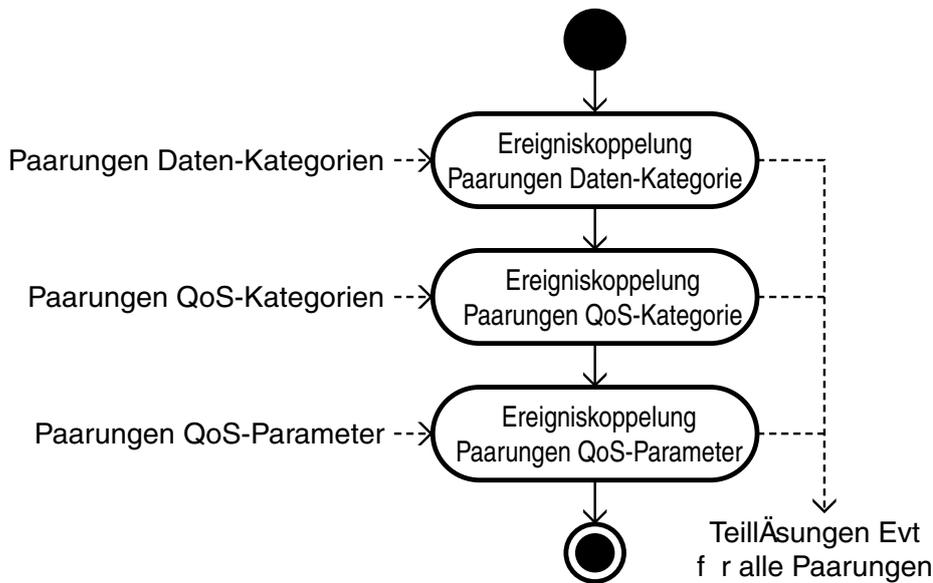


Abbildung 5.23: Aktivitäten zur Abstimmung von Ereignissen

Bei der Gegenüberstellung der einzelnen Ereignisdiagramme ist, wie schon bei deren Modellierung, die Grenze der expliziten generischen Behandlung erreicht. Ebenso wenig wie für die verschiedenen Semantiken und optionalen Daten von Ereignissen ein generisches, aber trotzdem aussagekräftiges Modell gegeben werden konnte, können nun keine allgemeine Abbildungsvorschriften explizit angegeben werden. An dieser Stelle ist also massiv die menschliche Intelligenz gefordert. Eine detailreiche Methodik, die das Potential für ein maschinell automatisiertes Verfahren in sich birgt, kann leider nicht angegeben werden.

Allerdings lassen sich einige Fragen und Schemen nennen, die bei der Abbildung zwischen den Ereignisdiagrammen helfen können:

■ Überprüfung, ob ein Ereignis bereits implizit behandelt wird

Möglicherweise sind manche Ereignisse durch eine der bereits vorher ermittelten Teillösungen abgedeckt. Ein klassisches Beispiel:

Für eine *Daten-Kategorie* mit *Verbindungsorientiert==true* sei in dem Ereignisdiagramm der Auf- und Abbau einer Verbindung als explizite Ereignisse modelliert. Diese müssen nicht mehr behandelt werden, da diese durch eine Lösung aus der Fallanalyse KM-Vo-* (Unterabschn. 5.3.1.1) bereits abgedeckt sind.

■ Abbildung von Spontanereignissen

Spontanereignisse (Unterabschn. 4.3.4.3) sind in der Regel zwischen den Diensten abzubilden. Spontanereignisse haben ihre Wurzel häufig im Fehlermanagement; mit ihnen werden Fehlerzustände signalisiert. Für eine korrekte Funktion des Gesamtsystems sind sie meist abzubilden. Allerdings gibt es auch Ausnahmen.

Beispielsweise sendet die ATM-QoS-Kategorie *ABR_Kat* zur Anzeige einer Veränderung in den aktuellen QoS-Parametern ein Ereignis. Da bei *ABR_Kat* aber keine Garantien gegeben werden, hat dieses Spontanereignis nur informellen Charakter. Kann es nicht abgebildet werden, führt das nicht zu einer Einschränkung im gekoppelten System.

■ Abbildung von Interaktionsereignissen

Bei den von einem Dienst angebotenen Interaktionsereignissen (Unterabschn. 4.3.4.3) muss unterschieden werden zwischen solchen, ...

- ▼ ... die für die korrekte Funktion des Dienstes benötigt werden, beispielsweise das Ereignis, das den Aufbau einer Verbindung auslöst.
- ▼ ... die optional ausgelöst werden können. Ein Ereignis, das einen *QoS-Parameter* nachträglich verändert, wäre ein Beispiel.

■ Tunnelung von Ereignissen

Liegt der Fall einer Tunnel-Koppelung vor, dann muss bei jedem ausgehenden Ereignis des Quelldienstes überprüft werden, ob es genügt das Ereignis zum Tunneldienst zu übermitteln. Möglicherweise ist es nämlich notwendig, dass das Ereignis über den Tunneldienst hinweg auch zum Zieldienst gesandt werden muss. Wenn dies der Fall ist, sind [L1-\(KM-gZ-Tu4\)](#), [L2-\(KM-gZ-Tu4\)](#) und [L3-\(KM-gZ-Tu4\)](#) Kandidaten für Lösungsmuster, die entsprechend adaptiert angewendet werden können.

■ Einschränkung durch unvollständige oder fehlende Abbildung

Kann ein Ereignis des einen Dienstes nur unvollständig oder gar nicht auf den anderen Dienst abgebildet werden, muss individuell entschieden werden, ob dadurch die Leistung des Gesamtsystems eingeschränkt wird. Pauschal lassen sich keine Aussagen treffen.

5.4 VERGLEICH DER DYNAMISCHEN MODELLETEILE

Bei einer Tunnel-Koppelung bietet der Tunneldienst ein Interaktionsereignis zur Veränderung der Einstellungen der *QoS-Parameter*. Der Quell- / Zieldienst bietet aber überhaupt keine Möglichkeit *QoS-Parameter* zu verändern, weil er z.B. auf einer CoS-Architektur beruht. Hier ist es unkritisch für das Gesamtsystem, wenn das Ereignis des Tunneldienstes nicht abgebildet wird.

Der Tunneldienst kann aber auch ein Spontanereignis aussenden, das anzeigt, dass keine weiteren Daten mehr übertragen werden können. Wenn dieses Ereignis nicht auf den Quell- / Zieldienst übertragen werden kann, dann ist dies fatal für die Gesamtfunktionalität. Es wäre nämlich ein Abbruch der Datenübertragung möglich, ohne dass dies im Quell- / Zieldienst bemerkt werden würde.

Bei der Gegenüberstellung der Ereignisse eines Zeitkorridors ist noch eine weitere Problematik zu beachten: Die zu verbindenden Ereignisse müssen auch in ihrer zeitlichen Lage „zueinander passen“. „Passend“ bedeutet hier, dass die beiden Ereignisse im gleichen Abschnitt im Lebenszyklus der Koppelung liegen müssen. Dies muss nicht der Fall sein, denn in den beiden zu koppelnden Diensten kann der gleiche Korridor in den Diensten jeweils unterschiedlich im Lebenszyklus angesiedelt sein. Zwei prinzipielle Fälle können unterschieden werden:

- Die beiden Korridore überschneiden sich

Dann entsteht kein Problem, da es immer eine gemeinsame Lebenszyklusphase zwischen den Korridoren gibt.

- Die beiden Korridore überschneiden sich nicht

Kann zwischen zwei Korridoren keine zeitliche Überschneidung hergestellt werden, dann bedeutet das immer, dass eine Lebenszyklusgrenze zwischen den Korridoren liegt². Damit liegen die Korridore in verschiedenen Lebenszyklusphasen. Diesem Umstand ist bei der Verbindung der Ereignisse individuell Rechnung zu tragen.

Nachfolgende Fallbezeichnungen sind für die gefundenen Teilfälle zu verwenden:

- ◆ **KM-Evt-class:** Abbildung von Ereignis(sen) der Klasse *class*.

Fall KM-Evt-class

²Dies ist deswegen der Fall, weil die Lebenszyklusgrenzen die einzigen absoluten Zeitmarken zur Einordnung der Korridore sind.

5.5 ABSCHLIESSENDE SCHRITTE DER KOPPELUNG

Mit dem Abschluss der Koppelung der statischen und der dynamischen Teile der Nutzfunktionalität kann die gesamte Koppelung der beiden Netz-QoS-Dienste *A* und *B* abgeschlossen werden. Abbildung 5.24 zeigt die verbleibenden Schritte.

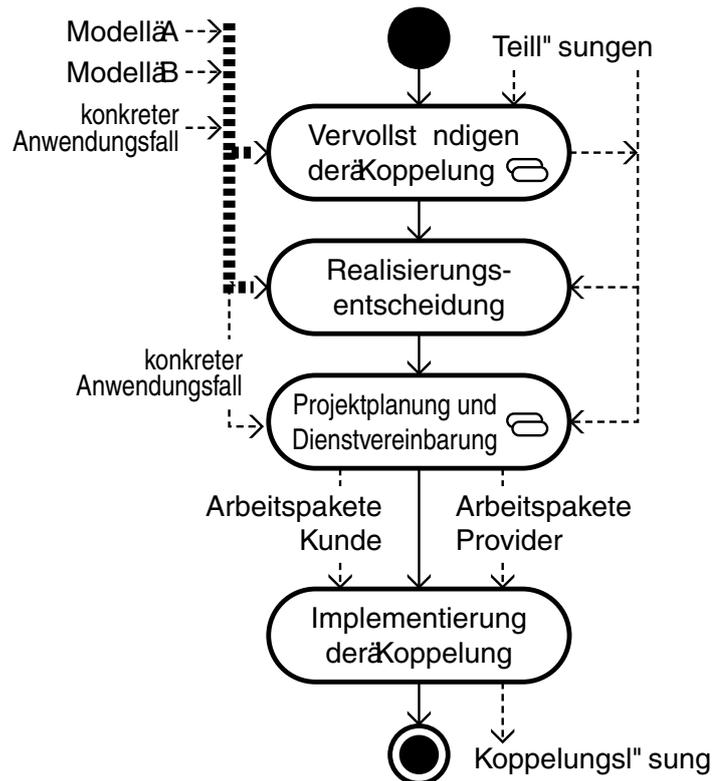


Abbildung 5.24: Aktivitäten zum Abschluss der Methodik

- Vervollständigung Der unmittelbar nächste Schritt in Abschnitt 5.5.1 vervollständigt den Vergleich der verbleibenden Teile aus der Dienstmodellierung von Sektion 4.4.
- Realisierungsentscheidung Nachdem alle Teile aus dem Dienstmodell abgearbeitet sind, kann in Abschnitt 5.5.2 final darüber entschieden werden, ob die Koppelung durchgeführt werden soll.
- Projektplanung Nach einer positiven Entscheidung können konkrete Projektplanungen zwischen Kunde und Provider zur Umsetzung der Koppelung beginnen. Das Ergebnis sind Arbeitspakete, die von beiden Seiten abzarbeiten sind. Zuvor werden die getroffenen Vereinbarungen formal in einer Dienstvereinbarung festgehalten (Abschn. 5.5.3).
- Dienstvereinbarung
- Implementierung Im letzten Schritt kann die Koppelungslösung auf beiden Seiten implementiert werden (Abschn. 5.5.4).

5.5.1 Vervollständigung der Dienstkoppelung

Im Dienstmodell aus Abbildung 4.30, Seite 129 sind bis jetzt drei Teile unbehandelt geblieben, nämlich der *service access point*, die gesamte Dienstmanagementfunktionalität und das *service agreement*. Die beiden erstgenannten Teile werden in diesem Abschnitt behandelt (gezeigt in Abb. 5.25), dem *service agreement* wird sich Unterabschnitt 5.5.3.3 widmen.

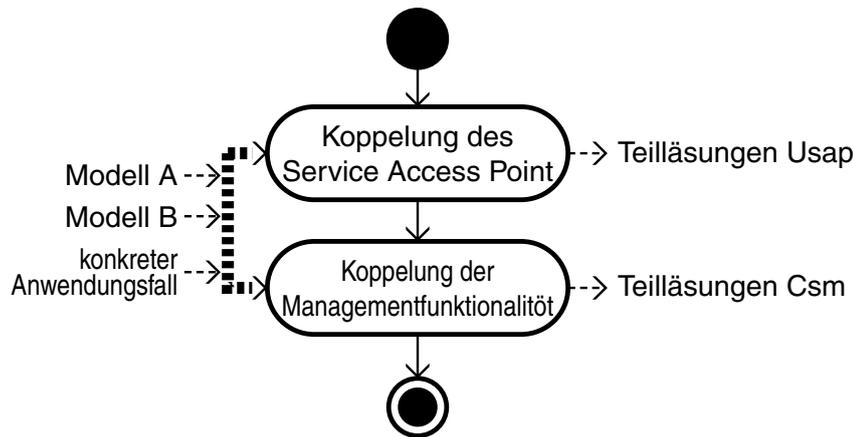


Abbildung 5.25: Aktivitäten zur Vervollständigung der Koppelung

5.5.1.1 Koppelung der Service Access Points

Im Bereich der Nutzfunktionalität des Dienstmodells verbleibt noch die Koppelung des *service client* an den *service access point*.

Die beiden Hauptaufgaben, die hier zu erledigen sind, sind zum einen die Umsetzung konkreter Datenformate, zum anderen gilt es technisch einen Anschluss an den Übergabepunkt zu realisieren. Dadurch, dass die Beschreibung des *service access point* selbst sehr individuell ausgelegt ist, kann die Koppelung auch nur individuell gestaltet werden. Nachfolgende Fall- / Lösungsbezeichnungen sind für die Dokumentation zu verwenden:

Umsetzung von Datenformaten, Anschluss an Übergabepunkt

◆ **KM-Usap-***: Koppelung an den *service access point* der Nutzfunktionalität.

Fall KM-Usap-*

★ **L1-(KM-Usap-*)**: Individuelle Ankoppelung.

Lösung L1-(KM-Usap-*)

Muss individuell erstellt werden.

5.5.1.2 Koppelung der Managementfunktionalität

Zur Koppelung der Managementfunktionalität der Dienste wird, wie schon bei deren Modellierung, wieder auf die Arbeit [Nerb 01] von Nerb zurückgegriffen. Neben der Modellierung von CSM-Schnittstellen enthält diese Arbeit auch eine Methodik, wie CSM-Schnittstellen im konkreten Anwendungsfall

Koppelung Managementfunktionalität mit Nerb

zu realisieren sind. Diese Methodik kommt hier zum Tragen. Für die dort entwickelten Lösungen sind nachfolgende Fall- / Lösungsbezeichnungen zu verwenden:

- Fall KM-Csm-* ◆ **KM-Csm-***: Koppelung der Dienstmanagementseite.
- Lösung ★ **L1-(KM-Csm-*)**: Individuelle Ankoppelung.
L1-(KM-Csm-*) Muss individuell nach [Nerb 01] erstellt werden.

5.5.2 Realisierungsentscheidung

Mit dem vorangegangenen Teilprozess ist die Erarbeitung der Teillösungen zur Koppelung abgeschlossen. Zwei wichtige Punkte sind somit fixiert:

- Der gesamte Aufwand, der benötigt wird um die Koppelung durchzuführen.
- Die Gesamteigenschaften, die das gekoppelte Gesamtsystem erhalten wird.

Entscheidung am Anwendungsfall Anhand dieser beiden Punkte kann nun entschieden werden, ob die Koppelung sinnvoll ist und auch realisiert werden soll. Diese Entscheidung kann natürlich wieder nur vor dem Hintergrund des konkreten Anwendungsfalls getroffen werden.

Sinnhaftigkeit Um die Sinnhaftigkeit beurteilen zu können, sollten alle Teillösungen beachtet werden, die Einschränkungen im Gesamtsystem mit sich bringen. Die Gesamtheit aller Einschränkungen muss dann dem konkreten Anwendungsfall gegenübergestellt werden. So kann dann entschieden werden, ob die Koppelung überhaupt sinnvoll ist.

Problempotentiale von Teillösungen Zusätzlich sollten auch die Problempotentiale der Teillösungen untersucht werden. Möglicherweise bringen sie zukünftige Probleme mit sich, die sich negativ auf die strategische Ausrichtung von Kunde oder Provider auswirken. Auch vor diesem Hintergrund kann eine Koppelung als nicht sinnvoll eingestuft werden.

Aufwandsbewertung Wird die Koppelung als sinnvoll angesehen, dann kann anhand der Lösungsworkflows und der Lösungskomponenten der Aufwand bewertet werden, den eine Realisierung der Koppelung mit sich bringen würde. Hier müssen wirtschaftliche Überlegungen zusammen mit dem konkreten Anwendungsfall angestellt werden. Wird auch der zu betreibende Aufwand akzeptiert, dann kann die Koppelung schließlich realisiert werden.

Wenn die Realisierungsentscheidung negativ ausgefallen ist, dann bleiben im Wesentlichen zwei Möglichkeiten:

- Auswählen eines anderen Dienstes
- Modifikation eines oder beider Dienste

Durch Zurückverfolgen der Problempunkte über jene Lösungsmöglichkeiten, die zur negativen Entscheidung geführt haben, erhält man direkte Hinweise darüber, worin sich ein anderer Dienst unterscheiden muss. Soll ein Dienst modifiziert werden, dann geben die Definitionskorridore der einzelnen Netz-QoS-Architekturen Hinweise darüber, wo Modifikationen vorgenommen werden könnten.

Methodik zeigt Problempunkte auf
Definitionskorridore zeigen Modifikationspunkte

5.5.3 Projektplanung und Dienstvereinbarung

Wenn entschieden worden ist, dass die Koppelung durchgeführt werden soll, dann tritt der Lebenszyklus der Koppelung in die nächste Phase ein. Die Planungsphase ist beendet, es beginnt die Verhandlungsphase.

Eintritt in Verhandlungsphase

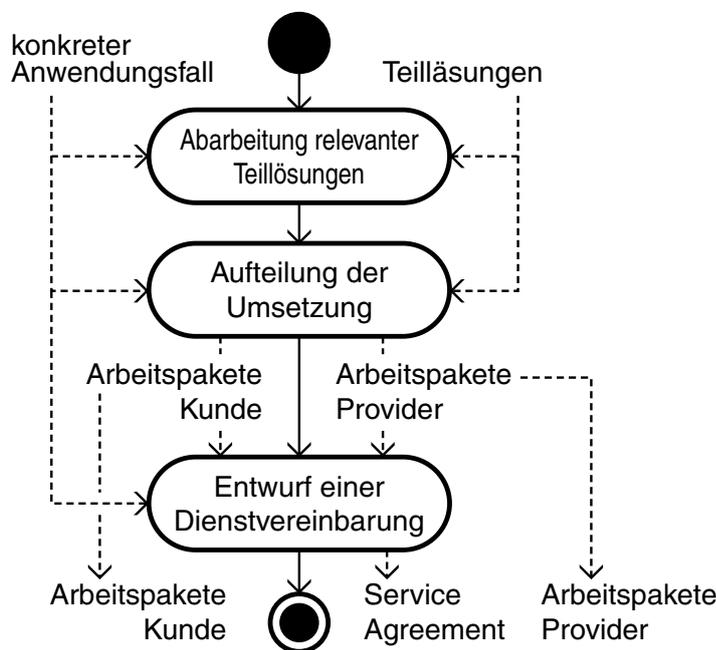


Abbildung 5.26: Aktivitäten zu Projektplanung und Dienstvereinbarung

Die einzelnen Schritte, die in der Verhandlungsphase durchzuführen sind, werden in Abbildung 5.26 gezeigt. Beschrieben werden die Schritte in den nachfolgenden Unterabschnitten.

5.5.3.1 Abarbeitung relevanter Teillösungen

Bevor weitere Teilaktivitäten erfolgen können, müssen zunächst aus der Menge der erarbeiteten Teillösungen jene herausgesucht werden, die Aktivitäten in der Verhandlungsphase erfordern.

Wurden solche Teillösungen gefunden, sind diese jetzt abzuarbeiten. Ist deren Abarbeitung abgeschlossen, dann können jene Teillösungen deren Lösungsworkflows ausschließlich Aktionen in der Verhandlungsphase benö-

Abarbeitung von Lösungen der Verhandlungsphase

tigen, aus der Lösungsmenge entfernt werden. Sie sind abgearbeitet und spätere Aktionen sind nicht notwendig.

5.5.3.2 Aufteilung der Umsetzung

Aufteilung zwischen Kunde und Provider	Zu Beginn der Verhandlungsphase steht durch die vorangegangenen Schritte fest, welche Lösungen zur Durchführung der Koppelung zu realisieren sind. Offen geblieben ist dabei zum jetzigen Zeitpunkt, welche Teillösung von wem – Kunde oder Provider – umgesetzt wird. Im Schritt der Projektplanung wird genau das zwischen Kunde und Provider ausgehandelt.
Vergleich der Hierarchiestufen	Die prinzipiellen Möglichkeiten, wie die zu erbringenden Lösungen zwischen den Seiten aufgeteilt werden kann, wird zunächst durch die Stellung der Beteiligten in der Providerhierarchie beeinflusst:
Gleiche Hierarchiestufe	<p>I) Beide Partner befinden sich auf der gleichen Hierarchiestufe</p> <p>In diesem Fall wird die Aufteilung der Arbeitspakete zwischen Kunde und Provider individuell vereinbart werden, da beide „auf gleicher Augenhöhe“ miteinander verhandeln. Nahezu beliebige Varianten sind denkbar, von „einer macht alles“ bis „jeder kümmert sich um die Teile, die von seinem Dienst verursacht wurden“.</p>
Ungleiche Hierarchiestufe	<p>II) Der Kunde befindet sich auf einer höheren Hierarchiestufe</p> <p>Liegt dieser Fall vor, dann ist der Kunde meist einer unter vielen. Er ist potentiell in der schwächeren Position; er verhandelt nicht auf „gleicher Augenhöhe“. Die Unterscheidung verschiedener Diensttypen auf Providerseite, wie sie in [SchmH 01] und [Dreo 02] gemacht wird, liefert dann weitere Hinweise:</p>
Unterscheidung von Diensttypen	<p>a) Massendienst und Standarddienst</p> <p>Bei diesem Diensttyp können durch den Kunden keine (Massendienst) oder nur wenige und fest definierte (Standarddienst) Veränderungen an dem durch den Provider angebotenen Dienst vorgenommen werden. Dementsprechend wird der Provider keinen Beitrag zur Umsetzung der Lösungen leisten und alles wird vom Kunden zu erledigen sein.</p>
Massendienst, Standarddienst	
Individualdienst	<p>b) Individualdienst</p> <p>Bei einem Individualdienst wird der Dienst durch den Provider individuell auf die Kundenbedürfnisse zugeschnitten. Die Verhandlungsposition des Kunden ist viel stärker als bei Massen- und Standarddiensten. Wie für den Fall, dass beide Partner auf gleicher Hierarchiestufe stehen, wird die Aufteilung der Umsetzung individuell verhandelt werden.</p>
Arbeitspakete	Das Ergebnis der Aufteilung der Teillösungen wird Arbeitspakete genannt. Aus den Arbeitspaketen ergibt sich zum einen, wer die von den Teillösungen

gen benötigten Lösungskomponenten betreibt, und zum anderen, wer die benötigten Lösungsworkflows ausführt und umsetzt.

5.5.3.3 Entwurf einer Dienstvereinbarung

- | | |
|---|---|
| <p>Ist die Aufteilung der Umsetzung in Arbeitspakete abgeschlossen, dann gilt es diese Aufteilung vertraglich zu fixieren. Das hierfür vorgesehene Dokument ist eine Dienstvereinbarung – das <i>service agreement</i> aus Abbildung 4.30, Seite 129.</p> | <p>Dienstvereinbarung / SLA</p> |
| <p>Für die obigen Aufteilungsvariante II.a ist an dem bestehenden <i>service agreement</i> keine Änderung für die Koppelung notwendig. Im Fall eines Standarddienstes muss der Kunde nur die für die Umsetzung seiner Lösung benötigten Optionen auswählen.</p> | <p>Standard-SLA bei Massen- / Standarddienst</p> |
| <p>In den Fällen I und II.b dagegen muss ein individuelles <i>service agreement</i> erstellt werden. H. Schmidt hat hierzu in [SchmH 01] eine Arbeit vorgelegt, die exakt diese Problematik löst.</p> | <p>Individuelles SLA</p> |
| <p>In [SchmH 01] wird ein Lösungsansatz für die Erstellung und die Darstellung von Dienstvereinbarungen vorgestellt, der Konzepte der Workflow-Modellierung auf <i>service agreements</i> anwendet. Prinzipiell können auch andere Ansätze als der von H. Schmidt für die Erstellung des <i>service agreement</i> benutzt werden, aber [SchmH 01] ist hier geradezu prädestiniert: Dadurch, dass in den Teillösungen zur Koppelung bereits Teilworkflows erarbeitet wurden und Schmidts Ansatz selbst Workflows zur Spezifikation des <i>service agreement</i> verwendet, lassen sich die erarbeiteten Teilworkflows auch direkt in der Erstellung des <i>service agreement</i> weiterverwenden. Weiterhin ergänzen sich der von Schmidt spezifizierte Ablauf zur Erstellung der Dienstvereinbarung und die in dieser Arbeit vorgestellte Methodik optimal.</p> | <p>H. Schmidts Ansatz integriert sich nahtlos</p> |

5.5.4 Implementierung der Koppelung

- | | |
|--|---|
| <p>Nachdem das <i>service agreement</i> erstellt und von beiden Seiten unterzeichnet worden ist, wird die Verhandlungsphase verlassen und die Koppelung tritt in die Phase der Bereitstellung.</p> | <p>Eintritt in Bereitstellungsphase</p> |
| <p>Jetzt werden auf Seiten von Kunde und Provider die in der Projektplanung vorgenommene Aufteilung der Teillösungen, die im <i>service agreement</i> formal fixiert wurde, umgesetzt. Parallel führen Kunde und Provider die in den Teillösungen spezifizierten Workflows aus und implementieren so die Koppelungslösung.</p> | <p>Beidseitig parallele Implementierung</p> |
| <p>Am Ende ist mit dem Abschluss der Bereitstellungsphase die Koppelung vollzogen und das gekoppelte Gesamtsystem betritt die Betriebsphase.</p> | <p>Koppelung vollzogen</p> |

5.6 ZUSAMMENFASSUNG UND ÜBERBLICK ÜBER DIE KOPPELUNGSMETHODIK

Hauptschritte eingeordnet in Lebenszyklus In den vorangegangenen Sektionen dieses Kapitels wurde eine vollständige und generische Methodik zur Koppelung von Netz-QoS-Architekturen entworfen. Abbildung 5.27 zeigt die Hauptschritte der Methodik und ihre Einordnung in den Lebenszyklus des gekoppelten Gesamtsystems:

Planungsphase Die Planungsphase beginnt damit jeweils ein Modell der beiden zu koppeln- den Netz-QoS-Dienste zu erstellen. Die einzigen Eingabeartefakte sind die zu modellierenden Dienste. Der konkrete Anwendungsfall der Koppelung spielt noch keine Rolle. Ergebnis dieses Schritts sind zwei Dienstmodelle der zu koppeln- den Dienste, die für alle restlichen Schritte der Planungspha- se als Eingabeartefakte dienen.

Im zweiten Schritt, der Analyse von Grundeigenschaften, wird erstmals der konkrete Anwendungsfall betrachtet. Ein wichtiges Ergebnis ist u.a. die Einteilung in einen der beiden Grundanwendungsfälle Transitiv-Koppelung oder Tunnel-Koppelung. Die gefundenen Grundeigenschaften sind Eingabe- artefakte aller nachfolgenden Schritte der Planungsphase.

Der folgende Schritt führt erstmals einen direkten Vergleich der Dienstmo- delle durch, nämlich eine Gegenüberstellung der statischen Modellteile der Nutzfunktionalität. Das Ergebnis besteht aus einer Menge von Teillösungen für die Koppelung.

Die dynamischen Aspekte der Nutzfunktionalität werden im nächsten Schritt verglichen. Die im vorangegangenen Schritt gefundene Menge der Teillösun- gen dient zum einen als Eingabeartefakte, zum anderen wird sie im Laufe des Teilprozesses um weitere Teillösungen erweitert.

Der vorletzte Schritt der Planungsphase bearbeitet dann die verbliebenen Teile aus den Dienstmodellen. Wiederum werden dabei weitere Teillösungen gefunden.

Die Planungsphase wird abgeschlossen mit der Entscheidung, ob eine Kop- pelung zwischen den Diensten realisiert werden soll. Die Basis für diese Entscheidung ist durch die Menge der Teillösungen gegeben. Anhand derer lässt sich der Aufwand der Koppelung und die Eigenschaften des Gesamtsy- stems für den konkreten Anwendungsfall ablesen.

Verhandlungsphase Wenn die Koppelung durchgeführt werden soll, dann beginnt nun die Ver- handlungsphase. Darin werden zuerst die gefundenen Teillösungen zwi- schen Kunde und Provider in Arbeitspakete aufgeteilt. Diese Aufteilung wird formal dann zwischen den beiden Parteien in einer Dienstvereinbarung fest- gehalten. Die Eingabeartefakte für beide Vorgänge sind der konkrete An- wendungsfall und die Menge der Teillösungen. Die Dienstmodelle selbst werden nicht mehr zwingend benötigt, wenngleich sie Eingang in die Dienst- vereinbarung finden können.

5.6 ZUSAMMENFASSUNG UND ÜBERBLICK ÜBER DIE KOPPELUNGSMETHODIK

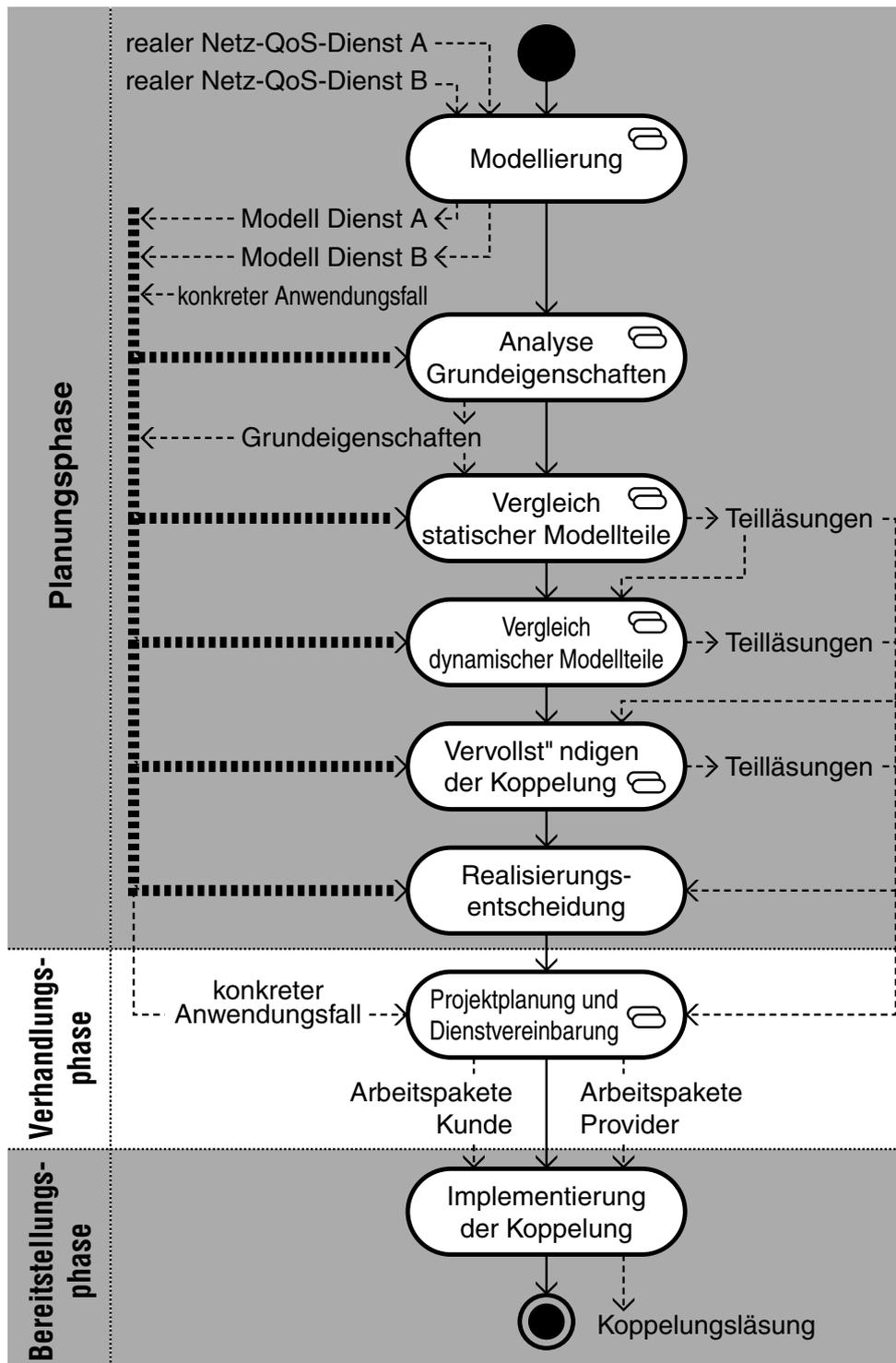


Abbildung 5.27: Überblick über die gesamte Methodik

Mit der Unterzeichnung der Dienstvereinbarung ist die Verhandlungsphase abgeschlossen und es beginnt die Bereitstellungsphase. Darin implementieren Kunde und Provider die ihnen zugewiesenen Arbeitspakete. Ist die Implementierung abgeschlossen, dann ist Koppelungslösung realisiert und das Gesamtsystem kann in die Betriebsphase übergehen.

5.6.1 Abgleich mit den Anforderungen

Mit der jetzt vollständig entwickelten Lösung der Koppelungsproblematik gilt es diese mit den Anforderungen aus Sektion 2.4 abzugleichen.

Allgemeine Anforderungen Der Vergleich der vollständigen Lösung mit den allgemeine Anforderungen aus Abschnitt 2.4.1 ergibt:

- Anforderung: Nachhaltige und allgemeine Anwendbarkeit

Die Methodik richtet sich vollständig nach der Modellierung aus Kapitel 4. Deren nachhaltige und allgemeine Anwendbarkeit wurde bereits in Abschnitt 4.5.1 gezeigt. Dies lässt sich direkt auch auf die Methodik übertragen, weil bei deren Entwicklung wiederum keinerlei Einschränkungen gemacht werden, die über die Dienstorientierung hinausgehen. Die Anforderung ist vom Lösungsansatz dieser Arbeit voll erfüllt.

- Anforderung: Produktorientierung

Die Produktorientierung wurde bereits durch die Modellierung umgesetzt (Abschn. 4.5.1). Die Anforderung ist vom Lösungsansatz dieser Arbeit voll erfüllt.

- Anforderung: szenariospezifische Ergebnisorientierung

Der szenariospezifischen Ergebnisorientierung wird in der Methodik durchgehend Rechnung getragen. Zu den einzelnen Teilproblemen werden (soweit möglich) mehrere Lösungsmöglichkeiten angeboten. Die konkrete Auswahl der Lösungsmöglichkeiten muss dann am konkreten Anwendungsfall erfolgen. Dadurch kann die Lösung nach dem gewünschten Ergebnis gestaltet werden. Die Anforderung ist vom Lösungsansatz dieser Arbeit voll erfüllt.

- Anforderung: Realisierbarkeit

Die Realisierbarkeit des Lösungsansatzes zeigt sich in den einzelnen Teilentscheidungen. Dort werden realisierbare Einzelkomponenten mit Angabe der notwendigen Aktivitäten zu deren Realisierung genannt. Die Gesamtlösung setzt sich ausschließlich aus den realisierbaren Teillösungen zusammen. Damit ist die Gesamtlösung selbst auch realisierbar. Die Anforderung ist vom Lösungsansatz dieser Arbeit voll erfüllt.

Anforderungen an Methodik In Abschnitt 2.4.4 wurden noch Anforderungen gestellt, die speziell von der Methodik zu erfüllen sind:

- Anforderung: Integrationsmöglichkeit für Teil- und Individuallösungen

In der Methodik selbst wurden zu einzelnen Teilproblemen bereits teilweise mehrere Lösungsmöglichkeiten angegeben. Diese Lösungslisten können auch nachträglich noch erweitert werden und so können andere Teil- und Individuallösungen in die Methodik aufgenommen werden. Durch das erweiterbare Namensschema (Abschn. 5.2.2) können existierende Teillösungen auch weiter diversifiziert werden. Durch die

5.6 ZUSAMMENFASSUNG UND ÜBERBLICK ÜBER DIE KOPPELUNGSMETHODIK

iterativen Vergleichsprozesse sind dabei aber häufig keine Änderungen an den Teilprozessen der Methodik notwendig. Die Anforderung ist durch die Methodik voll erfüllt.

■ Anforderung: Gesamter Lebenszyklus

Der gesamte Lebenszyklus ist sogar auf zweierlei Art fester Bestandteil der Arbeit. Zum einen behandeln Methodik (und auch Modellierung) den gesamten Lebenszyklus. Zum anderen erstreckt sich die Methodik selbst über alle für sie relevanten Phasen (Abb. 5.27). Die Anforderung ist durch die Methodik voll erfüllt.

■ Anforderung: Spezifikation von Komponenten und Workflows

Durch die in Abschnitt 5.2.2 eingeführte Notationsstruktur wurde die Spezifikation von sowohl Komponenten als auch Workflows schon vor der eigentlichen Methodik etabliert. Im Laufe der Methodik wurden dann Komponenten und Workflows (soweit beides sinnvoll möglich ist) für die einzelnen Teillösungen spezifiziert. Die Anforderung ist durch die Methodik voll erfüllt.

Tabelle 5.16 fasst den eben durchgeführten Abgleich mit den Anforderungen nochmals zusammen. Darin aufgenommen sind auch die Ergebnisse aus Abschnitt 4.5.1. Dort wurden bereits die Anforderungen, die nur an die Modellierung gestellt wurden, abgeglichen.

Wie aus Tabelle 5.16 zu ersehen ist, erfüllt die Arbeit die an sie gestellten Anforderungen. Ganz zu Beginn dieser Arbeit wurden auch einige Teilfragestellungen aufgeworfen, die jetzt größtenteils beantwortet werden können: Teilfragestellungen

- Die einzelnen angegebenen Teillösungen geben jeweils explizit Antwort darauf, ob und wie sie die Gesamtfunktionalität des gekoppelten Systems beeinflussen.
- Die Frage nach der Rolle des Dienst-Lebenszyklus wird durch dessen immanente Verwendung in Modell und Methodik beantwortet.
- Die Frage nach Teilvorgängen und architekturellen Komponenten, die für eine Koppelung notwendig sind, wird detailliert in den einzelnen Lösungsvarianten der Teilproblematik beantwortet.
- Der Einfluss des Einsatzzwecks ist ebenfalls im Laufe der Methodik an den einzelnen Teilproblemen aufgezeigt worden.
- Der formale Aufbau und die Teilprozesse der Methodik sind so gestaltet, dass mit mehreren Lösungsmöglichkeiten umgegangen werden kann und wird.

Einzig die Teilfragestellungen nach der Kategorisierung von Netz-QoS-Architekturen sowie der Nutzung und Einordnung bisheriger Arbeiten zur Koppelungsproblematik sind bisher noch nicht vollständig beantwortet worden. Diesen Fragen wird sich u.a. das nächste Kapitel widmen. Dort werden Anwendungsbereich der Modellierung und der Methodik aufgezeigt.

Allgemeine Anforderungen	
Anforderung	Bewertung
Nachhaltig und allgemein anwendbar	✓
Produktorientierung	✓
szenariospezifische Ergebnisorientierung	✓
Realisierbarkeit	✓

Anforderungen an Modellierung	
Anforderung	Bewertung
Normalisierung	✓
Dienstorientierung	✓
Trennung von Dienst und Realisierung	✓
Management integraler Dienstbestandteil	✓
Abdeckung des gesamten Lebenszyklus	✓

Anforderungen an Methodik	
Anforderung	Bewertung
Integrationsmöglichkeit für Teil- und Individuallösungen	✓
Gesamter Lebenszyklus	✓
Spezifikation von Komponenten und Workflows	✓

✓: erfüllt

Tabelle 5.16: Erfüllung der Anforderungen durch diese Arbeit

KAPITEL 6

ANWENDUNGSBEREICHE VON MODELLIERUNG UND METHODIK

Nachdem in den vorangegangenen Kapiteln sowohl die Modellierung als auch die Methodik zur Koppelung von Netz-QoS-Diensten entwickelt worden sind, beschäftigt sich dieses Kapitel mit Aspekten der Anwendungen beider Teile. Anwendungsaspekte

Die Erfüllung der gestellten Anforderungen wurde bereits am Ende des vorangegangenen Kapitels gezeigt. Die explizite Beantwortung einiger der in der Einleitung aufgeworfenen Teilfragestellungen ist jedoch bis jetzt offen geblieben. Diese werden im Laufe des Kapitels aufgegriffen und beantwortet. Damit wird die letzte verbliebene größere Lücke in der Arbeit geschlossen. Verbliebene
Teilfragestellungen
werden beantwortet

Im Laufe des Kapitels wird deutlich, dass nicht nur die Methodik insgesamt bedeutsam ist, sondern dass auch die Modellierung alleine ihre Anwendung finden kann.

6.1 ERGEBNISABSCHÄTZUNG EINER KOPPELUNG

Schnelle Abschätzung der Gesamtfunktionalität	In der Einführung zu dieser Arbeit wurde die Teilfragestellung nach der Gesamtfunktionalität eines gekoppelten Gesamtsystems gestellt. Diese Frage lässt sich natürlich bereits detailliert beantworten, indem man die gesamte Methodik vollständig anwendet. In einigen Fällen ist jedoch eine initiale Abschätzung der Gesamtfunktionalität von Interesse. Diese muss sich möglichst einfach und vor allem schnell erstellen lassen.
Voraussetzung: Modelle der Dienste Näherung durch Architekturmodelle	Für eine einfache Abschätzung lassen sich Teile der Methodik verwenden. Die Ausgangsvoraussetzung ist, dass die abzuschätzenden Dienste nach Sektion 5.1 bzw. Abbildung 5.2, Seite 137 modelliert sind. Liegen nur Modellierung der den Diensten zugrunde liegenden Netz-QoS-Architekturen vor, dann können diese als Näherung benutzt werden. Die Abschätzung wird dadurch natürlich unzuverlässiger.
Schritt: Grundeigenschaften des Anwendungsfalls	Für den ersten Schritt der Abschätzung müssen die Grundeigenschaften des Anwendungsfalls geklärt werden. Das geschieht identisch zur vollständigen Ausführung der Methodik nach Abschnitt 5.2.1 und Abbildung 5.6, Seite 143.
Weitere Schritte angelehnt an Methodik	Auch die weiteren Schritte, nämlich jene zur Koppelung der statischen Modellteile (Abb. 5.9, S. 153), werden zur Abschätzung im Prinzip wie in Sektion 5.3 dargelegt, ausgeführt. Zur Beschränkung des Aufwands in der schnellen Abschätzung werden allerdings einige Näherungsmuster angewendet:
Kein vollständiger wechselseitiger Vergleich	<ul style="list-style-type: none"> ■ Alle Iterationen der Methodik, d.h. jene Teilprozesse die dem Muster aus Unterabschnitt 5.2.3.1 bzw. Abbildung 5.7, Seite 147 gehorchen, sollten nicht vollständig durchlaufen werden. Für einen beschleunigten Durchlauf sollten nur „dem Namen nach“ ähnliche bzw. vergleichbare Klassen gegenübergestellt werden.
Abkürzung Semantikvergleich	<ul style="list-style-type: none"> ■ Zur Gegenüberstellung der Klasse <i>Semantik</i> von <i>QoS-Parametern</i> kann ebenfalls der vergrößerte Vergleich nach „ähnlichen Namen“ verwendet werden um die Abschätzung zu beschleunigen. Beispielsweise beim Vergleich zwischen ATM und DiffServ AF PHB würde nur die «QoS-Parameter»-Paarung <i>CLR</i> (von ATM) und <i>Loss</i> (von DiffServ AF PHB) abgeglichen werden.
Abkürzung Fallanalysen	<ul style="list-style-type: none"> ■ Die größte Ersparnis in der Ergebnisabschätzung kommt durch die Abkürzung der Fallanalysen zustande. Für jede Fallanalyse wird in Sektion 5.3 eine Tabelle angegeben. Für jeden Fall ist darin in der Spalte „potenziell vollständig“ angegeben, ob es für diesen Fall eine im Sinne des Gesamtsystems vollständige Lösung geben kann. Das Symbol „✓“ zeigt an, dass es eine Lösungsmöglichkeit geben kann, bei der das Gesamtsystem die volle Funktionalität erhält. Ein „✗“ dagegen zeigt an, dass das Gesamtsystem in jedem Fall eingeschränkt sein wird.

6.1 ERGEBNISABSCHÄTZUNG EINER KOPPELUNG

Anstatt eine Lösung für den Fall auszuwählen, wird in der Ergebnisabschätzung jetzt nur der Eintrag in der Spalte „potentiell vollständig“ aufgesammelt.

Anstatt eine Menge von Teillösungen zu ermitteln, werden durch die Methodik nur eine Menge von Kurzbewertungen der Form „✓“ oder „X“ erstellt. Werden in den einzelnen Falltabellen nur mit „✓“ gekennzeichnete Fälle gefunden, dann besteht eine Chance, dass das Gesamtsystem die gewünschten Eigenschaften erhalten kann. Findet sich aber nur ein Fall, der mit „X“ gekennzeichnet ist, dann wird das Gesamtsystem in jedem Fall nicht den vollen gewünschten Umfang erhalten. Ergebnisinterpretation

Für die Ergebnisabschätzung endet die Methodik mit dem Vergleich der statischen Modellteile. Auf die Schritte „Vergleich dynamischer Modellteile“ und „Abschluss der Koppelung“ aus Abbildung 5.1, Seite 136 wird für die Abschätzung verzichtet. Für eine Abschätzung ist dies sinnvoll, da zumindest ein negativ-definites Ergebnis geliefert wird. Für ein positiv-definites Ergebnis muss ohnehin die gesamte Methodik vollständig abgearbeitet werden. Abbruch nach Vergleich statischer Teile
Negativ-definites Ergebnis

Wie bereits eingangs der Sektion erwähnt handelt es sich bei dem Ergebnis nur um eine grobe Abschätzung. Selbst wenn das Fazit positiv ausfällt, bedeutet das nicht, dass in Details keine weiteren Einschränkungen auftreten können, z.B. bei den exakten Wertebereichen. Auch für die Abschätzung des Aufwands zur tatsächlichen Implementierung der Koppelung muss die Methodik detaillierter angewendet werden, weil die ausgewählten Teillösungen den Aufwand maßgeblich beeinflussen. Keine Details
Keine Aufwandsabschätzung

6.2 TAXONOMIEN FÜR ARCHITEKTUREN, DIENSTE UND KOPPELUNGEN

Durch die Modellierung von Netz-QoS-Architekturen und -Diensten in Kapitel 5 wurden nicht nur eine vereinheitlichende Sicht auf Netz-QoS-Architekturen bzw. -Dienste gewonnen, die die wichtigste Ausgangsbasis für die generische Methodik darstellt. Die Modellierung alleine kann auch als Mittel zur Einordnung von Netz-QoS-Architekturen und -Diensten dienen.

6.2.1 Modellierung als Taxonomie

Taxonomie für
Architekturen und
Dienste

Um verschiedene Netz-QoS-Architekturen bzw. -Dienste zueinander in Beziehung zu stellen, liefert die Modellierung durch ihre Dienstorientierung eine Antwort aus Anwendersicht. Mit der Modellierung wurde, quasi „nebenbei“, eine **Taxonomie** für Netz-QoS-Architekturen und Netz-QoS-Dienste geschaffen. Der Begriff der Klassifikation wird an dieser Stelle absichtlich vermieden, denn häufig wird damit ein streng hierarchisch aufgebautes Gruppierungssystem impliziert. Die durch die Modellierung dargestellten Eigenschaften verhalten sich jedoch nur in Teilen hierarchisch zueinander. An den Stellen, wo sich hierarchische Beziehungen finden, wurde das im (Meta-) Modell bereits dargestellt, z.B. bei der Klasse *Garantiety* (Abb. 4.12, S. 92).

Überprüfung auf
Eindeutigkeit und
Vollständigkeit

Durch die Taxonomie werden Netz-QoS-Modelle bzw. -Dienste untereinander aus Anwendersicht vergleichbar: eine Eigenschaft, die die Basis für die generische Methodik bildet. Aber die Modellierung alleine kann auch genutzt werden, um eine Spezifikation auf Eindeutigkeit bzw. Vollständigkeit zu überprüfen. Können nämlich Teile des (Meta-) Modells für eine konkrete Architektur oder einen konkreten Dienst nicht instanziiert werden, dann ist das ein Zeichen dafür, dass die Spezifikation nicht vollständig ist. Auch Mehrdeutigkeiten werden durch die Modellierung aufgedeckt.

Im Modell von DiffServ Assured Forwarding PHB kann man unmittelbar erkennen, dass die «QoS-Kategorien» *AF_1* bis *AF_4* hinsichtlich ihrer QoS-Eigenschaften nicht voneinander unterschieden werden können. Abbildung 6.1 zeigt einen Ausriss von Abbildung 4.19, Seite 105. Darauf erkennt man, dass den «QoS-Kategorien» *AF_1* bis *AF_4* keinerlei spezifische «QoS-Parameter» zugeordnet sind, die sie unterscheidbar machen würden. Vom Standard selbst wird nur der gemeinsame «Korrektheitsparameter» *Loss* spezifiziert.

Definitionskorridore
zeigen Freiheiten

Am Vorhandensein von Definitionskorridoren (Unterabschn. 4.3.4.2), die über die Standardisierungsphase hinausreichen, kann für jeden Teil der Modellierung individuell abgelesen werden, inwieweit der Standard Freiheiten bzw. Lücken lässt.

6.2 TAXONOMIEN FÜR ARCHITEKTUREN, DIENSTE UND KOPPELUNGEN

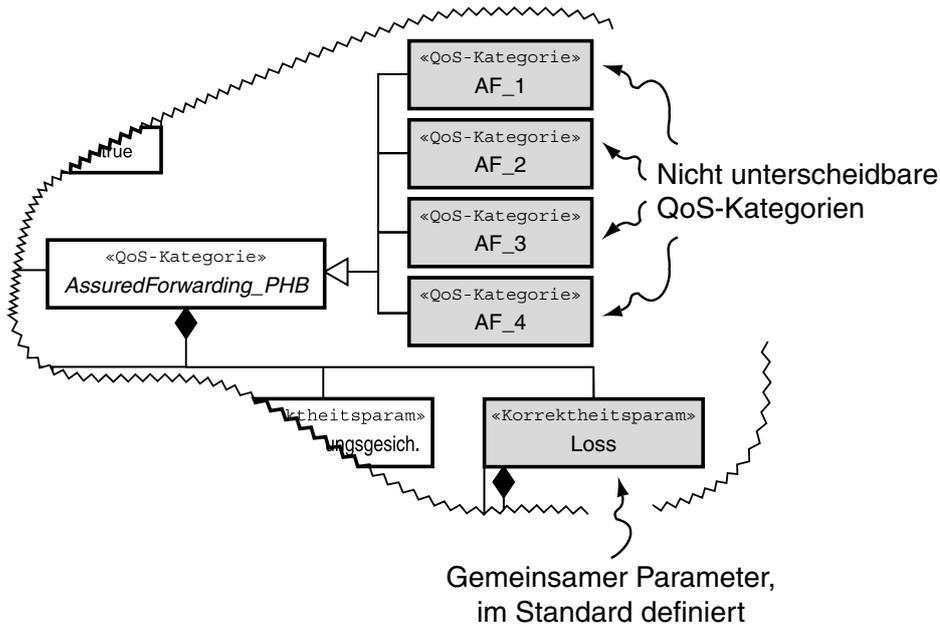


Abbildung 6.1: Ausriss aus Abbildung 4.19

Mit den Modellen wird auch noch ein weiterer Effekt erzielt, der bereits bei der Erstellung des MNM-Dienstmodells eine wichtige Motivation war: Eine einheitliche und gemeinsame **Terminologie**. Durch die Bezeichnungen der Klassen im (Meta-) Modell werden die einzelnen Teile von Architekturen und Diensten mit einheitlichen Begriffen versehen. Einheitliche Terminologie

In DiffServ werden *QoS-Kategorien* als „*QoS classes*“ bezeichnet. ATM dagegen benennt die *QoS-Kategorien* als „*QoS categories*“. Die Modellierung stellt die einheitliche Bezeichnung *QoS-Kategorien* her.

Mit der einheitlichen Terminologie und der Taxonomie, die durch die Modellierungen realisiert sind, ist auch die Teilfragestellung nach der Kategorisierung von Netz-QoS-Architekturen beantwortet. Frage der Kategorisierung beantwortet

6.2.2 Methodik als Taxonomie

Auch die Methodik kann als Taxonomie angesehen werden und zwar für Koppelungsfälle und deren Lösungen. Im Laufe der Methodik werden für einen Koppelungsfall bestimmte Teilfälle identifiziert. Jeder dieser Teilfälle ist Dank des verwendeten Namensschemas (Abschn. 5.2.2) mit einem eindeutigen Bezeichner versehen. Sammelt man während der Abarbeitung der Methodik die Menge aller Teilfallbezeichner, dann erhält man automatisch eine kompakte und standardisierte Beschreibung des Koppelungsfalls. Diese Beschreibung des Koppelungsfalls ist sogar unabhängig von konkreten Netz-QoS-Architekturen und -Diensten. Damit ist eine allgemeine Taxonomie für Koppelungsfälle gegeben. Taxonomie für Koppelungsfälle

KAPITEL 6: ANWENDUNGSBEREICHE VON MODELLIERUNG UND METHODIK

- Strukturvergleich
von
Koppelungsfällen
- Mit der Menge an Teilfallbezeichner lassen sich Koppelungsfälle bezüglich ihrer strukturellen Eigenschaften auch untereinander schnell vergleichen. Strukturell gleiche Fälle haben unabhängig von den konkreten Diensten die gleichen Teilfallbezeichner. Je unterschiedlicher die Mengen der Teilfallbezeichner sind, desto stärker unterscheiden sich zwei Koppelungsfälle.
- Taxonomie für Koppelungslösungen
- Analog zu den Koppelungsfällen bilden die Mengen von Lösungsbezeichnern eine Taxonomie für Koppelungslösungen. Ebenso wie beim Vergleich von Koppelungsfällen, können unterschiedliche Koppelungslösungen über die Mengen der Lösungsbezeichner verglichen werden.

6.3 ERSCHLISSUNG BESTEHENDER ARBEITEN

Die Frage nach der Integration bestehender Arbeiten im Koppelungsumfeld wurde bis jetzt noch nicht beantwortet. Das geschieht in dieser Sektion.

Integration bestehender Arbeiten

Die Voraussetzung zur Erschließung bestehender Arbeiten wurde bereits vor der Schilderung der eigentlichen Methodik gelegt. In Abschnitt 5.2.2 wurde eine einheitliche Notationsstruktur festgelegt und zur (Quer-) Referenzierung ein Namensschema eingeführt.

Voraussetzungen geschaffen

Das Namensschema ist in zwei Richtungen erweiterbar:

■ Horizontale Erweiterung

Es können neue Teilfälle eingeführt werden. Sowohl auf Ebene der Teilbereichsbezeichner (2. Teil im Namensschema) als auch auf Ebene der Teilfälle (3. Teil im Namensschema) können Erweiterungen hinzugefügt werden. Dabei ist nur darauf zu achten, dass auf der jeweiligen Stelle der neu eingeführte Bezeichner noch nicht existiert.

Horizontale Erweiterung

■ Hierarchische Verfeinerung

Sowohl bei Fallbezeichnern als auch bei Lösungsbezeichnern können neue Hierarchiestufen durch Anhängen im Namensschema erzeugt werden. Damit können bereits in der Methodik existierende Fälle und Lösungen weiter verfeinert werden.

Hierarchische Verfeinerung

Durch die Notationsstruktur wird auch die „Aufschreibung“ für Ergänzungen und Erweiterungen bereits vorgegeben.

Um eine Lösung aus einer bestehenden Arbeit in die Methodik integrieren zu können verbleibt der Schritt der Einordnung. Anhand des (Meta-) Modells muss die Stelle in der Methodik gefunden werden, an der die in der bestehenden Arbeit entwickelte Teillösung einzusortieren ist. Hierzu muss die meist konkrete Lösung relativ zum Modell abstrahiert werden. Ist dies geschehen, kann die Arbeit entweder als horizontale Erweiterung oder als hierarchische Verfeinerung in die Methodik integriert werden.

Einordnung in Methodik

Abstraktion

Der folgenden Abschnitt führt das geschilderte Verfahren an einem konkreten Beispiel vor.

Beispiel zur Integration einer bestehenden Arbeit

Die Arbeit [KZS oo] (siehe auch Abschn. 3.2.2) beschäftigt sich mit der Koppelung von IntServ nach ATM. Das dort betrachtete Szenario untersucht die Koppelung zweier Domänen ohne Einbeziehung dritter Domänen. Daher ist, in der Terminologie dieser Arbeit, der Grundanwendungsfall eine **Transitiv-Koppelung**.

RSVP → ATM In der Arbeit werden drei unterschiedliche Varianten zum Verbindungsaufbau im Übergang RSVP → ATM vorgestellt, die in Abbildung 6.2 dargestellt sind. RSVP ist in IntServ ein Protokoll zur Vereinbarung der *QoS-Kategorien*. Wie in Abschnitt 4.3.3 gezeigt, handelt es sich bei IntServ und ATM um Architekturen, bei denen für jeden Flow eine eigene Instanz für *QoS-Kategorien* erzeugt werden. In der Terminologie dieser Arbeit beschäftigt sich [KZS oo] also mit einer Transitiv-Koppelung mit **Mehrfachinstanz-QoS-Kategorie** → **Mehrfachinstanz-QoS-Kategorie**. Nach Tabelle 5.15 ist dies Fall KM-Qkl-Tr₄.

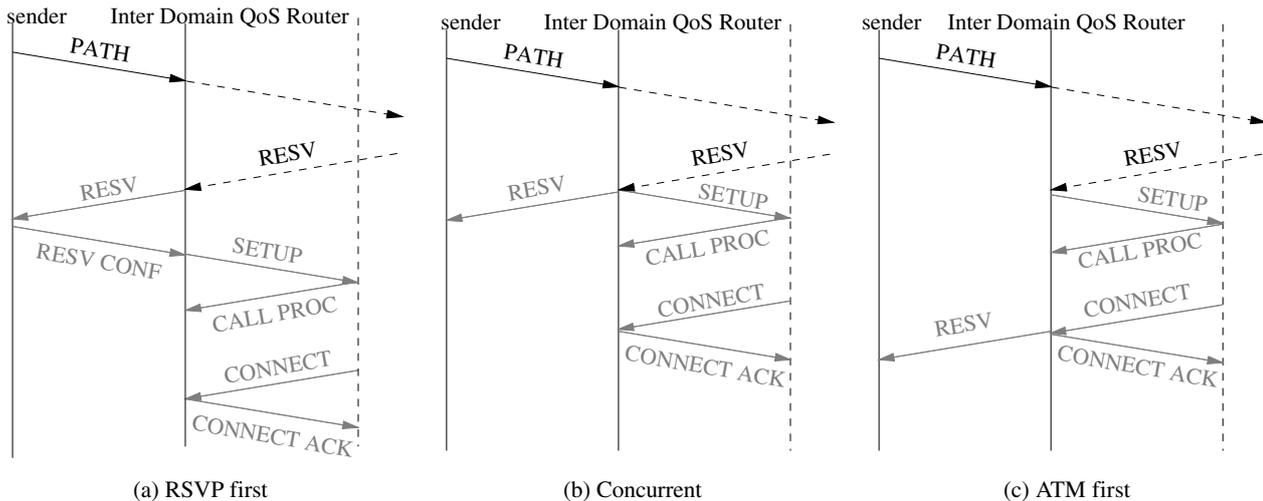


Abbildung 6.2: Varianten zum RSVP → ATM Verbindungsaufbau aus [KZS oo]

Spezialfälle zur QoS Vereinbarung Abstrahiert man die drei Fälle aus Abbildung 6.2, dann entspricht die RSVP-Seite (links) dem Quelldienst, die ATM-Seite (rechts) dem Zieldienst. Alle drei Teilabbildungen beschreiben unterschiedliche Varianten für den dynamischen Verbindungsaufbau. Sie sind demnach Verfeinerungen der Teillösung L1-(KM-Qkl-Tr₄), der **dynamischen Instanziierung**.

Aus den Analysen ergibt sich die Integration in die Methodik der drei Teilfälle aus Abbildung 6.2 als Verfeinerung der Lösung L1-(KM-Qkl-Tr₄):

- Fall KM-Qkl-Tr₄ ♦ **KM-Qkl-Tr₄**: Transitiv-Koppelung mit Mehrfachinstanz-QoS-Kategorie → Mehrfachinstanz-QoS-Kategorie.
- Lösung L1-1-(KM-Qkl-Tr₄) ★ **L1-1-(KM-Qkl-Tr₄)**: Dynamische Instanziierung, Quelldienst zuerst.
- ▶ **Analoge Lösung:** L1-(KM-Vo-Tr₄).
 - ▶ **Zusätzliche Aktivitäten:** Instanziierung erfolgt im Quelldienst zuerst, nach [KZS oo, Abb. 6.2(a)].
- Lösung L1-2-(KM-Qkl-Tr₄) ★ **L1-2-(KM-Qkl-Tr₄)**: Dynamische Instanziierung, Quell- und Zieldienst gleichzeitig.
- ▶ **Analoge Lösung:** L1-(KM-Vo-Tr₄).
 - ▶ **Zusätzliche Aktivitäten:** Instanziierung erfolgt in Quell- und Zieldienst gleichzeitig, nach [KZS oo, Abb. 6.2(b)].

6.3 ERSCHLIESSUNG BESTEHENDER ARBEITEN

- ★ **L1-3-(KM-Qkl-Tr4)**: Dynamische Instanziierung, Zieldienst zuerst. Lösung L1-3-(KM-Qkl-Tr4)
- ▶ **Analoge Lösung**: L1-(KM-Vo-Tr4).
 - ▶ **Zusätzliche Aktivitäten**: Instanziierung erfolgt im Zieldienst zuerst, nach [KZS oo, Abb. 6.2(c)].

Damit ist auch allgemein und an einem Beispiel gezeigt, wie das Wissen aus bereits existierenden Arbeiten mit und für die Methodik dieser Arbeit erschlossen werden kann.

6.4 GESCHÄFTLICHE ANWENDUNGSFÄLLE

Zur weiteren Illustration der praktischen Relevanz dieser Arbeit werden in dieser Sektion einige geschäftliche Anwendungsfälle darauf untersucht, wie sich Modellierung und Methodik einsetzen lassen. Die betrachteten Anwendungsfälle sind jene aus [GHHK 02], die schon im Rahmen des MNM-Dienstmodells untersucht worden sind.

6.4.1 Anwendungsfall Ausschreibung

Ein Unternehmen möchte sein eigenes QoS-fähiges Netz in einem Netzverbund erreichbar machen. Hierzu soll das günstigste Angebot zwischen verschiedenen Anbietern ermittelt werden. Das probate Verfahren hierfür ist eine (öffentliche) Ausschreibung. Dabei werden potentielle Anbieter zur Abgabe eines Angebotes aufgefordert.

Zur Erstellung der Ausschreibung leistet die Modellierung des QoS-fähigen Netzes für alle Beteiligten einen wertvollen Beitrag:

- | | |
|-------------------------|---|
| Vollständigkeit | ■ Durch die Modellierung wird sichergestellt, dass alle für die Erstellung eines Angebotes relevanten Angaben den Teilnehmern bekannt gemacht werden. Verzögerungen im Angebotsprozess und spätere Streitfälle können dadurch minimiert werden. |
| Verbergen von Details | ■ Das Netz des ausschreibenden Unternehmens wird durch die Modellierung in einer Sicht dargestellt, die Interna der Implementierung des Netzes des Unternehmens verbirgt. Für das Unternehmen bedeutet das einen Sicherheitsaspekt. |
| Zielgerichtete Anfrage | ■ Die Teilnehmer an der Ausschreibung erhalten durch die Modellierung nur die für sie relevanten Angaben. Sie werden nicht mit unnötigen Details überfrachtet. Dadurch können Angebote schneller und zielgerichteter verfasst werden. |
| Gemeinsames Verständnis | ■ Durch die Modellierung wird die Ausschreibung in einer für alle Teilnehmer nachvollziehbaren und eindeutigen Terminologie verfasst. Missverständnisse durch unterschiedliche Begriffsbildungen werden ausgeschlossen. |

Zur Erstellung der Modellierung kann das ausschreibende Unternehmen auf den ersten Teil der Methodik zurückgreifen. Sektion 5.1 gibt die notwendigen Teilaktivitäten an.

6.4.2 Anwendungsfall Angebotserstellung

Soll ein Provider ein Angebot zur Koppelung seines QoS-fähigen Netzes mit dem eines Kunden erstellen, dann können Modellierung und Methodik auch hier Verwendung finden:

6.4 GESCHÄFTLICHE ANWENDUNGSFÄLLE

- Wurde seitens des Anfragers die Modellierung dieser Arbeit noch nicht verwendet, dann kann der Provider das Angebot intern zuerst im als Instanz des (Meta-) Modells darstellen. Er realisiert dadurch die bereits im vorangegangenen Abschnitt erläuterten Vorteile der Vollständigkeit, der Fokussierung auf relevante Angaben und einer gemeinsamen Terminologie. Anfragenormierung
- Um festzustellen, ob im Sinne der Gesamtfunktionalität der Koppelung überhaupt ein sinnvolles Angebot möglich ist, kann der Provider die schnelle Ergebnisabschätzung aus Sektion 6.1 verwenden. Schnelle Vorüberprüfung
- Zur Ermittlung der Details eines Angebots kann der Provider dann die Koppelungsmethodik vollständig bis zum Schritt „Realisierungsentscheidung“ aus Abbildung 5.27, Seite 205 abarbeiten. Angebotsdetails
- Neben (technischen) Details für das Angebot, ermittelt der Provider dabei auch seinen internen Aufwand. Darauf basierend kann er seine Angebotskalkulation durchführen. Kalkulation
- Können die angefragten Leistungen im Rahmen einer Koppelung nicht unmittelbar umgesetzt werden, gibt die Methodik explizite Hinweise, welche Teile der Provider entweder intern modifizieren muss oder welche Änderungen der Kunde an seinen Wünschen akzeptieren müsste. Änderungen
- Werden Provider und Kunde sich prinzipiell handelseinig, dann können zur Umsetzung der Koppelung die restlichen Schritte aus Abbildung 5.27, Seite 205 ab „Realisierungsentscheidung“ ausgeführt werden. Umsetzungsplanung

Die vorgestellten Punkte unterstützen einen Provider vor allem in Fällen, in denen ein Individualdienst (Unterabschn. 5.5.3.2) für den Kunden erstellt werden soll.

6.4.3 Anwendungsfall Angebotsvergleich

Gilt es im Rahmen eines Koppelungsvorhabens aus einer Menge von Angeboten eines auszuwählen, dann können auch hier Modellierung und Methodik nützlich sein.

Fast alle Punkte, die im Bereich der Angebotserstellung unterstützt werden (siehe vorangegangenen Abschnitt) spielen auch beim Angebotsvergleich eine Rolle:

- Alle Angebote können durch die Modellierung und durch die im Rahmen der Methodik entwickelten Teillösungen normiert dargestellt werden. Dadurch werden die einzelnen Angebote miteinander vergleichbar. Vergleichbarkeit durch Normierung
- Die schnelle Ergebnisabschätzung aus Sektion 6.1 kann auch beim Angebotsvergleich angewendet werden. Sie wird für eine schnelle Vorsortierung der Angebote benutzt. Vorsortierung

KAPITEL 6: ANWENDUNGSBEREICHE VON MODELLIERUNG UND METHODIK

- Koppelungsdetails ■ Die für eine Koppelung notwendigen Arbeit für ein bestimmtes Angebot können im Detail mit der Koppelungsmethodik ermittelt werden.
- Kalkulation ■ Ebenso wie bei der Angebotserstellung können auf Kundenseite die internen Kosten einer Koppelung ermittelt werden, weil durch die Methodik die notwendigen Arbeitsschritte und Komponenten ermittelt werden können.
- Umsetzungsplanung ■ Wird ein bestimmtes Angebot ausgewählt, dann können Kunde und Provider die Umsetzung der Koppelung nach Abbildung 5.27, Seite 205 ab dem Schritt „Realisierungsentscheidung“ ausführen.
- Gezielte Neuanfragen ■ Konnte kein zufriedenstellendes Angebot ermittelt werden, dann liefert die Methodik explizite Hinweise, welche Eigenschaften von anderen Anbietern zu erfüllen sind bzw. welche Teile neue Angebote umfassen müssen.

6.5 PROBLEMFALL ZENTRALE AUSTAUSCHPUNKTE

In der Providerhierarchie, wie sie in Abschnitt 2.1.3 erläutert wurde, gibt es einen neuralgischen Punkt. In der hierarchischen Topologie werden fast alle Koppelungen bilateral ausgeführt, mit einer Ausnahme.

An den zentralen Austauschpunkten (z.B. [StarTap-HP, DE-CIX-HP]) treffen unter Umständen viele verschiedene Netz-QoS-Dienste aufeinander. Dies stellt deshalb ein Problem dar, da für einen möglichst verlustfreien Übergang zwischen allen Netzen, der Dienst eines Austauschpunktes alle Fähigkeiten aller in sich vereinen muss. Mit einer wachsenden Vielfalt der angeschlossenen Dienste bedeutet das, dass der Austauschpunkt-Dienst gegen einen Dienst konvergiert, der auf einer Umbrella-Architektur U aus Unterabschnitt 2.4.2.2 aufbaut. In Unterabschnitt 2.4.2.2 wurde aber auch dargelegt, dass eine solche Architektur extrem schwierig zu realisieren ist.

Austauschpunkt-Dienst muss mächtig sein

Die Methodik dieser Arbeit kann in diesem Fall helfen, um eine individuelle „Beinahe-Umbrella-Architektur“ sukzessive zu entwickeln. Beim Aufbau eines QoS-fähigen zentralen Austauschpunktes sollte der Betreiber mit einer per se bereits sehr mächtigen Architektur beginnen. Mit Hinzukommen von Kunden kann die Methodik dann eingesetzt werden, um den jeweiligen Erweiterungsbedarf im Austauschpunkt zu ermitteln. Durch die sukzessive Hinzunahme weiterer Kunden kann dann der Austauschpunkt weiterentwickelt werden. Allerdings muss angemerkt werden, dass eine beliebige Weiterentwicklung des Austauschpunktes mit einer wachsenden Zahl an verschiedenen Diensten technisch sehr problematisch werden kann.

Sukzessiver Aufbau des Austauschpunktes

Für den Betreiber des Austauschpunktes bleibt bei konsequenter Verwendung der Methodik ein Vorteil. Der Aufwand zur Koppelung jener Kunden sinkt, die Dienste verwenden, die zu einem bereits gekoppelten Dienst identisch oder ähnlich sind. Dies ist der Fall, weil die notwendigen Teillösungen und deren Umsetzung bereits einmal durchgeführt worden sind und dies durch die Methodik dokumentiert wurde.

Erarbeitete Teillösungen durch Methodik wiederverwendbar

ZUSAMMENFASSUNG UND AUSBLICK

Die Koppelung von Netz-QoS-Architekturen und von Diensten, die auf solchen Architekturen basieren, ist heute ein Problem, das in seiner ganzen Allgemeinheit noch nicht gelöst ist. Diese Tatsache steht im scharfen Gegensatz zu den Erfordernissen, die durch die Realität gestellt werden. Immer mehr Anwendungen benötigen für ihre korrekte Funktion Qualitätssicherungen aus den Netzen, auf denen sie aufbauen. Prominente Beispiele sind Multimedia-Anwendungen wie Videotelefonie und Videostreaming.

Zu Beginn dieser Arbeit wurde eine Einordnung der Koppelungsproblematik vorgenommen. Es wurde dabei festgestellt, dass die Vielschichtigkeit der Koppelungsproblematik zuerst einmal in der großen Spannbreite dessen begründet liegt, was man unter einer Netz-QoS-Architektur verstehen kann. Auf dem Markt sind nahezu alle Abstufungen zu finden; von „schwachen“ Architekturen wie DiffServ, die nur relative Garantien bieten, bis hin zu „starken“ Varianten wie ATM. Wie weiterhin festgestellt wurde, trägt auch die Tatsache, dass ein und dieselbe Architektur unterschiedlich in Diensten umgesetzt werden kann, ihr Übriges zur Verschärfung der Problematik bei.

Im weiteren Verlauf wurde dann aufgezeigt, dass die bisherigen Ansätze aus Standardisierungsgremien und Forschung keinen systematischen Beitrag liefern, der es ermöglichen würde, die Koppelungsproblematik dienstorientiert, nachhaltig und generisch zu lösen. Wie ebenfalls aufgezeigt wurde, wäre der Versuch einen Ansatz zu verfolgen, der eine universell anwendbare und gleichzeitig monolithische Lösung zum Ziel hat, höchstwahrscheinlich zum Scheitern verurteilt. Deshalb wurde in dieser Arbeit ein anderer Weg beschritten. Der Ansatz dieser Arbeit war die Entwicklung einer generischen Methodik, die ermöglicht, für unterschiedlichste Anwendungsfälle schnell eine spezifische Lösung zu entwerfen.

Um eine Ausgangsbasis für eine generische Methodik zu schaffen, wurde zunächst ein generisches Modell für Netz-QoS-Architekturen entwickelt. Dieses Modell ist strikt dienstorientiert und stellt eine Architektur in einer Black-Box-Sicht dar, in der nur noch jene Teile sichtbar sind, die von einem Nutzer wahrgenommen werden können. Das entwickelte Modell kann dabei nicht nur die statischen Teile einer Netz-QoS-Architektur darstellen, auch die dynamischen Vorgänge über den gesamten Lebenszyklus hinweg finden im Modell Beachtung. Auf dem Architektur-Modell aufbauend wurde dann ein

Modell entwickelt, das einen Netz-QoS-Dienst, ebenfalls dienstorientiert, darstellt. Durch die zweistufige Modellierung wurde der Tatsache Rechnung getragen, dass die Erscheinung einer Netz-QoS-Architektur bei verschiedenen Diensten unterschiedlich gestaltet sein kann.

Nachdem mit der Modellierung die Ausgangsbasis gelegt wurde, ist der eigentlich Lösungsansatz – die generische Methodik – entwickelt worden. Ausgehend von den generischen Modellen, wurden die einzelnen Teilschritte zur Koppelung von Netz-QoS-Diensten systematisch vorgestellt und besprochen. Durch die einzelnen Teilschritte werden Teillösungen gewonnen. Die Methodik zeichnet sich dabei dadurch aus, dass in den einzelnen Teilschritten kein starres Schema vorgegeben wird, sondern an allen Stellen die Spezifika des konkreten Anwendungsfalls berücksichtigt werden können. Dieser Umstand drückt sich auch darin aus, dass für viele Teilprobleme mehrere Lösungsmöglichkeiten angegeben werden konnten. Zum Abschluss der Methodik wurde dann geschildert, welche weiteren Schritte von der Menge der gewonnenen Teillösungen hin zu einer komplett implementierten Koppelungslösung zu beschreiten sind. Zusammenfassend konnte dann festgestellt werden, dass die Einheit aus den entwickelten Modellen und der generischen Methodik alle an sie gestellten Anforderungen erfüllt.

Zum Abschluss der Arbeit wurden dann noch konkrete Anwendungsbereiche der Modelle und der Methodik diskutiert. Für eine Ergebnisabschätzung einer Koppelung wurde eine beschleunigte Anwendung der Methodik vorgestellt. Weiterhin wurde gezeigt, dass die entwickelten Modelle eine Taxonomie für Netz-QoS-Architekturen und -Dienste liefern. Außerdem wurde auch die Frage geklärt, wie sich die bestehenden Arbeiten in die Methodik dieser Arbeit integrieren lassen.

AUSBLICK AUF WEITERFÜHRENDEN ARBEITEN

Die vorliegende Arbeit klärt zwar viele wichtige Fragestellungen im Bereich der Koppelung von Netz-QoS-Architekturen und Netz-QoS-Diensten. Dennoch ist die Gesamtproblematik bei weitem noch nicht endgültig erschlossen.

Zunächst ist im Rahmen des Ansatzes der Arbeit selbst noch Spielraum für fortführende Arbeiten:

- Adressierung und Routing ■ Die Themenfelder Adressierung und Routing wurden aus Gründen des Umfangs explizit von der Bearbeitung ausgenommen. Diese beiden Themengebiete könnten sowohl in die Modellierung als auch in die Methodik integriert werden.
- Ereignisabbildung ■ Im Rahmen der Methodik wären weiterreichende Untersuchungen im Bereich der Ereignismodellierung und Ereignisabbildung wichtig. Wenn

es gelänge dort konkretere Modelle und Arbeitsabläufe zu spezifizieren, dann würde die Methodik noch weiter an Präzision gewinnen.

- Weitere Arbeiten könnten sich mit der expliziten Integration der Arbeit von Nerb in die Modellierung und die Methodik beschäftigen. Bis jetzt wurde auf die Arbeit von Nerb nur verwiesen. Würde durch die Integration von [Nerb 01] der Bereich des CSM unmittelbar in Modellierung und die Methodik zur Verfügung stehen, so wäre dies eine weitere Aufwertung des gesamten Ansatzes. Explizites CSM
- Gleiches gilt für den Bereich des SLA. Hier könnte sich eine Arbeit mit der expliziten Integration des Ansatzes von H. Schmidt auseinandersetzen. Explizites SLA
- Durch die bereits sehr formalisierte Darstellung von Modellierung und Methodik wurde die Grundlage für die Entwicklung einer Toolunterstützung gelegt. Ein Tool ist denkbar zur Unterstützung der Instanziierung des (Meta-) Modells für konkrete Architekturen und Dienste. Hierfür müsste u.a. eine maschinengerechte Repräsentierung der Modellierungen entwickelt werden. Darauf aufbauend ist ein weiteres Tool vorstellbar, das bei der Anwendung der Methodik behilflich sein könnte. Ein konkretes Tool könnte sich in Konzeption und Implementierung an dem bereits realisierten Tool für die MNM-Dienstmodellmethodik aus [Schm 02a] orientieren. Toolunterstützung

Die Anwendung von Modellierung und Methodik gibt ebenfalls Raum für weitere Arbeiten:

- Da die Methodik sehr generisch gehalten ist, wäre zu überprüfen, inwiefern ihr Anwendungsgebiet erweitert werden kann. Denkbar wären zwei Richtungen:

„Vertikal“ entlang des OSI-Schichtenmodells: In der vorliegenden Arbeit werden Protokolle der Anwendungsschicht explizit ausgenommen. Eine fortführende Arbeit könnte überprüfen, welche Teile der Methodik für Problemstellungen der Anwendungsschicht und darüber hinaus geeignet sind.

„Horizontal“ im Netzverbund: Die Methodik spart bis jetzt die Endsysteme aus. Eine weitere Arbeit könnte diese Beschränkung aufheben und die Endsysteme in die Methodik miteinbeziehen.
- Die Modellierung der Arbeit kann benutzt werden, um eine systematische Aufarbeitung aller bisherigen Netz-QoS-Architekturen zu beginnen. Dadurch entstünde eine Art Datenbank der dienstorientierten Sichten auf existierende Netz-QoS-Architekturen. Ausgedehnte Anwendung
- Analog könnten aus den bisherigen Forschungsarbeiten systematisch die darin enthaltenen Teillösungen extrahiert und abstrahiert werden. Dadurch würden sie in Kombination mit der Methodik einer nachhaltigen und breiteren Verwendbarkeit zugeführt werden. Aufbau einer Architektur-Datenbank
- Analog könnten aus den bisherigen Forschungsarbeiten systematisch die darin enthaltenen Teillösungen extrahiert und abstrahiert werden. Dadurch würden sie in Kombination mit der Methodik einer nachhaltigen und breiteren Verwendbarkeit zugeführt werden. Aufbau einer Teillösungs-Datenbank

KAPITEL 7: ZUSAMMENFASSUNG UND AUSBLICK

Die eben genannten weiterführenden Arbeiten sollen nur als Beispiele dienen, die zeigen, wie die vorliegende Arbeit fortgeführt werden kann. Selbstverständlich sind weitere Themengebiete denkbar.

Mit den genannten Beispielen wurde zum Abschluss gezeigt, dass die Arbeit nicht nur einen eigenen Beitrag leistet, sondern auch das Potential zur Stimulierung weiterer Forschungsaktivitäten birgt.

ANHANG A

STATISCHES MODELL DER UMTS-QoS-ARCHITEKTUR REVISION 6

In diesem Anhang wird ohne nähere Erläuterung das statische Modell der QoS-Architektur von UMTS Revision 6 modelliert. Die zugrunde liegenden Standards sind:

- Für Unicast: [[TS 123 107](#), [TS 123 207](#)]
- Für Multicast: **Multimedia Broadcast / Multicast Service (MBMS)**
[[TS 122 246](#), [TS 123 246](#)]

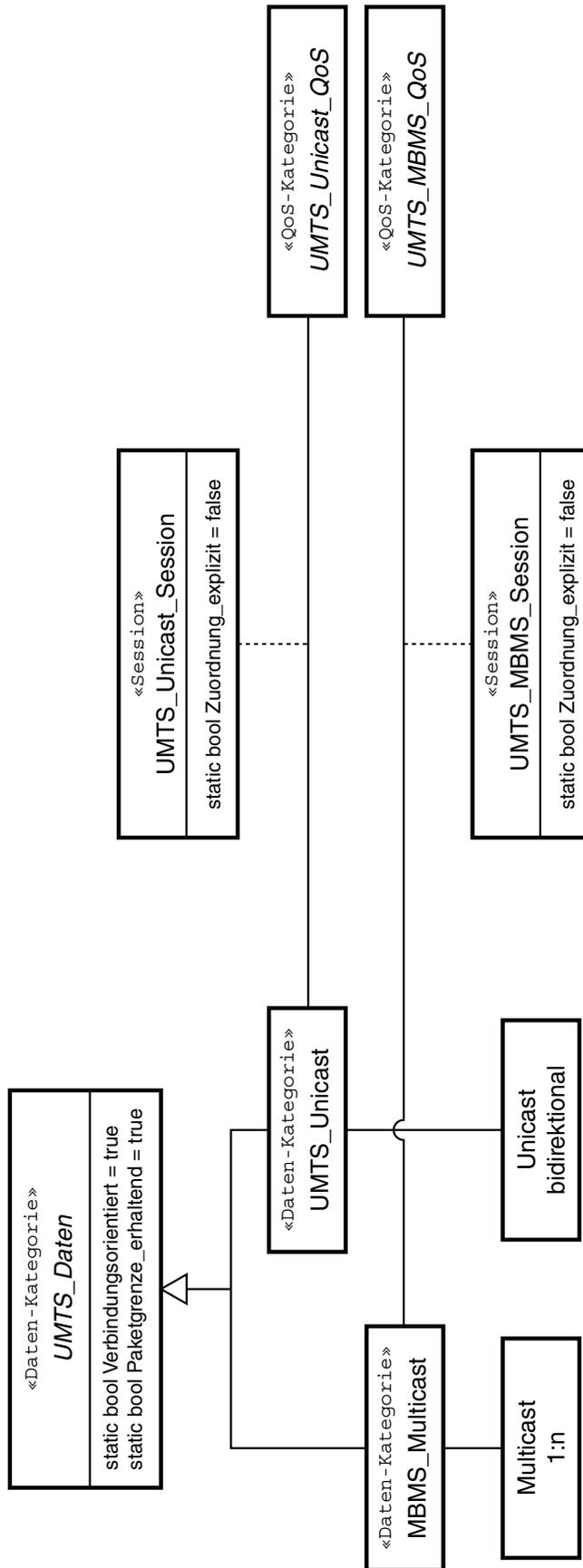


Abbildung A.1: Daten-Kategorien von UMTS

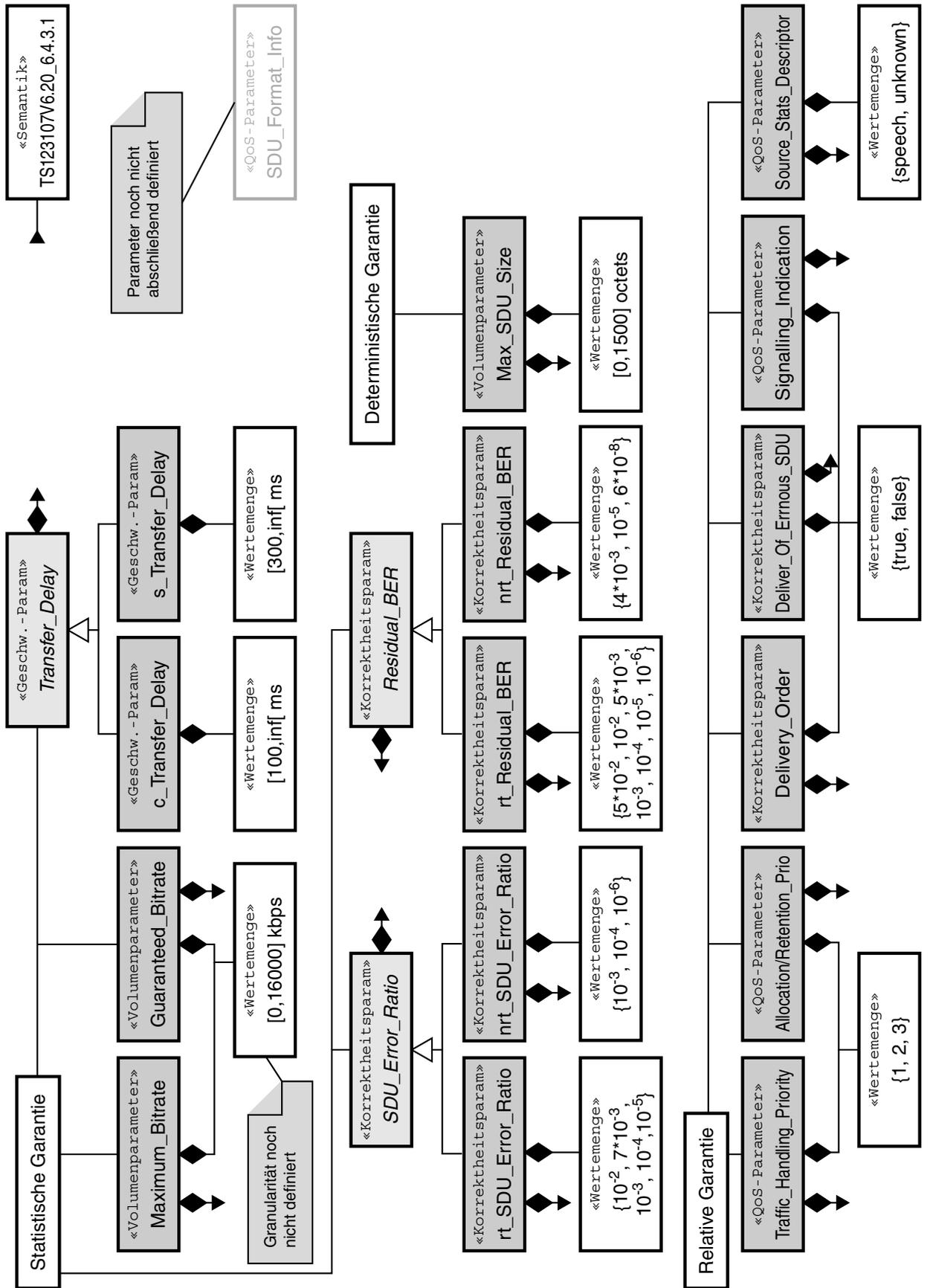


Abbildung A.2: QoS-Parameter von UMTS Unicast

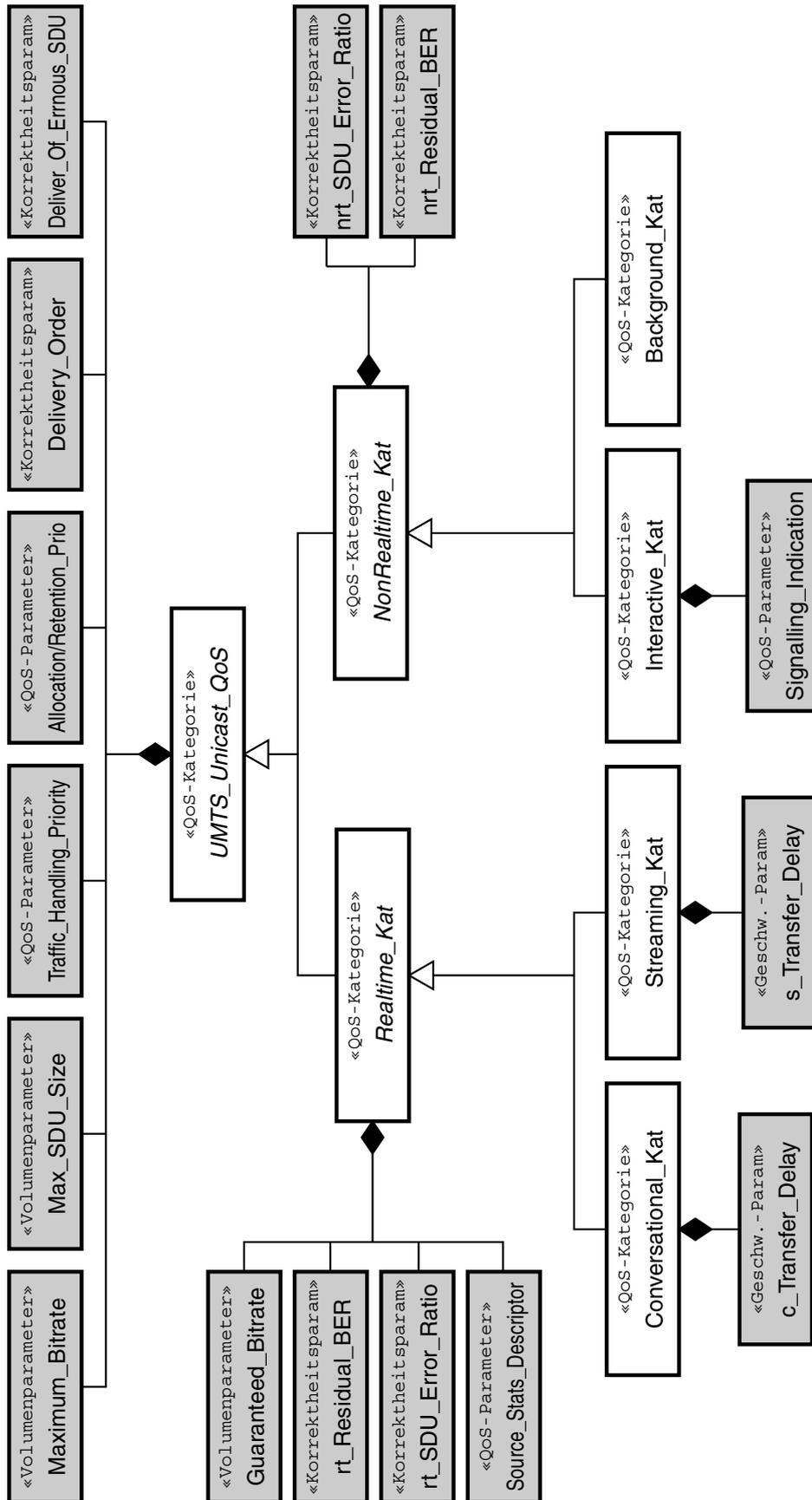


Abbildung A.3: QoS-Kategorien von UMTS Unicast

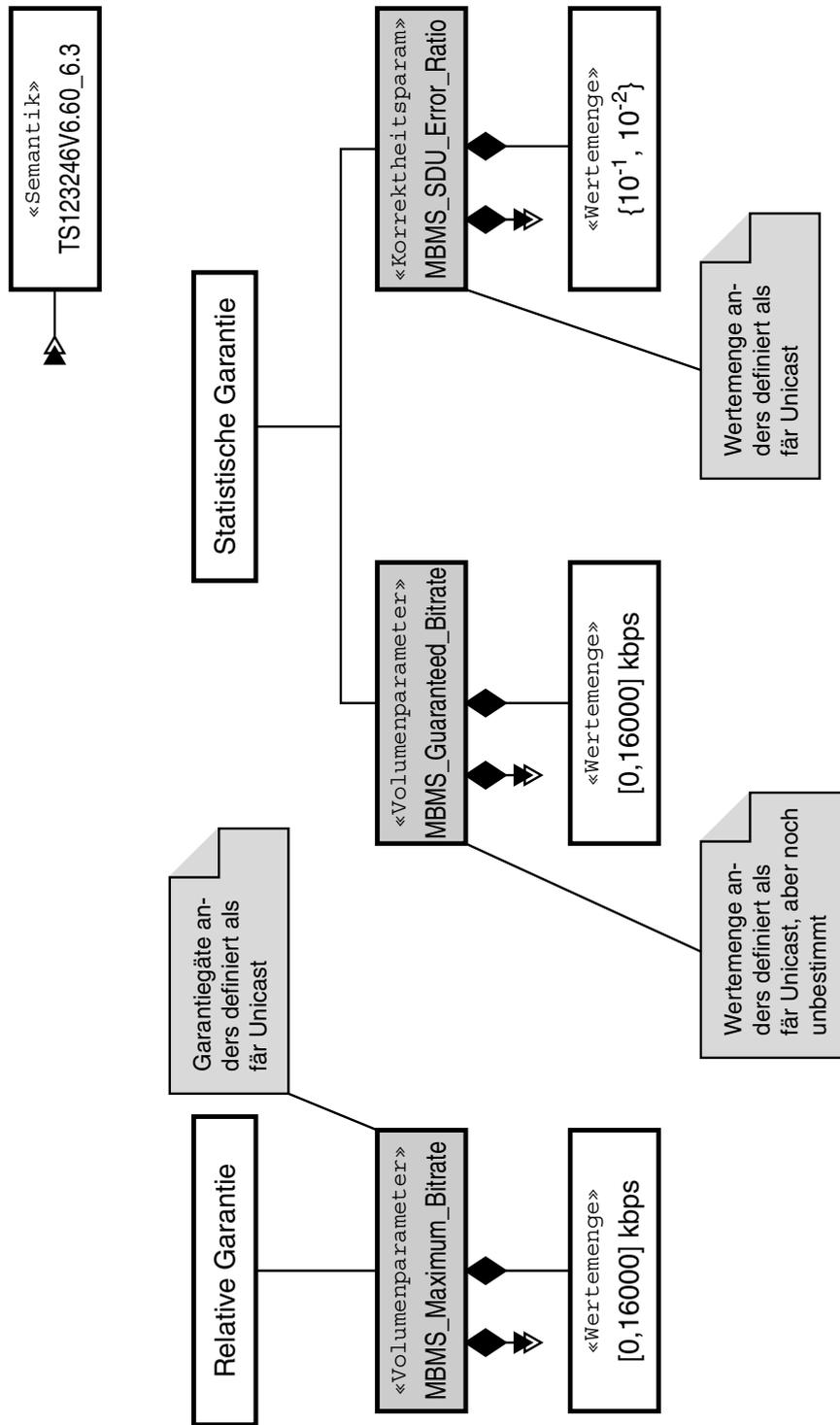


Abbildung A.4: QoS-Parameter von UMTS MBMS

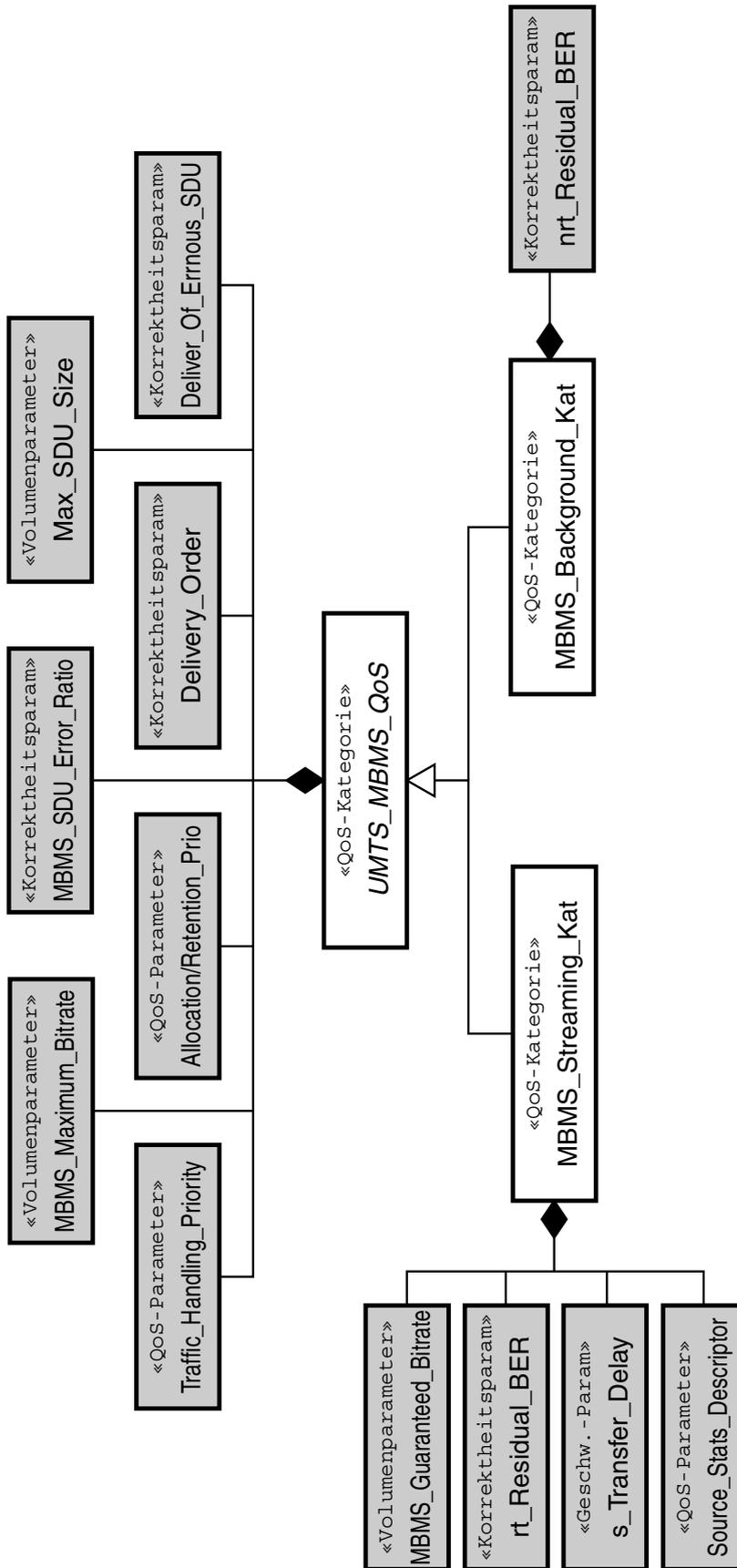


Abbildung A.5: QoS-Kategorien von UMTS MBMS

ABKÜRZUNGEN

ABR	Available Bit Rate
AF	Assured Forwarding
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CBR	Constant Bit Rate
CLR	Cell Loss Rate
CoS	Class-of-Service
CSM	Customer Service Management
DFN	Deutsches Forschungsnetz
DiffServ	Differentiated Services
DS	DiffServ
DSCP	DiffServ Codepoint
EBNF	Extended Backus-Naur Form
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
ER	Entity Relationship
ETSI	European Telecommunications Standards Institute
FEC	Forwarding Equivalency Class
FR	Frame Relay
G-WiN	Gigabit-Wissenschaftsnetz

ABKÜRZUNGEN

GFR	Guaranteed Frame Rate
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union – Telecom Standardization
LAL	Layer - Aspect - Life cycle
LAN	Local Area Network
LANE	LAN Emulation Over ATM
LDP	Label Distribution Protocol
maxCDT	maximum Cell Transfer Delay
MBMS	Multimedia Broadcast / Multicast Service
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MFS	Maximum Frame Size
MNM	Munich Network Management Team
MPLS	Multi Protocol Label Switching
MPOA	Multi-Protocol Over ATM
NNI	Network-Network Interface
nrtVBR	non-real-time Variable Bit Rate
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PCR	Peak Cell Rate
PDF	Portable Document Format

peakToPeakCDV	peak-to-peak Cell Delay Variation
PHB	Per Hop Behaviour
PoP	Point-of-Presence
QoS	Quality-of-Service
RFC	Request For Comments
RIP	Routing Information Protocol
RSVP	Resource Reservation Setup Protocol
rtVBR	real-time Variable Bit Rate
SAP	Service Access Point
SCR	Sustained Cell Rate
SDU	Service Data Unit
SLA	Service Level Agreement
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time To Live
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
UNI	User-Network Interface
VCI	Virtual Channel Identifier
VoIP	Voice-over-IP
VPI	Virtual Path Identifier
VPN	Virtual-Private-Network

ABKÜRZUNGEN

ABBILDUNGSVERZEICHNIS

1.1	Grobschichtenmodell verteilter IT-Anwendungen	2
1.2	Vorgehensmodell der Arbeit im Überblick	8
2.1	Standortvernetzung mit separaten Verbindungen	12
2.2	Standortvernetzung mit gemeinsamen Verbindungen	13
2.3	Ankoppelung eines neuen Zugangsproviders	15
2.4	Exemplarisches Schema eines hierarchischen Netzverbundes	17
2.5	Grobschichtenmodell verteilter IT-Anwendungen	19
2.6	Grobmodellierung von Netz-QoS-Architekturen	23
2.7	Basis-Modell (<i>Basic Model</i>) des MNM-Dienstmodells (nach [GHKR+ 01])	25
2.8	Basis-Modell für das Szenario der VoIP-Koppelung	26
2.9	Beziehungen zwischen Basisbegriffen	28
2.10	Dimensionen der Koppelungsproblematik	29
2.11	Koppelung wesentlicher Teilbereiche von Netz-QoS-Architekturen	30
2.12	Individuallösung zwischen zwei Netz-QoS-Diensten	36
2.13	Umbrella-Koppelung umspannt beliebige Netz-QoS-Dienste	37
2.14	Extreme für einen Lösungsansatz	39
2.15	Übersicht über Methodik-basierten Lösungsansatz	41
3.1	Konzeptuelles ER-Modell nach [SchmJB 01]	52
3.2	Grundidee von MPLS (nach [Roel 01])	62
3.3	MPLS im OSI-Schichtenmodell (nach [Roel 01])	63
3.4	LAL-Kubus (nach [GaRo 04])	65
4.1	Aktivitätsdiagramm für das Vorgehen in diesem Kapitel	68
4.2	Aktivitätsdiagramm der Methodikanwendung	70
4.3	Ebenen Begriffe, Modell und reale Instanzen	71
4.4	Dienstsicht (<i>Service View</i>) des MNM-Dienstmodells	73
4.5	Exemplarische Detailsicht auf Netzverbund	77
4.6	Black-Box-Sicht auf Domänen des Netzverbunds aus Abb. 4.5	79
4.7	Statisches Grundmodell für Netz-QoS-Parameter	83
4.8	Ebenen der Modellierung	84
4.9	Klasse <i>Daten-Kategorie</i> des statischen Grundmodells	86
4.10	IP Daten-Kategorien von DiffServ	90
4.11	Klasse <i>QoS-Parameter</i> des statischen Grundmodells	91
4.12	Statisches Modell des <i>Garantietyps</i> eines <i>QoS-Parameters</i> (nach [SchmJB 01])	92

ABBILDUNGSVERZEICHNIS

4.13	Vollständiges Modell für <i>QoS-Parameter</i> mit Standard-Parameter	94
4.14	«QoS-Parameter» von DiffServ mit Assured Forwarding PHB	95
4.15	Restliche Standard-Parameter von IP	96
4.16	«QoS-Kategorien» von DiffServ mit Assured Forwarding PHB	98
4.17	«QoS-Kategorien» von ATM	101
4.18	Vollständiges statisches Grundmodell ohne Standard-Parameter	104
4.19	Statisches Grundmodell von DiffServ mit Assured Forwarding PHB	105
4.20	Lebenszyklus von Netz-QoS-Architekturen	106
4.21	IP «Daten-Kategorien» mit Singleton-Notation	112
4.22	Singleton-«QoS-Kategorie»	113
4.23	Aufgefaltene Singleton-«QoS-Kategorie»	114
4.24	Aufgefaltene «QoS-Kategorien» von DiffServ AF PHB	115
4.25	Kardinalitäten: Jeweils genaue eine Instanz	116
4.26	Kardinalitäten: Alle <i>QoS-Kategorien</i> teilen sich genau eine <i>Daten-Kategorie</i> .	116
4.27	Kardinalitäten: Alle <i>Daten-Kategorien</i> teilen sich genau eine <i>QoS-Kategorie</i> .	117
4.28	Kardinalitäten: Jeweils eine <i>Daten-Kategorie</i> mit einer <i>QoS-Kategorie</i>	117
4.29	Aufgeteilte Klasse <i>QoS parameters</i>	127
4.30	Erweitertes MNM-Dienstmodell in der Dienstsicht	129
5.1	Aktivitäten zur gesamten Methodik	136
5.2	Aktivitäten zur Modellierung	137
5.3	Aktivitäten zur Architektur-Modellierung	138
5.4	Aktivitäten zur Dienstmodellierung	139
5.5	Grundanwendungsfälle	141
5.6	Aktivitäten zur Feststellung grundlegender Koppelungseigenschaften	143
5.7	Allgemeines Aktivitätsmuster für Ermittlung optimaler Paarungen	147
5.8	Allgemeines Aktivitätsmuster für Koppelungssymmetrien	148
5.9	Aktivitäten zum Vergleich statischer Modellteile	153
5.10	Aktivitäten zur Koppelung der Klasse <i>Daten-Kategorie</i>	154
5.11	Aktivitäten zur Koppelung des Attributs <i>Verbindungsorientiert</i>	159
5.12	Aktivitäten zur Koppelung der Klasse <i>Kommunikationsrelation</i>	167
5.13	Aktivitäten zur Koppelung des Attributs <i>Paketgrenze_erhaltend</i>	171
5.14	Aktivitäten zur Koppelung der Klasse <i>QoS-Kategorie</i>	172
5.15	Aktivitäten zur Koppelung der Klasse <i>QoS-Parameter</i>	174
5.16	Aktivitäten zur Koppelung der Klasse <i>Semantik</i>	176
5.17	Aktivitäten zur Koppelung der Klasse <i>Garantietyp</i>	178
5.18	Aktivitäten zur Koppelung der Klasse <i>Wertemenge</i>	179
5.19	Aktivitäten zur Behandlung ungepaarter <i>QoS-Parameter</i>	187
5.20	Aktivitäten zum Vergleich dynamischer Modellteile	188
5.21	Aktivitäten zur Anpassung der Kardinalitäten	189
5.22	Aktivitäten zur Anpassung der Kardinalitäten von <i>Daten-Kategorien</i>	194
5.23	Aktivitäten zur Abstimmung von Ereignissen	195
5.24	Aktivitäten zum Abschluss der Methodik	198
5.25	Aktivitäten zur Vervollständigung der Koppelung	199
5.26	Aktivitäten zu Projektplanung und Dienstvereinbarung	201
5.27	Überblick über die gesamte Methodik	205

6.1	Ausriss aus Abbildung 4.19	213
6.2	Varianten zum RSVP → ATM Verbindungsaufbau aus [KZS oo]	216
A.1	Daten-Kategorien von UMTS	228
A.2	QoS-Parameter von UMTS Unicast	229
A.3	QoS-Kategorien von UMTS Unicast	230
A.4	QoS-Parameter von UMTS MBMS	231
A.5	QoS-Kategorien von UMTS MBMS	232

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

2.1	QoS-Parameter der ATM QoS-Kategorien (nach [ATM-TM 99])	21
2.2	Erfüllung von Anforderungen durch prinzipielle Lösungsansätze	39
3.1	Erfüllung von Anforderungen durch [Schm]B 01]	54
3.2	Erfüllung der Anforderungen durch den Status quo im Überblick	60
4.1	Zeitpunkte im Lebenszyklus	109
4.2	Korridore der Grundvorgänge	119
4.3	Erfüllung der Anforderungen durch Modelle dieser Arbeit	133
5.1	Komponenten von Bereichs- und Einzelfallbezeichnern	145
5.2	Koppelungsmuster für Attribut <i>Verbindungsorientiert</i>	159
5.3	Koppelungsmuster für Kommunikationsrelationen <i>Unicast / Unicast</i>	162
5.4	Koppelungsmuster für Kommunikationsrelationen <i>Multicast / Multicast</i>	163
5.5	Koppelungsmuster für Kommunikationsrelationen <i>Unicast / Multicast</i>	166
5.6	Koppelungsmuster für Kommunikationsrelationen <i>Multicast / Unicast</i>	166
5.7	Koppelungsmuster für Attribut <i>Paketgrenze_erhaltend</i>	171
5.8	Koppelungsmuster für Klasse <i>Semantik</i>	176
5.9	Koppelungsmuster für Klasse <i>Garantietyp</i>	177
5.10	Koppelungsmuster für Anpassungen der Klasse <i>Wertemenge</i>	180
5.11	Koppelungsmuster für Abbildungen der Klasse <i>Wertemenge</i>	182
5.12	Koppelungsmuster für Klasse <i>Garantietyp</i>	185
5.13	Koppelungsmuster für ungepaarte <i>QoS-Parameter</i>	187
5.14	Koppelungsmuster für Kardinalitäten der Klasse <i>Daten-Kategorie</i>	193
5.15	Koppelungsmuster für Kardinalitäten der Klasse <i>QoS-Kategorie</i>	194
5.16	Erfüllung der Anforderungen durch diese Arbeit	208

TABELLENVERZEICHNIS

LITERATUR

- [ATM-ARCH 02] ATM-FORUM: *ATM User – Network Interface (UNI) Specification Version 4.1*. Specification af-arch-0193.000, ATM-Forum, November 2002, <ftp://ftp.atmforum.com/pub/approved-specs/af-arch-0193.000.pdf>.
- [ATMForum-HP] *ATM-Forum*, <http://www.atmforum.com>. Homepage.
- [ATM-LANE 98] ATM-FORUM: *LAN Emulation Client Management Specification Version 2.0*. Specification af-lane-0093.000, ATM-Forum, Oktober 1998, <ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0093.000.pdf>.
- [ATM-LUNI 97] ATM-FORUM: *LAN Emulation over ATM, Version 2 – LUNI Specification*. Specification af-lane-0084.000, ATM-Forum, Juli 1997, <ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0084.000.pdf>.
- [ATM-MPOA 99] ATM-FORUM: *Multi-Protocol Over ATM Version 1.1*. Specification af-mpoa-0114.000, ATM-Forum, Mai 1999, <ftp://ftp.atmforum.com/pub/approved-specs/af-mpoa-0114.000.pdf>.
- [ATM-PHY 01] ATM-FORUM: *Cell-Based 1000 Mbit/s (CB1G) Physical Layer Specification over Single-mode or Multi-mode Fiber and Category 6 Twisted Pair Copper Cabling*. Specification af-phy-0162.000, ATM-Forum, April 2001, <ftp://ftp.atmforum.com/pub/approved-specs/af-phy-0162.000.pdf>.
- [ATM-PNNI 02] ATM-FORUM: *Private Network – Network Interface Specification Version 1.1 (PNNI 1.1)*. Specification af-pnni-0055.002, ATM-Forum, April 2002, <ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.002.pdf>.
- [ATM-TM 99] ATM-FORUM: *Traffic Management Specification Version 4.1*. Specification af-tm-0121.000, ATM-Forum, März 1999, <ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0121.000.pdf>.
- [ATM-UNI 02] ATM-FORUM: *ATM User – Network Interface (UNI) Signalling Specification Version 4.1*. Specification af-sig-0061.002, ATM-Forum, April 2002, <ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.002.pdf>.

LITERATUR

- [BaBa 04] BAHATI, RAPHAEL M. und MICHAEL A. BAUER: *Quality of Service Provisioning for VoIP applications with Policy-Enabled Differentiated Services*. In: *Managing Next Generation Convergence Networks and Services: Proceedings of the 9th IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, Korea, April 2004. IFIP/IEEE, http://www.comsoc.org/tech_focus/pdfs/2005/jan/0335baha.pdf.
- [BBS 98] BAJAJ, SANDEEP, LEE BRESLAU und SCOTT SHENKER: *Is Service Priority Useful in Networks?* In: *Proceedings of ACM SIGMETRICS '98/PERFORMANCE '98 Joint International Conference on Measurement and Modeling of Computer Systems*, Seiten 66–77, Madison, WI, USA, Juni 1998. ACM, <http://www.cs.wpi.edu/~sigmet98/bajaj.ps>.
- [BCCE+ M4] BURGSTÄHLER, LARS, GEORG CARLE, PAUL CHRIST, LUTZ ENDER, TASSILO ERLEWEIN, YURI GLICKMAN, JÜRGEN JÄHNERT, HYUNG-WOO KIM, LUTZ MARK, JENS TIEMANN, RUDOLF ROTH, VIJAI KRISHNA SRIVIVASAN und TANJA ZSEBY: *Milestone 4: Quasar QoS Architecture – First Test Results*. Projektbericht, Projekt QUASAR, 2002, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M4.pdf>.
- [BCCE+ M5] BURGSTÄHLER, LARS, CEMAL COEMERT, PAUL CHRIST, TASSILO ERLEWEIN, JÜRGEN JÄHNERT, HYUNG-WOO KIM, LUTZ MARK, JENS TIEMANN, RUDOLF ROTH, CARSTEN SCHMOLL, VIJAI KRISHNA SRIVIVASAN und TANJA ZSEBY: *Milestone 5: QUASAR QoS Architecture*. Projektbericht, Projekt QUASAR, Juni 2002, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M5.pdf>.
- [BCCG+ M3] BURGSTÄHLER, LARS, GEORG CARLE, PAUL CHRIST, YURI GLICKMAN, JÜRGEN JÄHNERT, LUTZ MARK, JENS TIEMANN, RUDOLF ROTH, VIJAI KRISHNA SRIVIVASAN und TANJA ZSEBY: *Milestone 3: Preliminary QoS Architecture*. Projektbericht, Projekt QUASAR, jan 2002, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M3.pdf>.
- [BCCK+ M6] BURGSTÄHLER, LARS, CEMAL COEMERT, PAUL CHRIST, HYUNG-WOO KIM, LUTZ MARK, JENS TIEMANN, RUDOLF ROTH, CARSTEN SCHMOLL und TANJA ZSEBY: *Milestone 6: QUASAR Final Report*. Projektbericht, Projekt QUASAR, dec 2002, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M6.pdf>.
- [BCJM+ M2] BURGSTÄHLER, LARS, GEORG CARLE, JÜRGEN JÄHNERT, LUTZ MARK, JENS TIEMANN, RUDOLF ROTH und TANJA ZSEBY: *Milestone 2: QUASAR QoS Reference Model*. Projektbericht, Projekt QUASAR, 2001, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M2.pdf>.
- [BHPB+ 00] BROY, M., H.-G. HEGERING, A. PICOT, A. BUTTERMANN, M. GARSCHHAMMER, R. HAUCK und S. VOGEL: *Kommunikations- und Informationstechnik 2010 Trends in Technologie und Markt*. SecuMedia Verlag, Ingelheim, Deutschland, September 2000. ISBN 3-922746-35-7.

- [BJMT+ M1] BURGSTÄHLER, LARS, JÜRGEN JÄHNERT, LUTZ MARK, JENS TIEMANN und RUDOLF ROTH: *Milestone 1: Technology Overview*. Projektbericht, Projekt QUASAR, mar 2001, <http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M1.pdf> .
- [Blac 02] BLACK, UYLESS D.: *Voice Over IP (2nd Edition)*. Prentice Hall Series in Advanced Communications Technologies. Prentice Hall, 2nd Edition Auflage, 2002. ISBN 0-13-065204-0.
- [Bles 04] BLESS, ROLAND: *Towards Scalable Management of QoS-based End-to-End Services*. In: *Managing Next Generation Convergence Networks and Services: Proceedings of the 9th IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, Korea, April 2004. IFIP/IEEE, <http://doc.tm.uka.de/2004/noms04-crc-rb-tm-uka-de.pdf> .
- [BrSh 98] BRESLAU, LEE und SCOTT SHENKER: *Best-Effort versus Reservations: A Simple Comparative Analysis*. In: *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, Seiten 3–16, Vancouver, British Columbia, Canada, 1998. ACM, ACM Press, <http://www.cs.columbia.edu/~zwb/my/oral/qos/sigcomm98/bs.pdf> .
- [BSFR 05] BLEFARI-MELAZZI, N., D. DI SORTE, M. FEMMINELLA und G. REALI: *A tariff model to charge IP services with guaranteed quality: effect of users' demand in a case study*. In: CLEMM, A., O. FECTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE, http://www.ist-simplicity.org/_publications/C14Atariffmodelto.pdf .
- [Camp 96] CAMPBELL, ANDREW T.: *A Quality of Service Architecture*. PhD Thesis, Lancaster University, England, Januar 1996, <http://comet.ctr.columbia.edu/~campbell/papers/thesis.pdf> .
- [CDZ 97] CALVERT, KENNETH L., MATTHEW B. DOAR und ELLEN W. ZEGURA: *Modeling Internet Topology*. IEEE Communications Magazine, 35(6):160–163, June 1997, <http://www.cc.gatech.edu/fac/Ellen.Zegura/papers/model.ps.gz> .
- [CrJa 02] CRISTALLO, GEOFFREY und CHRISTIAN JACQUENET: *An Approach to Inter-domain Traffic Engineering*. In: *Proceedings of XVIII World Telecommunications Congress (WTC 2002)*, Paris, Frankreich, 2002. , <http://www.ist-tequila.org/publications/wtc2002-idte.pdf> . Projekt TEQUILA.
- [DaRe 00] DAVIE, BRUCE und YAKOV REKHTER: *MPLS – Technology and Applications*. Morgan Kaufmann Publishers, 2000. ISBN 1-55860-656-4.
- [DE-CIX-HP] DE-CIX, <http://www.decix.de/> . Homepage.

LITERATUR

- [DGCG+ D3-4a] DAMILATIS, T. (HERAUSG.), D. GODERIS, G. CRYSTALLO, P. GEORGATSOS, T. DAMILATIS, M. MEGALOOIKONOMOU, C. JACQUENET, M. BOUCADAI, S. VAN DEN BERGHE, P. VAN HEUVEN, E. MYKONIATI, D. GIANNAKOPOULOS, M. MAUROGIORGIS, H. ASGARI, M. IRONS, R. EGAN, D. GRIFFIN, M. FENG, J. GRIEM, P. TRIMINTZIOS, P. FLEGKAS und G. PAVLOU: *Final System Evaluation – Part A: Tests and Results*. Projektbericht D3-4a, TEQUILA Consortium, Oktober 2002, <http://www.ist-tequila.org/deliverables/D3-4a.pdf>.
- [DGCG+ D3-4b] DAMILATIS, T. (HERAUSG.), D. GODERIS, G. CRYSTALLO, P. GEORGATSOS, T. DAMILATIS, M. MEGALOOIKONOMOU, C. JACQUENET, M. BOUCADAI, C. RISCHETTE, C. DURET, J. LATTMANN, V. LASPRESES, VAN DEN BERGHE, P. VAN HEUVEN, E. MYKONIATI, D. GIANNAKOPOULOS, M. MAUROGIORGIS, H. ASGARI, M. IRONS, R. EGAN, D. GRIFFIN, M. FENG, P. TRIMINTZIOS, P. FLEGKAS und G. PAVLOU: *Final System Evaluation – Part B: Final Architecture, Protocol and Algorithm Specification*. Projektbericht D3-4b, TEQUILA Consortium, Oktober 2002, <http://www.ist-tequila.org/deliverables/D3-4b.pdf>.
- [DGCG+ D3-4c] DAMILATIS, T. (HERAUSG.), D. GODERIS, G. CRYSTALLO, P. GEORGATSOS, T. DAMILATIS, M. MEGALOOIKONOMOU, C. JACQUENET, M. BOUCADAI, C. RISCHETTE, C. DURET, J. LATTMANN, V. LASPRESES, S. VAN DEN BERGHE, P. VAN HEUVEN, E. MYKONIATI, D. GIANNAKOPOULOS, M. MAUROGIORGIS, H. ASGARI, M. IRONS, R. EGAN, D. GRIFFIN, P. TRIMINTZIOS, P. FLEGKAS und G. PAVLOU: *Final System Evaluation – Part C: Scalability and Stability Analysis*. Projektbericht D3-4c, TEQUILA Consortium, Oktober 2002, <http://www.ist-tequila.org/deliverables/D3-4c.pdf>.
- [DiffServ-WG] *Differentiated Services (diffserv)*, <http://www.ietf.org/html.charters/diffserv-charter.html>. IETF Working Group.
- [DPG 00] DAVIDSON, JONATHAN, JAMES PETERS und BRIAN GRACELY: *Voice over IP Fundamentals*. Cisco Press, 2000. ISBN 1578701686.
- [Dreo 02] RODOSEK, GABRIJELA DREO: *A Framework for IT Service Management*. Habilitation, Ludwig-Maximilians-Universität München, Juni 2002.
- [FHP 00] FIEDLER, ULRICH, POLLY HUANG und BERNHARD PLATTNER: *Over-provisioning or Differentiated Services – A Case Study on integrating services over IP*, citeseer.nj.nec.com/384952.html.
- [FLNN+ 00] FERRARI, TIZIANA, SIMON LEINEN, JAN NOVAK, SIMON NYBROE, HERVE PRIGENT, VICTOR REIJS, ROBERTO SABATINO und ROBERT STOY: *Report on Results of the Quantum Test Programme*. Projektbericht, Projekt QUANTUM, Juni 2000, <http://www.dante.net/quantum/qtp/final-report.pdf>.
- [GaRo 04] GARSCHHAMMER, M. und H. ROELLE: *Requirements on Quality Specification Posed by Service Orientation*. In: *Proceedings of the 15th IFIP/IEEE International Workshop on Distributed Systems: Operations*

- and Management (DSOM 2004)*, Seiten 1–14, Davis, CA, USA, November 2004. IFIP/IEEE, Springer.
- [Gars 04] GARSCHHAMMER, M.: *Entwicklung und Anwendung eines universellen Konzepts zur rechnergestützten Spezifikation und Messung von Dienstgütemerkmalen*. Dissertation, Ludwig-Maximilians-Universität München, August 2004.
- [GBTB+ 02] GODERIS, D., S. VAN DEN BOSCH, Y. T'JOENS, G. PAVLOU, P. TRIMINTZIOS, D. GRIFFIN, P. GEORGATSOS, G. MEMENIOS, E. MYKONIATI und C. JACQUENET: *A Service-Centric Quality of Service Architecture for IP-based Next Generation Networks*. In: STADLER, R. und ULEMA M. (Herausgeber): *Proceedings of the 8th International IFIP/IEEE Network Operations and Management Symposium (NOMS 2002)*, Florence, Italy, April 2002. IFIP/IEEE, IEEE Publishing, <http://www.ee.ucl.ac.uk/~dgriffin/papers/Goder-02a.pdf>.
- [GHHK+ 01] GARSCHHAMMER, M., R. HAUCK, H.-G. HEGERING, B. KEMPTER, M. LANGER, M. NERB, I. RADISIC, H. ROELLE und H. SCHMIDT: *Towards generic Service Management Concepts – A Service Model Based Approach*. In: PAVLOU, G., N. ANEROUSIS und A. LIOTTA (Herausgeber): *Proceedings of the 7th International IFIP/IEEE Symposium on Integrated Management (IM 2001)*, Seiten 719–732, Seattle, Washington, USA, Mai 2001. IFIP/IEEE, IEEE Publishing, <http://wwwnmteam.informatik.uni-muenchen.de/php-bin/pub/show{ }pub.php?key=smtf01>.
- [GHHK 02] GARSCHHAMMER, M., R. HAUCK, H.-G. HEGERING, B. KEMPTER, I. RADISIC, H. ROELLE und H. SCHMIDT: *A Case-Driven Methodology for Applying the MNM Service Model*. In: STADLER, R. und M. ULEMA (Herausgeber): *Proceedings of the 8th International IFIP/IEEE Network Operations and Management Symposium (NOMS 2002)*, Seiten 697–710, Florence, Italy, April 2002. IFIP/IEEE, IEEE Publishing, <http://wwwnmteam.informatik.uni-muenchen.de/php-bin/pub/show{ }pub.php?key=ghhk02>.
- [GHJV 95] GAMMA, ERICH, RICHARD HELM, RALPH JOHNSON und JOHN VLISIDES: *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison-Wesley, 1995, <http://www.amazon.com/exec/obidos/ISBN=0201633612>. ISBN 0201633612.
- [GHKR+ 01] GARSCHHAMMER, M., R. HAUCK, B. KEMPTER, I. RADISIC, H. ROELLE und H. SCHMIDT: *The MNM Service Model – Refined Views on Generic Service Management*. *Journal of Communications and Networks*, 3(4):297–306, Dezember 2001, <http://wwwnmteam.informatik.uni-muenchen.de/php-bin/pub/show{ }pub.php?key=ghkr01>.
- [GHSB 03] GARSCHHAMMER, M., H.-G. HEGERING, M. SCHIFFERS, M. BROY und

LITERATUR

- A. PICOT: *Kommunikations- und Informationstechnik 2010+3*. Bundesamt für Sicherheit in der Informationstechnik, Bonn, Deutschland, Mai 2003. ISBN 3-922746-48-9.
- [GrSe 05] GRAMPIN, E. und J. SERRAT: *Cooperation of control and management plane for provisioning in MPLS networks*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE.
- [Guti 04] GUTIERREZ, PEDRO A. ARANDA: *BGP-4 Protocol Patterns and Their Impact on QoS Behavior*. Inter-domain Performance and Simulation Workshop, Projekt IST-INTERMON: Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation, Budapest, März 2004, http://www.ist-intermon.org/overview/ips_2004/ips2004_031.pdf.
- [HAN 99] HEGERING, H.-G., S. ABECK und B. NEUMAIR: *Integrated Management of Networked Systems – Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, 1999. ISBN 1-55860-571-1.
- [HCMO+ 04] HWANG, JUNSEOK, STEVE J. CHAPIN, HACI A. MANTAR, IBRAHIM T. OKUMUS, RAJEESH REVURU und ASEEM SALAMA: *An Implementation Study of a Dynamic Inter-Domain Bandwidth Management Platform in DiffServ Networks*. In: *Managing Next Generation Convergence Networks and Services: Proceedings of the 9th IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, Seoul, Korea, April 2004. IFIP/IEEE, http://www.hpdc.syr.edu/~chapin/papers/pdf/BMP_impl.pdf.
- [HeRa 05] HENIG, ASAF und DANNY RAZ: *Efficient management of transcoding and multicasting multimedia streams*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE, <http://www.cs.technion.ac.il/~cshenig/images/TPMapproximation.pdf>.
- [IntServ-WG] *Integrated Services (intserv)*, <http://www.ietf.org/html.charters/intserv-charter.html>. IETF Working Group.
- [ITU-T I.150] ITU-T: *I.150 – Integrated Services Digital Network (ISDN) – General Structure – B-ISDN Asynchronous Transfer Mode Functional Characteristics*. ITU-T Recommendation, International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), Dezember 1995.
- [ITU-T I.431] ITU-T: *I.431 – Primary Rate User – Network Interface – Layer 1 Specification*. ITU-T Recommendation, International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), März 1993.

- [ITU-T X.200] ITU-T: *X.200 – Data Networks and Open System Communications – Open Systems Interconnection-Model and Notation – Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. ITU-T Recommendation, International Telecommunication Union – Telecommunication Standardization Sector (ITU-T), Juli 1994.
- [ITUT-HP] *International Telecommunication Union – Telecom Standardization (ITU-T)*, <http://www.itu.int/ITU-T/> . Homepage.
- [Kell 98] KELLER, A.: *CORBA-basiertes Enterprise Management: Interoperabilität und Managementinstrumentierung verteilter kooperativer Managementsysteme in heterogener Umgebung*. Dissertation, Technische Universität München, Dezember 1998, http://wwwmmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=kell98 .
- [KZS 00] KRASNODEMBSKI, KAI, MARTINA ZITTERBART und JOACHIM SOKOL: *End-to-End QoS Provisioning Across Heterogeneous Domains*. In: *IEEE Conference on High Performance Switching and Routing (ATM 2000)*, Heidelberg, Deutschland, Juni 2000. , http://www.ibr.cs.tu-bs.de/users/krasno/atm2000_krasno.ps .
- [LaNe 00] LANGER, M. und M. NERB: *Customer Service Management: An Information Model for Communication Services*. In: LINNHOFF-POPIEN, C. und H.-G. HEGERING (Herausgeber): *Trends in Distributed Systems: Towards a Universal Service Market. Proceedings of the third International IFIP/GI Working Conference, USM 2000*, Nummer 1890 in *Lecture Notes in Computer Science (LNCS)*, Munich, Germany, September 2000. Ludwig-Maximilians-Universität München, Springer.
- [LaOw 05] LARRIEU, N. und P. OWEZARSKI: *Measurement based networking approach applied to congestion control in the multi-domain internet*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE, http://dbserver.laas.fr/pls/LAAS/publis.rech_doc?langage=ENG&clef=101206 .
- [MPLS-WG] *Multiprotocol Label Switching (mpls)*, <http://www.ietf.org/html.charters/mpls-charter.html> . IETF Working Group.
- [Nerb 01] NERB, M.: *Customer Service Management als Basis für interorganisationales Dienstmanagement*. Dissertation, Technische Universität München, März 2001, <http://tumb1.biblio.tu-muenchen.de/publ/diss/in/2001/nerb.html> .
- [OMG 03-03-01] OMG: *Unified Modeling Language Specification, v1.5*. OMG Specification formal/03-03-01, Object Management Group, März 2003, <http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf> .

LITERATUR

- [PHS 00] PAN, P., E. HAHNE und H. SCHULZRINNE: *BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations*. Journal of Communications and Networks, 2(2):157–167, Juni 2002, <http://www1.cs.columbia.edu/~pingpan/papers/bgrp.pdf>.
- [QGMT+ 05] QUICKENBORNE, FREDERIC VAN, FILIP DE GREVE, INGRID MOERMAN, FILIP DE TURCK und PIET DEMEESTER: *On the management of aggregation networks with rapidly moving traffic demands*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE.
- [QUASAR-HP] *Projekt QUASAR – Quality of Service Architectures*, <http://www.ind.uni-stuttgart.de/Content/Quasar/index.html>. Homepage.
- [Radi 03] RADISIC, I.: *Ein prozessorientierter, policy-basierter Ansatz für ein integriertes, dienstorientiertes Abrechnungsmanagement*. Dissertation, Ludwig-Maximilians-Universität München, Februar 2003, http://wwwmmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=radi03.
- [RFC 1771] REKHTER, Y. und T. LI: *RFC 1771: A Border Gateway Protocol 4 (BGP-4)*. RFC, IETF, März 1995, <http://www.ietf.org/rfc/rfc1771.txt>.
- [RFC 2026] BRADNER, S.: *RFC 2026: The Internet Standards Process – Revision 3*. RFC, IETF, Oktober 1996, <http://www.ietf.org/rfc/rfc2026.txt>.
- [RFC 2205] BRADEN, R., ED., L. ZHANG, S. BERSON, S. HERZOG und S. JAMIN: *RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2205.txt>.
- [RFC 2208] MANKIN, A., ED., F. BAKER, B. BRADEN, S. BRADNER, M. O'DELL, A. ROMANOW, A. WEINRIB und L. ZHANG: *RFC 2208: Resource ReSerVation Protocol (RSVP) – Version 1 Applicability Statement Some Guidelines on Deployment*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2208.txt>.
- [RFC 2209] BRADEN, R. und L. ZHANG: *RFC 2209: Resource ReSerVation Protocol (RSVP) – Version 1 Message Processing Rules*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2209.txt>.
- [RFC 2210] WROCLAWSKI, J.: *RFC 2210: The Use of RSVP with IETF Integrated Services*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2210.txt>.
- [RFC 2211] WROCLAWSKI, J.: *RFC 2211: Specification of the Controlled-Load Network Element Service*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2211.txt>.

- [RFC 2212] SHENKER, S., C. PARTRIDGE und R. GUERIN: *RFC 2212: Specification of Guaranteed Quality of Service*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2212.txt>.
- [RFC 2215] SHENKER, S. und J. WROCLAWSKI: *RFC 2215: General Characterization Parameters for Integrated Service Network Elements*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2215.txt>.
- [RFC 2216] SHENKER, S. und J. WROCLAWSKI: *RFC 2216: Network Element Service Specification Template*. RFC, IETF, September 1997, <http://www.ietf.org/rfc/rfc2216.txt>.
- [RFC 2328] MOY, J.: *RFC 2328: OSPF Version 2*. RFC, IETF, April 1998, <http://www.ietf.org/rfc/rfc2328.txt>.
- [RFC 2381] GARRETT, M. und M. BORDEN: *RFC 2381: Interoperation of Controlled-Load Service and Guaranteed Service with ATM*. RFC, IETF, August 1998, <http://www.ietf.org/rfc/rfc2381.txt>.
- [RFC 2382] CRAWLEY, E., ED., L. BERGER, S. BERSON, F. BAKER, M. BORDEN und J. KRAWCZYK: *RFC 2382: A Framework for Integrated Services and RSVP over ATM*. RFC, IETF, August 1998, <http://www.ietf.org/rfc/rfc2382.txt>.
- [RFC 2453] MALKIN, G.: *RFC 2453: RIP Version 2*. RFC, IETF, November 1998, <http://www.ietf.org/rfc/rfc2453.txt>.
- [RFC 2474] NICHOLS, K., S. BLAKE, F. BAKER und D. BLACK: *RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC, IETF, Dezember 1998, <http://www.ietf.org/rfc/rfc2474.txt>.
- [RFC 2475] BLAKE, S., D. BLACK, M. CARLSON, E. DAVIES, Z. WANG und W. WEISS: *RFC 2475: An Architecture for Differentiated Service*. RFC, IETF, Dezember 1998, <http://www.ietf.org/rfc/rfc2475.txt>.
- [RFC 2597] HEINANEN, J., F. BAKER, W. WEISS und J. WROCLAWSKI: *RFC 2597: Assured Forwarding PHB Group*. RFC, IETF, Juni 1999, <http://www.ietf.org/rfc/rfc2597.txt>.
- [RFC 2998] BERNET, Y., P. FORD, R. YAVATKAR, F. BAKER, L. ZHANG, M. SPEER, R. BRADEN, B. DAVIE, J. WROCLAWSKI und E. FELSTAIN: *RFC 2998: A Framework for Integrated Services Operation over Diffserv Networks*. RFC, IETF, November 2000, <http://www.ietf.org/rfc/rfc2998.txt>.
- [RFC 3031] ROSEN, E., A. VISWANATHAN und R. CALLON: *RFC 3031: Multiprotocol Label Switching Architecture*. RFC, IETF, Januar 2001, <http://www.ietf.org/rfc/rfc3031.txt>.
- [RFC 3034] CONTA, A., P. DOOLAN und A. MALIS: *RFC 3034: Use of Label Switching on Frame Relay Networks Specification*. RFC, IETF, Januar 2001, <http://www.ietf.org/rfc/rfc3034.txt>.

LITERATUR

- [RFC 3035] DAVIE, B., J. LAWRENCE, K. MCCLOGHRIE, E. ROSEN, G. SWALLOW, Y. REKHTER und P. DOOLAN: *RFC 3035: MPLS using LDP and ATM VC Switching*. RFC, IETF, Januar 2001, <http://www.ietf.org/rfc/rfc3035.txt> .
- [RFC 3086] NICHOLS, K. und B. CARPENTER: *RFC 3086: Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification*. RFC, IETF, April 2001, <http://www.ietf.org/rfc/rfc3086.txt> .
- [RFC 3140] BLACK, D., S. BRIM, B. CARPENTER und F. LE FAUCHEUR: *RFC 3140: Per Hop Behavior Identification Codes*. RFC, IETF, Juni 2001, <http://www.ietf.org/rfc/rfc3140.txt> .
- [RFC 3168] RAMAKRISHNAN, K., S. FLOYD und D. BLACK: *RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP*. RFC, IETF, September 2001, <http://www.ietf.org/rfc/rfc3168.txt> .
- [RFC 3209] AWDUCHE, D., L. BERGER, D. GAN, T. LI, V. SRINIVASAN und G. SWALLOW: *RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels*. RFC, IETF, Dezember 2001, <http://www.ietf.org/rfc/rfc3209.txt> .
- [RFC 3246] DAVIE, B., A. CHARNY, J.C.R. BENNET, K. BENSON, J.Y. LE BOUDEC, W. COURTNEY, S. DAVARI, V. FIROIU und D. STILIADIS: *RFC 3246: An Expedited Forwarding PHB (Per-Hop Behavior)*. RFC, IETF, März 2002, <http://www.ietf.org/rfc/rfc3246.txt> .
- [RFC 3247] CHARNY, A., J. BENNET, K. BENSON, J. BOUDEC, A. CHIU, W. COURTNEY, S. DAVARI, V. FIROIU, C. KALMANEK und K. RAMAKRISHNAN: *RFC 3247: Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior)*. RFC, IETF, März 2002, <http://www.ietf.org/rfc/rfc3247.txt> .
- [RFC 3248] ARMITAGE, G., B. CARPENTER, A. CASATI, J. CROWCROFT, J. HALPERN, B. KUMAR und J. SCHNIZLEIN: *RFC 3248: A Delay Bound alternative revision of RFC 2598*. RFC, IETF, März 2002, <http://www.ietf.org/rfc/rfc3248.txt> .
- [RFC 3260] GROSSMAN, D.: *RFC 3260: New Terminology and Clarifications for Diff-serv*. RFC, IETF, April 2002, <http://www.ietf.org/rfc/rfc3260.txt> .
- [RFC 3270] FAUCHEUR, F. LE, L. WU, B. DAVIE, S. DAVARI, P. VAANANEN, R. KRISHNAN, P. CHEVAL und J. HEINANEN: *RFC 3270: Multi-Protocol Label Switching (MPLS) Support of Differentiated Services*. RFC, IETF, Mai 2002, <http://www.ietf.org/rfc/rfc3270.txt> .
- [RFC 768] POSTEL, J.: *RFC 768: User Datagram Protocol*. RFC, IETF, August 1980, <http://www.ietf.org/rfc/rfc768.txt> .
- [RFC 791] POSTEL, J.: *RFC 791: Internet Protocol*. RFC, IETF, September 1981, <http://www.ietf.org/rfc/rfc791.txt> .

- [RFC 793] POSTEL, J.: *RFC 793: Transmission Control Protocol*. RFC, IETF, September 1981, <http://www.ietf.org/rfc/rfc793.txt>.
- [Roel 01] ROELLE, H.: *Introduction to MPLS and its Application in a Customer/Provider Scenario*. In: *8th International Workshop of the HP OpenView University Association (HPOVUA 2001)*, Berlin, Germany, Juni 2001.
- [Sail 02] SAILER, M.: *Klassifizierung und Bewertung von VPN-Lösungen für die Neuausrichtung der europaweiten Extranetstrategie der BMW AG*. Diplomarbeit, Technische Universität München, August 2002, http://wwwmmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=sail02.
- [Schm 02a] SCHMITZ, D.: *Konzeption und prototypische Implementierung eines Werkzeugs zur Unterstützung der MNM Dienstmodellierungsmethodik*. Diplomarbeit, Ludwig-Maximilians-Universität München, September 2002, http://wwwmmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=schm02a.
- [SchmH 01] SCHMIDT, H.: *Entwurf von Service Level Agreements auf der Basis von Dienstprozessen*. Dissertation, Ludwig-Maximilians-Universität München, Juli 2001.
- [Schm]B 01] SCHMITT, JENS BURKHARD: *Heterogeneous Network Quality of Service Systems*. Kluwer Academic Publishers, 2001. ISBN 0-7923-7410-X.
- [SGGG+ 01] SERRAT, J., E. GRAMPIN R. GARCIA, A. GALIS, P. BAYVEL, M. DASER, C. TODD, H. BALEMANS, W. VERHOEF, J. SANCHEZ, L. RAPTIS, BRUNO MAYERSOHN, N. KATSANOU, F. KARAYANNIS, J.A. LOZANO, D. CHRONIS, D. JOSEF und N. SILVA: *Integrated Management for IP end-to-end Transport Services over WDM Networks*. In: PAVLOU, G., N. ANEROUSIS und A. LIOTTA (Herausgeber): *Proceedings of the 7th International IFIP/IEEE Symposium on Integrated Management (IM 2001)*, Seattle, Washington, USA, Mai 2001. IFIP/IEEE, IEEE Publishing.
- [StarTap-HP] *StarTap*, <http://www.startap.net/startap/>. Homepage.
- [Tane 03] TANENBAUM, ANDREW S.: *Computer Networks*. Prentice Hall, fourth edition Auflage, 2003. ISBN 0-13-066102-3.
- [TAPC+ 01] P. TRIMINTZIOS AND I. ANDRIKOPOULOS AND G. PAVLOU AND C.F. CAVALCANTI AND D. GODERIS AND Y. T'JOENS AND P. GEORGATSOS AND L. GEORGIADIS AND D. GRIFFIN AND R. EGAN AND C. JACQUENET AND G. MEMENIOS: *An Architectural Framework for Providing QoS in IP Differentiated Services Networks*. In: PAVLOU, G., N. ANEROUSIS und A. LIOTTA (Herausgeber): *Proceedings of the 7th International IFIP/IEEE Symposium on Integrated Management (IM 2001)*, Seattle, Washington, USA, Mai 2001. IFIP/IEEE, IEEE Publishing, <http://www.ee.surrey.ac.uk/Personal/G.Pavlou/Publications/Conference-papers/Trimin-01b.pdf>.

LITERATUR

- [TEQUILA-HP] *Traffic Engineering for Quality of Service in the Internet, at Large Scale*, <http://www.ist-tequila.org/> . Homepage.
- [Thae 01] THAENS, K.: *Entwicklung eines Kriterienkatalogs zur Bestimmung der Einsatzreife von Netzinfrastrukturen zum Einsatz von VoIP*. Diplomarbeit, Ludwig-Maximilians-Universität München, Januar 2001, http://wwwmmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=thae01 .
- [TS 122 246] ETSI: *Digital cellular telecommunications system (Phase 2+) – Universal Mobile Telecommunications System (UMTS) – Multimedia Broadcast/Multicast Service (MBMS) user services – Stage 1 (3GPP TS 22.246 version 6.2.0 Release 6)*. ETSI Technical Specification ETSI TS 122 246 V6.2.0, European Telecommunications Standards Institute (ETSI), September 2004, <http://pda.etsi.org/pda/home.asp?wkr=DTS/TSGS-0122246v620> .
- [TS 123 107] ETSI: *Universal Mobile Telecommunications System (UMTS) – Quality of Service (QoS) concept and architecture (3GPP TS 23.107 version 6.2.0 Release 6)*. ETSI Technical Specification ETSI TS 123 107 V6.2.0, European Telecommunications Standards Institute (ETSI), Dezember 2004, <http://pda.etsi.org/pda/home.asp?wkr=RTS/TSGS-0223107v620> .
- [TS 123 207] ETSI: *Digital cellular telecommunications system (Phase 2+) – Universal Mobile Telecommunications System (UMTS) – End-to-end Quality of Service (QoS) concept and architecture (3GPP TS 23.207 version 6.4.0 Release 6)*. ETSI Technical Specification ETSI TS 123 207 V6.4.0, European Telecommunications Standards Institute (ETSI), September 2004, <http://pda.etsi.org/pda/home.asp?wkr=RTS/TSGS-0223207v640> .
- [TS 123 246] ETSI: *Universal Mobile Telecommunications System (UMTS) – Multimedia Broadcast/Multicast Service (MBMS) – Architecture and functional description (3GPP TS 23.246 version 6.6.0 Release 6)*. ETSI Technical Specification ETSI TS 123 246 V6.6.0, European Telecommunications Standards Institute (ETSI), März 2005, <http://pda.etsi.org/pda/home.asp?wkr=RTS/TSGS-0223246v660> .
- [VPMK 04] VALI, DIMITRA, SARANTIS PASKALIS, LAZAROS MERAKOS und ALEXANDROS KALOXYLOS: *A Survey of Internet QoS Signaling*. IEEE Communications Surveys and Tutorials, 6(4):32–43, 2004, <http://www.comsoc.org/livepubs/surveys/public/2004/oct/pdf/VALI.pdf> .
- [VVWM+ 02] VERMEULEN, BRECHT, STEFAAN VANHASTEL, JEROEN WELLEN, CARMEN MAS, FREDERIK SCHOLAERT und PIET DEMEESTER: *A Generic End-to-end Distributed QoS Management Architecture and its application to IP-DiffServ over a WDM Access Feeder Network*. In: STADLER, R. und ULEMA M. (Herausgeber): *Proceedings of the 8th International IFIP/IEEE*

- Network Operations and Management Symposium (NOMS 2002)*, Florence, Italy, April 2002. IFIP/IEEE, IEEE Publishing.
- [XiBo 05] XIAO, J. und R. BOUTABA: *QoS aware Service composition in large scale multi-domain networks*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005) – Managing New Networked Worlds*, Nice, France, Mai 2005. IFIP/IEEE.
- [XiNi 99] XIAO, X. und L. M. NI: *Internet QoS: A Big Picture*. IEEE Network, 13:8–18, März 1999, <http://www.cs.virginia.edu/~cs757/papers/xiao99internet.pdf>.
- [XLWN 02] XIAO, L., K. LUI, J. WANG und K. NAHRSTEDT: *QoS extension to BGP*. In: *10th IEEE International Conference on Network Protocols*, Paris, November 2002., <http://citeseer.ist.psu.edu/694925.html>.
- [XWLN 04] XIAO, L., J. WANG K. LIU und K. NAHRSTEDT: *Advertising Interdomain QoS Routing Information*. IEEE Journal on Selected Areas in Communications, 22(10), Dezember 2004, http://www.comsoc.org/livepubs/sac/public/2004/dec/1949_22jsac10-xiao.html.

LITERATUR

TEILFALLINDEX

KM-B...	
KM-bereich-einzelfall.....	145, 145
KM-C...	
KM-Csm-*	200
KM-D...	
KM-Dkl-*	192
KM-Dkl-Tr1	190, 190 , 193
KM-Dkl-Tr2	190 , 191, 193
KM-Dkl-Tr3	190 , 191, 193
KM-Dkl-Tr4	190 , 193
KM-Dkl-Tu1	190 , 193
KM-Dkl-Tu2	191 , 193
KM-Dkl-Tu3	191 , 192, 193
KM-Dkl-Tu4	192 , 193
KM-E...	
KM-Evt-class	197
KM-G...	
KM-gZ-*	155
KM-gZ-Tr1	149 , 151
KM-gZ-Tr2	149 , 151
KM-gZ-Tr3	150 , 151, 156
KM-gZ-Tr4	150
KM-gZ-Tu1	151
KM-gZ-Tu2	151
KM-gZ-Tu3	151 , 152
KM-gZ-Tu4	152
KM-K...	
KM-Kr-MMTr1	162
KM-Kr-MMTr2	162
KM-Kr-MMTr3	161 , 162
KM-Kr-MMTr4	162
KM-Kr-MMTu1	163
KM-Kr-MMTu1/2/4	161 , 162
KM-Kr-MMTu2	163
KM-Kr-MMTu3	162 , 163
KM-Kr-MMTu4	163
KM-Kr-MUTr1	166
KM-Kr-MUTr1/2/4	164, 164
KM-Kr-MUTr2	166
KM-Kr-MUTr3	164 , 166
KM-Kr-MUTr4	166
KM-Kr-MUTu1	166
KM-Kr-MUTu1/2/4	163, 165
KM-Kr-MUTu2	166
KM-Kr-MUTu3	163, 165 , 166
KM-Kr-MUTu4	166
KM-Kr-UMTr1	165
KM-Kr-UMTr1/2/4	163, 163
KM-Kr-UMTr2	165
KM-Kr-UMTr3	163 , 166
KM-Kr-UMTr4	166
KM-Kr-UMTu1	166
KM-Kr-UMTu1/2/4	163
KM-Kr-UMTu2	166
KM-Kr-UMTu3	164 , 166
KM-Kr-UMTu4	166
KM-Kr-UUTr1	162
KM-Kr-UUTr1/2/4	164
KM-Kr-UUTr2	162
KM-Kr-UUTr3	160 , 162, 164
KM-Kr-UUTr4	162
KM-Kr-UUTu1	162
KM-Kr-UUTu2	162
KM-Kr-UUTu3	161 , 162
KM-Kr-UUTu4	162
KM-P...	
KM-Pe-*	183
KM-Pe-BdTr1	171
KM-Pe-BdTr2	169 , 171
KM-Pe-BdTr3	171
KM-Pe-BdTu1	171
KM-Pe-BdTu2	169 , 171
KM-Pe-BdTu3	171
KM-Pe-UdTr1	170
KM-Pe-UdTr2	170
KM-Pe-UdTr3	167 , 170
KM-Pe-UdTr4	170
KM-Pe-UdTu1	170
KM-Pe-UdTu2	170

TEILFALLINDEX

KM-Pe-UdTu3.....	168 , 169, 170
KM-Pe-UdTu4.....	170

KM-Q...

KM-Qkl-*.....	192
KM-Qkl-Tr1.....	193
KM-Qkl-Tr2.....	193
KM-Qkl-Tr3.....	193
KM-Qkl-Tr4.....	193, 216, 216
KM-Qkl-Tu1.....	193
KM-Qkl-Tu2.....	193
KM-Qkl-Tu3.....	193
KM-Qkl-Tu4.....	194
KM-QpG-Tr1.....	177, 180
KM-QpG-Tr2.....	177
KM-QpG-Tr3.....	177, 177 , 180
KM-QpG-Tu1.....	177, 180
KM-QpG-Tu2.....	177
KM-QpG-Tu3.....	177, 177 , 180
KM-QpS-Vgl.....	175 , 176
KM-QpUp-ATr1.....	186 , 187
KM-QpUp-ATr1/2/Tu1/2.....	186
KM-QpUp-ATr2.....	186 , 187
KM-QpUp-ATu1.....	186 , 187
KM-QpUp-ATu2.....	186 , 187
KM-QpUp-STr1.....	187
KM-QpUp-STr2.....	187
KM-QpUp-STu1.....	187
KM-QpUp-STu2.....	187
KM-QpW-Semmod.....	179 , 180
KM-QpW-Tr1.....	183 , 185
KM-QpW-Tr2.....	183 , 185
KM-QpW-Tr3.....	183 , 185
KM-QpW-Tr4.....	183 , 185
KM-QpW-Trans.....	182, 182
KM-QpW-Tu1.....	184 , 185
KM-QpW-Tu2.....	184 , 185
KM-QpW-Tu3.....	184 , 185
KM-QpW-Tu4.....	184 , 185
KM-QpW-TuSig.....	184
KM-QpW-Typmod.....	180, 180

KM-U...

KM-Usap-*.....	199
----------------	------------

KM-V...

KM-Vo-*.....	196
KM-Vo-Tr1.....	155 , 156, 158
KM-Vo-Tr2.....	155 , 156, 158
KM-Vo-Tr3.....	156, 156 , 157, 158
KM-Vo-Tr4.....	156 , 158
KM-Vo-Tu1.....	156 , 158
KM-Vo-Tu2.....	156 , 158
KM-Vo-Tu3.....	157 , 158
KM-Vo-Tu4.....	158 , 159, 184

TEILLÖSUNGSINDEX

...-(KM-B...)		...-(KM-P...)	
L1-(KM-bereich-einzelfall)	145, 145	L1-(KM-Pe-BdTr2)	169, 171
...-(KM-C...)		L2-(KM-Pe-BdTr2)	169, 171
L1-(KM-Csm-*)	200	L1-(KM-Pe-BdTu2)	169, 171
...-(KM-D...)		L2-(KM-Pe-BdTu2)	170, 171
L1-(KM-Dkl-Tr2)	190, 191, 193	L3-(KM-Pe-BdTu2)	170, 171
L2-(KM-Dkl-Tr2)	190, 191, 193	L4-(KM-Pe-BdTu2)	170, 171
L1-(KM-Dkl-Tr4)	190, 192, 193	L1-(KM-Pe-UdTr3)	168, 170
L1-(KM-Dkl-Tu2)	191, 193	L2-(KM-Pe-UdTr3)	168, 170
L2-(KM-Dkl-Tu2)	191, 193	L1-(KM-Pe-UdTu3)	168, 170
L1-(KM-Dkl-Tu3)	191, 192, 193	L2-(KM-Pe-UdTu3)	168, 170
L2-(KM-Dkl-Tu3)	191, 192, 193	L3-(KM-Pe-UdTu3)	169, 170
L1-(KM-Dkl-Tu4)	192, 193	L4-(KM-Pe-UdTu3)	169, 170
L2-(KM-Dkl-Tu4)	192, 193	...-(KM-Q...)	
L3-(KM-Dkl-Tu4)	192, 193	L1-(KM-Qkl-Tr2)	193
...-(KM-G...)		L2-(KM-Qkl-Tr2)	193
L1-(KM-gZ-Tr2)	150, 151	L1-(KM-Qkl-Tr4)	193, 216
L2-(KM-gZ-Tr2)	150, 151	L1-1-(KM-Qkl-Tr4)	216
L1-(KM-gZ-Tr4)	150, 152	L1-2-(KM-Qkl-Tr4)	216
L1-(KM-gZ-Tu2)	151	L1-3-(KM-Qkl-Tr4)	217
L2-(KM-gZ-Tu2)	151	L1-(KM-Qkl-Tu2)	193
L1-(KM-gZ-Tu3)	151, 152	L2-(KM-Qkl-Tu2)	193
L2-(KM-gZ-Tu3)	152, 152	L1-(KM-Qkl-Tu3)	193
L1-(KM-gZ-Tu4)	152, 196	L2-(KM-Qkl-Tu3)	193
L2-(KM-gZ-Tu4)	152, 196	L1-(KM-Qkl-Tu4)	194
L3-(KM-gZ-Tu4)	152, 196	L2-(KM-Qkl-Tu4)	194
...-(KM-K...)		L3-(KM-Qkl-Tu4)	194
L1-(KM-Kr-MMTr3)	161, 162	L1-(KM-QpG-Tr3)	177, 177
L1-(KM-Kr-MMTu1/2/4)	161, 163	L1-(KM-QpG-Tu3)	177, 177
L1-(KM-Kr-MMTu3)	162, 163	L1-(KM-QpS-Vgl)	175, 176, 178
L1-(KM-Kr-MUTr1/2/4)	164, 166	L2-(KM-QpS-Vgl)	175, 176, 178–180
L1-(KM-Kr-MUTu1/2/4)	161, 165, 165, 166	L3-(KM-QpS-Vgl)	175, 175, 176
L1-(KM-Kr-MUTu3)	162, 165, 166	L1-(KM-QpUp-ATr1)	186, 187
L1-(KM-Kr-UMTr1/2/4)	163, 163, 164–166	L1-(KM-QpUp-ATr1/2/Tu1/2)	187, 187
L1-(KM-Kr-UMTr3)	163, 166	L1-(KM-QpUp-ATr2)	186, 186, 187
L1-(KM-Kr-UMTu1/2/4)	164, 166	L1-(KM-QpUp-ATu1)	186, 187
L1-(KM-Kr-UMTu3)	164, 166	L1-(KM-QpUp-ATu2)	186, 187
L1-(KM-Kr-UUTr3)	160, 162, 163	L1-(KM-QpW-Semmod)	180, 180
L1-(KM-Kr-UUTu3)	161, 162, 164, 165	L1-(KM-QpW-Tr3)	183, 184, 185
		L1-(KM-QpW-Tr4)	183, 185
		L1-(KM-QpW-Trans)	182, 182

TEILLÖSUNGSINDEX

L2-(KM-QpW-Trans)	182, 182
L1-(KM-QpW-Tu1)	184 , 185
L1-(KM-QpW-Tu2)	184 , 185
L1-(KM-QpW-Tu3)	184 , 185
L1-(KM-QpW-Tu4)	184 , 185
L1-(KM-QpW-TuSig)	184, 184
L2-(KM-QpW-TuSig)	184, 184
L3-(KM-QpW-TuSig)	184, 184
L1-(KM-QpW-Typmod)	180, 180

...-(KM-U...)

L1-(KM-Usap-*)	199
----------------------	------------

...-(KM-V...)

L1-(KM-Vo-Tr2)	155 , 156, 158, 190
L2-(KM-Vo-Tr2)	155 , 156, 158, 190
L1-(KM-Vo-Tr4)	156 , 158, 190, 216, 217
L1-(KM-Vo-Tu2)	156 , 158
L2-(KM-Vo-Tu2)	156 , 158
L1-(KM-Vo-Tu3)	157 , 158
L2-(KM-Vo-Tu3)	157 , 158
L1-(KM-Vo-Tu4)	158 , 159, 184
L2-(KM-Vo-Tu4)	158 , 159, 184
L3-(KM-Vo-Tu4)	158 , 159, 184

SCHLAGWORTINDEX

A

ABR..... *siehe* Available Bit Rate
Absolute Garantie **93**
access provider **16**
Änderungsphase **108**
AF..... *siehe* Assured Forwarding
Anforderungen **34, 40, 42, 60, 131, 206**
Anwendungsfall **30, 141, 218**
Arbeitspakete **202**
Architekturstandards **46**
AS..... *siehe* Autonomous System
Assured Forwarding **49, 88, 98, 102, 114, 118, 125, 128, 180**
Asymmetrische Koppelung **142**
Asynchronous Transfer Mode ... **9, 9, 46, 97, 99, 102, 107, 108, 117, 120, 121, 126, 180, 196**
ATM *siehe* Asynchronous Transfer Mode, **126**
ATM-Forum **9, 46**
Autonomous System..... **78**
Available Bit Rate..... **100, 121, 196**

B

Basic Model..... **25**
Basis-Modell..... **25, 76**
Basis-Rahmenwerken..... **24, 61**
Basistechnologien **24, 61**
Beendigungskorridor **119**
Beendigungsphase **109**
Belegungskorridor **119**
Bereitstellungsphase **108, 205**
Best-Effort..... **93**
Betriebsphase..... **108, 205**
BGP..... *siehe* Border Gateway Protocol
Bidirektionale Kommunikationsrichtung **143, 160**
Black-Box-Sicht **78, 131, 132**
Border Gateway Protocol..... **22, 81**

C

CBR..... *siehe* Constant Bit Rate
Cell Loss Rate **97**
Class-of-Service..... **21, 49, 93, 112, 197**
CLR..... *siehe* Cell Loss Rate

Constant Bit Rate..... **99**
Controlled-Load Service **48**
core provider **17**
CoS..... *siehe* Class-of-Service
CSM..... *siehe* Customer Service Management
customer **25**
Customer Service Management . **25, 74, 76, 130, 200**
customer service management (CSM) client .. **75**
customer side **25**

D

Daten-Kategorie **85, 153**
Datenprotokolle **20**
Definitionskorridor..... **119, 126, 140, 201, 212**
Deterministische Garantie **93**
Deutsches Forschungsnetz..... **58**
DFN..... *siehe* Deutsches Forschungsnetz
Dienst..... **25, 72, 126**
Dienstimplementierung **75, 78**
Dienstleister **25**
Dienstleisterseite..... **25**
Dienstmanagement-Implementierung **75**
Dienstnehmerseite **25**
Dienstsicht **76**
Diensttypen **202**
Dienstvereinbarung **74**
Dienstzugangspunkt..... **78**
Differentiated Services. **9, 10, 49, 83, 89, 95, 98, 102, 103, 111, 114, 118, 120**
DiffServ *siehe* Differentiated Services, **102**
DiffServ Codepoint..... **102**
dropping..... **88**
DS *siehe* DiffServ
DSCP *siehe* DiffServ Codepoint

E

EBNF *siehe* Extended Backus-Naur Form
ECN *siehe* Explicit Congestion Notification
EF..... *siehe* Expedited Forwarding
Entity Relationship..... **52**
ER..... *siehe* Entity Relationship
Ereignisse..... **120, 125, 195**

SCHLAGWORTINDEX

Expedited Forwarding 49, 58
Explicit Congestion Notification 89
Extended Backus-Naur Form 144

F

FEC *siehe* Forwarding Equivalency Class
Flow 110
Forwarding Equivalency Class 62
FR *siehe* Frame Relay
Frame Relay 63
Fundamentalrollen 25

G

G-WiN *siehe* Gigabit-Wissenschaftsnetz
Garantietyp 92, 176
Geschwindigkeitsparameter 94, 181, 182, 185
GFR *siehe* Guaranteed Frame Rate
Gigabit-Wissenschaftsnetz 58
Grundanwendungsfälle 141
Grundvorgänge 118
Guaranteed Frame Rate 100
Guaranteed Quality of Service 48

H

HTTP *siehe* Hypertext Transfer Protocol
Hypertext Transfer Protocol 111

I

ICMP... *siehe* Internet Control Message Protocol
IETF *siehe* Internet Engineering Task Force
Individualdienst 202, 219
Instanziierungskorridor 119, 126
Integrated Services 18, 47, 83, 111, 116
Integrated Services Digital Network 12
Interaktionsereignisse 121
International Organization for Standardization ..
18
International Telecommunication Union –
Telecom Standardization 9, 46
Internet Control Message Protocol 22
Internet Engineering Task Force 10
IntServ *siehe* Integrated Services
ISDN . *siehe* Integrated Services Digital Network
ISO *siehe* International Organization for
Standardization
ITU-T *siehe* International Telecommunication
Union – Telecom Standardization

K

Kardinalitäten 110, 188
Kernnetzprovider 17
Klassifikation 212
Kommunikationsrelation 159
Kommunikationsrichtung 143, 160, 166
Koppelungssymmetrie 142
Korrektheitsparameter 94, 182, 212
Kunde 25

L

LAL *siehe* Layer - Aspect - Life cycle
LAN *siehe* Local Area Network
Layer - Aspect - Life cycle 64
Lebensabschnittskorridor 119, 126, 139, 197
Lebenskorridor 119
Lebenszyklus 32, 106, 117, 121, 126, 197, 204
Local Area Network 47

M

management functionality 72
Managementfunktionalität 72, 130, 199
Massendienst 202
maxCDT *siehe* maximum Cell Transfer Delay
Maximum Burst Size 97
maximum Cell Transfer Delay 97
Maximum Frame Size 97
MBMS . *siehe* Multimedia Broadcast / Multicast
Service
MBS *siehe* Maximum Burst Size
MCR *siehe* Minimum Cell Rate
MFS *siehe* Maximum Frame Size
Minimum Cell Rate 97
MNM-Dienstmodell 25, 72, 126, 130
MPLS *siehe* Multi Protocol Label Switching
MPOA *siehe* Multi-Protocol Over ATM
Multi Protocol Label Switching 20, 24, 61
Multi-Protocol Over ATM 47
Multicast 87, 160
Multicast 1:n 87
Multicast m:n 87
Multimedia Broadcast / Multicast Service .. 227

N

Network-Network Interface 21
Netz-QoS-Architektur 19
Netz-QoS-Dienste 27
Netz-QoS-Dienstinstanz 27
NNI *siehe* Network-Network Interface
non-real-time Variable Bit Rate 100
nrtVBR *siehe* non-real-time Variable Bit Rate
Nutzer 25
Nutzfunktionalität 72, 126
Nutzungsphase 108

O

Open Shortest Path First 22, 79
Open Systems Interconnection 18
OSI *siehe* Open Systems Interconnection
OSPF *siehe* Open Shortest Path First
Overprovisioning 3

P

Paketgrenze 86, 167
PCR *siehe* Peak Cell Rate
Peak Cell Rate 97

peak-to-peak Cell Delay Variation 97
 peakToPeakCDV.. *siehe* peak-to-peak Cell Delay
 Variation
 Per Hop Behaviour 49, 88
 PHB *siehe* Per Hop Behaviour
 Planungsphase 107, 204
 Point-of-Presence 16
 PoP *siehe* Point-of-Presence
 provider 25
 provider side 25
 Providerhierarchie 18, 31, 35, 37, 48

Q

QoS *siehe* Quality-of-Service
 QoS parameters 72
 QoS-Kategorie 20, 97, 172
 QoS-Parameter 20, 90, 173
 Quality-of-Service 1
 Quelldienst 142

R

real-time Variable Bit Rate 99
 Relative Garantie 93
 Request For Comments 97
 Resource Reservation Setup Protocol 48, 86
 RFC *siehe* Request For Comments
 RIP *siehe* Routing Information Protocol
 Routing Information Protocol 79
 RSVP *siehe* Resource Reservation Setup
 Protocol
 rtVBR *siehe* real-time Variable Bit Rate

S

Schmitt, Modell von 52
 SCR *siehe* Sustained Cell Rate
 SDU *siehe* Service Data Unit
 Semantik 90, 175
 service 25, 72
 service agreement 74
 service client 75
 Service Data Unit 85, 86
 service implementation 75, 78
 Service Level Agreement 107
 service management implementation 75
 Sichten 72
 SLA *siehe* Service Level Agreement
 Spontanereignisse 120
 Standarddienst 202
 Standardisierungsphase 107, 212
 Statistische Garantie 93
 Strom-Protokolle 86
 Sustained Cell Rate 97
 SVC *siehe* Switched Virtual Circuit
 Switched Virtual Circuit 21
 Symmetrische Koppelung 142
 Szenarios 12

T

Taxonomie 212
 TCP *siehe* Transmission Control Protocol
 Teilfragestellungen 4, 207, 210, 213
 Terminologie 213
 ToS *siehe* Type of Service
 Transitiv-Koppelung 31, 142
 Transmission Control Protocol 87
 Tunnel-Koppelung 31, 142
 Tunneldienst 142
 Type of Service 102

U

UBR *siehe* Unspecified Bit Rate
 UDP *siehe* User Datagram Protocol
 UML *siehe* Unified Modeling Language
 UMTS *siehe* Universal Mobile
 Telecommunications System
 UNI *siehe* User-Network Interface
 Unicast 87, 160
 Unicast bidirektional 87
 Unicast unidirektional 87
 Unidirektionale Kommunikationsrichtung...
 143, 160
 Unified Modeling Language 26
 Universal Mobile Telecommunications System...
 19, 50
 Unspecified Bit Rate 20, 100
 usage functionality 72
 user 25
 User Datagram Protocol 87
 User-Network Interface 21

V

VCI *siehe* Virtual Channel Identifier
 Veränderungskorridor 119, 126
 Verbindungsorientiert 85, 155
 Vereinbarungsphase 107
 Verhandlungsphase 204
 Verwerfen 88
 views 72
 Virtual Channel Identifier 102
 Virtual Path Identifier 102
 Virtual-Private-Network 13
 Voice-over-IP 12
 VoIP *siehe* Voice-over-IP
 Volumenparameter 93, 181, 185
 VPI *siehe* Virtual Path Identifier
 VPN *siehe* Virtual-Private-Network

W

Wertemenge 91, 178

Z

zentrale Austauschpunkte 17, 221
 Zieldienst 142
 Zugangsprovider 16

SCHLAGWORTINDEX

Lebenslauf

Name		Harald Rölle
Geburtsdatum		7.8.1971
Geburtsort		München
Schulbildung	09.1978 - 07.1982	städt. Grundschule an der Regina-Ullmann-Straße, München
	09.1982 - 07.1991	staatl. Luitpold-Gymnasium, München Abschluss: allgemeine Hochschulreife
Zivildienst	12.1991 - 02.1993	Klinik und Poliklinik für Psychiatrie und Psychotherapie der Ludwig-Maximilians-Universität München, Abteilung für Klinische Psychologie und Psychophysiologie
Studium	11.1993 - 05.2000	Informatik (Nebenfach Mathematik), Technische Universität München Abschluss: Diplom-Informatiker Univ.
Beruf	seit 06.2000	Wissenschaftlicher Mitarbeiter am Lehrstuhl für Kommunikationssysteme und Systemprogrammierung (Prof. Dr. Heinz-Gerd Hegering) am Department Institut für Informatik der Ludwig-Maximilians-Universität München

