



**FLEXIBLE ABRECHNUNGSSYSTEME FÜR  
DIGITALE PRODUKTE IM E-LEARNING**

**MARKUS GRÜNE**

---

*édition scientifique*  
**VVB LAUFERSWEILER VERLAG**

**Das Werk ist in allen seinen Teilen urheberrechtlich geschützt.**

Jede Verwertung ist ohne schriftliche Zustimmung des Autors oder des Verlages unzulässig. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung in und Verarbeitung durch elektronische Systeme.

1. Auflage 2007

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the Author or the Publishers.

1<sup>st</sup> Edition 2007

© 2007 by VVB LAUFERSWEILER VERLAG, Giessen  
Printed in Germany



**VVB LAUFERSWEILER VERLAG**  
édition scientifique

STAUFENBERGRING 15, D-35396 GIESSEN  
Tel: 0641-5599888 Fax: 0641-5599890  
email: [redaktion@doktorverlag.de](mailto:redaktion@doktorverlag.de)

[www.doktorverlag.de](http://www.doktorverlag.de)

# **Flexible Abrechnungssysteme für digitale Produkte im E-Learning**

**Markus Grüne**



## Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit am Fachbereich Wirtschaftswissenschaften an der Johann Wolfgang Goethe-Universität. Das Vorwort möchte ich dazu nutzen, all denen zu danken, die einen direkten oder indirekten Beitrag zur Arbeit geleistet haben, sei es durch die Unterstützung des Themas, durch fachliche Diskussion der Inhalte, durch die Förderung der kollegialen Atmosphäre an den Lehrstühlen oder durch mentale und nicht zuletzt finanzielle Unterstützung.

Zu besonderem Dank bin ich meinem Betreuer, Herrn *Prof. Dr. Andreas Oberweis*, verpflichtet, der sich des Themas angenommen hat und mir in den ersten Jahren der Promotion ermöglichte, reichhaltige Erfahrungen am „Lehrstuhl für Entwicklung betrieblicher Informationssysteme“ zu sammeln. Dazu gehörte auch die Mitarbeit im BMBF-Teilprojekt „Flexible Abrechnungssysteme“.

Nach dem Ruf von Professor Dr. Oberweis nach Karlsruhe übernahm Herr *Prof. Dr. Roland Holten* das BMBF-Teilprojekt und ich fand eine zweite Heimat an seinem neu gegründeten Lehrstuhl für „Information Systems Engineering“. Aus den intensiven Diskussionen in dieser Zeit entstammen insbesondere die umfangreichen Ausführungen zur Sprachtheorie, die sich durch die Arbeit ziehen. Auch Herrn Prof. Dr. Roland Holten, der zudem die Rolle des Zweitgutachters der Arbeit übernahm, bin ich zu besonderem Dank verpflichtet.

Ich freue mich außerdem sehr darüber, dass Herr *Prof. Dr. Kai Rannenberg* und Herr *Prof. Dr. Roland Eisen* sofort einverstanden waren, als Mitglieder der Prüfungskommission meine Arbeit zu bewerten. Aus eigener Erfahrung bei Diplomarbeiten weiß ich, dass bereits das Lesen einer Arbeit viel Zeit „schluckt“.

Das Leben an einem Lehrstuhl zeichnet sich dadurch aus, dass viele Personen kommen und gehen. Daneben gibt es eine kleine Schar von Festangestellten, die der Verwaltungsmühle Universität Tag für Tag die Stirn bieten und dem Lehrstuhl ein Gesicht und ihre persönliche Note geben. Danken möchte ich vor allem meinem langjährigen Wegbegleiter *Abdelghani Zafa*, der mir fachlich und persönlich immer zur Seite stand und den ich bereits seit meiner Zeit als studentische Hilfskraft kenne. Auch *Sabine Landvogt*, die das Sekretariat von Prof. Dr. Oberweis leitete, hatte stets ein offenes Ohr für sämtliche Probleme und

Fragen der Mitarbeiter und hauchte dem Lehrstuhl Oberweis – nicht nur den Hiwis – etwas Kultur ein. *Sandra Kessler* danke ich dafür, dass sie während der „Kriegswirren“ nach dem Wechsel des Lehrstuhlinhabers die Sekretariatsführung übernommen hat. Außerdem danke ich ihr für die Anschaffung des Espressoautomaten (☺), die vielen persönlichen Gespräche und Tipps für die Zeit nach der Uni.

In Karlsruhe möchte ich mich herzlich bei *Michaela Fischer* (Sekretariat der Forschungsgruppe von Prof. Dr. Oberweis) sowie bei *Markus Zaich* (Administrator) für die tatkräftige Unterstützung bedanken.

Einen wesentlichen Beitrag zur Arbeit leisteten *Gerlinde Löhberg* und *Günter Brinkmann*, indem sie die Rechtschreibung kritisch überprüften. Für mich ein Glücksgriff, denn wer hat schon zwei Lehrer an der Hand, die die Klippen der sprachlichen (Un-)Möglichkeiten – inklusive der neuen Rechtschreibung – kennen. Vielen Dank!

Mein Dank richtet sich auch an alle wissenschaftlichen und studentischen Mitarbeiter sowie „externe“ Doktoranden der Lehrstühle.

Die Mannschaft der ersten Halbzeit (Lehrstuhl Oberweis) in Frankfurt bildeten *Kirsten Lenz*, *Kirsten Keferstein*, *Agnes Koschmider*, *Andrijana Mandaric*, *Marco von Mevius*, *Oliver Paulzen*, *Primoz Perc*, *Hagen Sexauer*, *Swen Schneider*, *Volker Guth*, *Jürgen Powik*, *Alexander Nikolopoulos* (auch Spieler der zweiten Halbzeit) und *Carmelo Bonanno* (Dauer-Hiwi).

Viele Spieler der zweiten Halbzeit (Prof. Holten) wurden aus dem Münsterland „angeworben“. Neben *Christoph Rosenkranz* zählen zu den „Münsteranern“ *Matthias Biggeleben* und *Alexander Hoffmann*. Am Ende der zweiten Halbzeit wurde das Team durch die Frankfurter Spieler *Marianne Corvera Vargas* und *Harald Kolbe* erweitert.

Auch den Diplomanden, deren Diplomarbeiten einen Einfluss auf die Arbeit hatten, möchte ich für die gute Zusammenarbeit danken. Dies waren *Dinka Vugdalic*, *Ina Höppner*, *Marc Stranz* und *Elena Korschenewski*.

Bei der Weiterentwicklung der Prototypen half mir *Stefan Schmidt*, der trotz enger Zeitrahmen („Das schaffe mer schon!“) immer die Ruhe behielt und die Espressomaschine einem Stress-Test unterzog.

Mit dem Ende der Dissertation ist nun auch meine Zeit an der Uni Frankfurt zu Ende gegangen. Meinen neuen Kollegen bei der Deutschen Bank in Frankfurt-Sossenheim (GTO) *Alexandra Pelzl, Lydia Gracia-Esteban, Diana Fertig, Sabine Wolf, Katrin Fritzsich, Andreas Stipp, Thomas Zech* und *Mark Wolter* möchte ich für die nette Aufnahme in die „Kommune 1“ und für die familiäre Atmosphäre danken.

Bedanken möchte ich mich auch bei meinen Eltern *Christa* und *Franz-Josef Grüne*, die dafür gesorgt haben, dass ich (1) da bin, dass ich (2) da bin, wo ich jetzt bin und die (3) immer für mich da sind, wenn es darauf ankommt.

Zum Schluss möchte ich meiner Frau *Corinna Löhberg-Grüne* dafür danken, dass sie soviel Energie hat und beim Feuern vieler Transitionen in meinem Leben maßgeblich beteiligt ist. Und dass sie mich neben den Tieren auf und unter dem Sezier-Tisch (*Maya, Nelli* und *Juno*) sowie der Arbeit an ihrer eigenen Dissertation unterstützt hat. In meiner Welt aus „Kreisen und Vierecken“ sorgte sie für das nötige Maß an Ablenkung, Durchhalteparolen sowie eine ausreichende Vitamin- und Nahrungszufuhr.

*Wetzlar-Dutenhofen, Juli 2007*

*Markus Grüne*

# Inhaltsverzeichnis

<b>VORWORT</b> .....	<b>I</b>
<b>INHALTSVERZEICHNIS</b> .....	<b>IV</b>
<b>ABBILDUNGSVERZEICHNIS</b> .....	<b>XI</b>
<b>TABELLENVERZEICHNIS</b> .....	<b>XV</b>
<b>ABKÜRZUNGSVERZEICHNIS</b> .....	<b>XVI</b>
<b>1 EINLEITUNG</b> .....	<b>1</b>
<b>1.1 Problemstellung</b> .....	<b>3</b>
<b>1.2 Zielsetzung der Arbeit</b> .....	<b>4</b>
<b>1.3 Methodische Vorgehensweise</b> .....	<b>5</b>
<b>1.4 Gang der Untersuchung</b> .....	<b>6</b>
<b>2 HANDEL DIGITALER PRODUKTE IM E-LEARNING</b> .....	<b>9</b>
<b>2.1 Elektronische Handelsbereiche</b> .....	<b>9</b>
<b>2.2 Ein Framework zur Klassifizierung digitaler Produkte</b> .....	<b>12</b>
2.2.1 Produkte, Rechtstitel und Dienstleistungen.....	13
2.2.2 Digitale Informations- und Netzeffektgüter .....	16
2.2.3 Klassifizierung digitaler Produkte nach Hui und Chau.....	18
2.2.4 Erweiterungen des Frameworks .....	20
<b>2.3 Eigenschaften digitaler Produkte</b> .....	<b>22</b>
2.3.1 Veränderbarkeit .....	22
2.3.2 Reproduzierbarkeit .....	22
2.3.3 Kostenstruktur .....	22
2.3.4 Nicht-Abnutzbarkeit und Kapazitätsbeschränkung.....	23



2.3.5	Lager- und Transportkosten .....	23
2.3.6	Produkte als Eigenschaftsbündel und Produktlinien .....	24
2.3.7	Preisdeterminanten als Basis der Preisdifferenzierung .....	25
<b>2.4</b>	<b>Preisdifferenzierung bei digitalen Produkten.....</b>	<b>27</b>
2.4.1	Preisdifferenzierung im Internet.....	27
2.4.2	Ziele der Preisdifferenzierung .....	28
2.4.3	Grade der Preisdifferenzierung nach Pigou.....	28
2.4.4	Preisdifferenzierung nach Güterart und Kostenstruktur .....	31
2.4.5	Implementationsformen der Preisdifferenzierung .....	33
2.4.6	Preisdifferenzierung und E-Learning .....	37
2.4.6.1	Preisstrategien im E-Learning.....	37
2.4.6.2	Wahl der Implementationsform .....	39
<b>2.5</b>	<b>Vermarktung von E-Learning als digitales Produkt .....</b>	<b>39</b>
2.5.1	Lernobjekte als digitale Produkte im E-Learning.....	41
2.5.2	Digitale Dienstleistungen im E-Learning .....	43
2.5.3	Geschäftsmodelle im E-Learning .....	44
2.5.4	Beispiel: Direktvertrieb von E-Learning als Geschäftsmodell.....	45
2.5.5	Anwendungsfall: E-Learning am Penn State World Campus .....	48
2.5.5.1	Produktmodell.....	48
2.5.5.2	Erlösmodell.....	49
2.5.5.3	Preismodell .....	49
2.5.6	Beispiele für virtuelle Hochschulen und Studiengänge in Europa ....	51
2.5.6.1	European Online-Academy.....	51
2.5.6.2	Fernfachhochschule Schweiz.....	52
2.5.6.3	Virtual Global University und Winfoline .....	52
2.5.7	Herausforderungen bei der Umsetzung von E-Learning.....	53
<b>2.6</b>	<b>Zusammenfassende Betrachtung .....</b>	<b>54</b>
<b>3</b>	<b>ABLAUFMODELLIERUNG MIT PETRI-NETZEN.....</b>	<b>55</b>
<b>3.1</b>	<b>Elementare Petri-Netze .....</b>	<b>56</b>
3.1.1	Formale Definition .....	56
3.1.2	Grafische Darstellung.....	57
3.1.3	Strukturelle und dynamische Analyse von Petri-Netzen.....	58

<b>3.2</b>	<b>Stellen/Transitionen-Netze.....</b>	<b>58</b>
<b>3.3</b>	<b>Gefärbte Petri-Netze.....</b>	<b>59</b>
3.3.1	Überblick .....	59
3.3.2	Definition.....	60
3.3.3	Eigenschaften .....	63
3.3.4	Annotation von CP-Netzen mit CPN ML .....	63
3.3.5	Sprachbestandteile von CPN ML .....	64
3.3.5.1	Variablen.....	65
3.3.5.2	Referenzvariablen .....	65
3.3.5.3	Funktionen .....	66
3.3.5.4	Konstanten .....	67
3.3.5.5	Ausdrücke / Net-Expressions.....	68
3.3.5.6	Colour-Sets .....	68
3.3.6	Beispiel für ein CP-Netz im E-Learning .....	71
3.3.6.1	Codeblöcke als Transitionsinschriften in CP-Netzen.....	74
3.3.7	Analyse von CP-Netzen .....	75
3.3.7.1	Erreichbarkeitsanalyse mit CP-Netzen .....	76
<b>3.4</b>	<b>Zusammenfassende Bewertung.....</b>	<b>80</b>
<b>4</b>	<b>MODELLE UND SYSTEMENTWICKLUNG.....</b>	<b>81</b>
<b>4.1</b>	<b>Modellbegriff.....</b>	<b>81</b>
<b>4.2</b>	<b>Bewertung von Modellen und Beschreibungssprachen .....</b>	<b>83</b>
<b>4.3</b>	<b>Modellbildung – Überblick und Definitionen .....</b>	<b>85</b>
4.3.1	Intensionale Abstraktion.....	86
4.3.2	Extensionale Abstraktion.....	88
<b>4.4</b>	<b>Vorgehen zur Gestaltung von Modellierungssprachen.....</b>	<b>88</b>
4.4.1	Entwurf einer Orthosprache (Schritt 1) .....	88
4.4.2	Entwurf der Notationen (Schritt 2).....	89
4.4.3	Entwurf der Prozessunterstützung für den Anwender (Schritt 3).....	90
<b>4.5</b>	<b>Rahmenmodell zur IS-Entwicklung .....</b>	<b>91</b>

<b>5</b>	<b>PRIMOL – EINE BESCHREIBUNGSSPRACHE FÜR PREISMODELLE.....</b>	<b>97</b>
<b>5.1</b>	<b>Produktmodelle.....</b>	<b>97</b>
<b>5.2</b>	<b>Kundengruppenmodelle.....</b>	<b>99</b>
<b>5.3</b>	<b>Preismodelle .....</b>	<b>100</b>
5.3.1	Überblick .....	100
5.3.2	Pricing von Versicherungsprodukten .....	101
5.3.2.1	Kraftfahrtversicherung.....	102
5.3.2.2	Tarifierung und Tarifmerkmale der Kfz-Haftpflichtversicherung .....	102
5.3.2.3	Tarifierung und Tarifmerkmale der Kasko-Versicherung.....	103
5.3.2.4	Prämiendifferenzierung.....	104
5.3.2.5	Tarifmerkmalsbestimmung .....	104
5.3.2.6	Kalkulation von Tarifen mit Tariffaktoren .....	105
5.3.3	Pricing digitaler Produkte im Mobilfunk.....	106
5.3.3.1	Tarifmodellierung .....	108
5.3.3.2	Tarife für Sprachdienste.....	109
<b>5.4</b>	<b>PriMoL – eine Beschreibungssprache für Preismodelle.....</b>	<b>109</b>
5.4.1	Sprachkonzepte.....	109
5.4.1.1	Einfache Preise .....	110
5.4.1.2	Komplexe Preise.....	111
5.4.1.3	Aggregationsregeln.....	113
5.4.2	Struktur eines Preismodells .....	113
5.4.3	Definition der Orthosprache .....	114
5.4.4	Definition des fachkonzeptuellen Metamodells .....	121
5.4.5	Definition einer grafischen Notation .....	122
5.4.6	Definition einer relationalen Notation.....	124
5.4.6.1	Einfacher Preis.....	126
5.4.6.2	Komplexer Preis .....	127
5.4.6.3	Aggregationsregel.....	129
5.4.6.4	Auswertung von Preismodellen .....	130
<b>5.5</b>	<b>Einsatzmöglichkeiten und Bewertung von PriMoL .....</b>	<b>131</b>
5.5.1	Abbildung von Kfz-Haftpflichttarifen mit PriMoL.....	131
5.5.2	Abbildung von Mobilfunktarifen mit PriMoL .....	131
5.5.3	Bewertung anhand der Kriterien nach (Zelewski 1996).....	133

5.5.4	Bewertung anhand der GoM nach (Schütte 1998) .....	135
<b>5.6</b>	<b>Schlussbetrachtung und Kritik .....</b>	<b>137</b>
<b>6</b>	<b>ENTWICKLUNG EINES PROTOTYPS ZUR PREISMODELLIERUNG.....</b>	<b>139</b>
<b>6.1</b>	<b>Architektur-getriebene Systementwicklung .....</b>	<b>140</b>
6.1.1	Architekturbegriff.....	140
6.1.2	Architektur-getriebene Entwicklung .....	141
<b>6.2</b>	<b>IS-Architektur.....</b>	<b>142</b>
<b>6.3</b>	<b>Modellarten zur Beschreibung von IS-Architekturen .....</b>	<b>143</b>
6.3.1	Strukturmodelle .....	144
6.3.2	Funktionale Modelle.....	144
6.3.3	Organisationsmodelle .....	144
<b>6.4</b>	<b>Systemarchitektur – Komponenten von Abrechnungssystemen .....</b>	<b>146</b>
6.4.1	Kontextmodell .....	146
6.4.2	Komponenten .....	147
6.4.2.1	Usage-Data-Acquisition-Komponente.....	147
6.4.2.2	Data- & Process-Management-Komponente .....	148
6.4.2.3	Financial-Settlement-Komponente .....	148
6.4.2.4	Information-Provisioning-Komponente.....	150
<b>6.5</b>	<b>Vorgehensmodell zur Architektur-getriebenen Entwicklung eines Price-Modeler-Prototyps.....</b>	<b>150</b>
<b>6.6</b>	<b>Entwicklungsarchitektur .....</b>	<b>151</b>
6.6.1	Auswahl der zu implementierenden Funktionen .....	151
6.6.2	Statische Entwicklungsarchitektur .....	153
6.6.2.1	Präsentationsschicht.....	153
6.6.2.2	Model-Mapping-Layer.....	154
6.6.2.3	Anwendungsschicht.....	154
6.6.2.4	Persistenzschicht.....	155
6.6.2.5	Datenbankschicht.....	155
6.6.3	Dynamische Entwicklungsarchitektur .....	155
6.6.3.1	Anwendungsfälle .....	156

6.6.3.2	Modelle verwalten .....	157
6.6.3.3	Datenbank-Verbindungen verwalten .....	159
6.6.3.4	Modellkomponente anlegen oder löschen.....	160
6.6.3.5	Komplexen Preis anlegen .....	161
6.6.3.6	Beispiel: Umsetzung eines CP-Netzes in einen Wizard.....	162
6.6.3.7	Einfachen Preis anlegen.....	166
6.6.3.8	Eventbasierten Preis anlegen .....	166
6.6.3.9	Zeitbasierten Preis anlegen .....	168
6.6.3.10	Aggregationsregel anlegen.....	169
6.6.4	Paketstruktur des Price-Modeler-Prototyps.....	170
<b>7</b>	<b>EIN LERN-MANAGEMENTSYSTEM MIT INTEGRIERTEM PRICING.....</b>	<b>172</b>
<b>7.1</b>	<b>Ein prototypisches Lern-Managementsystem .....</b>	<b>172</b>
7.1.1	Entwurf eines prototypischen Lern-Managementsystems.....	172
7.1.2	Softwaretechnische Entwicklungsarchitektur des LMS .....	173
7.1.3	Anwendungsfälle .....	174
7.1.4	Anwendungsfälle aus Administrator-Sicht.....	174
7.1.4.1	Anlegen eines Benutzers.....	174
7.1.4.2	Anlegen eines Kurses.....	175
7.1.4.3	Lektion hinzufügen.....	175
7.1.4.4	Preismodell zuweisen .....	176
7.1.4.5	Anlegen von Übungen und Downloads .....	177
7.1.5	Anwendungsfälle aus Sicht der Lernenden .....	178
7.1.5.1	Buchung eines Kurses und Auswahl eines Preismodells .....	178
7.1.5.2	Aktivierung der Nutzung von Lernobjekten .....	180
7.1.6	Eine Laufzeitumgebung für Preismodelle .....	180
<b>7.2</b>	<b>Portal-Architektur eines Workflow-gestützten LMS.....</b>	<b>182</b>
7.2.1	Workflows im E-Learning.....	182
7.2.2	Portalarchitektur eines Lern-Managementsystems.....	186
7.2.3	Bewertung der Portalarchitektur für Lern-Managementsysteme ....	188
<b>8</b>	<b>SCHLUSSBETRACHTUNG .....</b>	<b>190</b>
<b>8.1</b>	<b>Zusammenfassung .....</b>	<b>190</b>
<b>8.2</b>	<b>Trends und Fortentwicklung der Konzepte.....</b>	<b>192</b>

<b>ANHÄNGE</b> .....	<b>196</b>
<b>A.1 Virtuelle Studiengänge, virtuelle Universitäten und Suchmaschinen für E-Learning</b> .....	<b>196</b>
<b>A.2 Sprachdefinition</b> .....	<b>202</b>
A.2.1 Accounting-Language-Definition in EBNF .....	202
A.2.2 XML-Schema von PriMoL.....	203
<b>A.3 Petri-Netze und Software Engineering</b> .....	<b>206</b>
A.3.1 Struktureigenschaften von Petri-Netzen .....	206
A.3.2 Dynamische Eigenschaften von Petri-Netzen .....	207
A.3.3 Stellen/Transitionen-Netze .....	211
A.3.4 Petri-Netze im Software Engineering.....	214
A.3.5 CPN ML .....	219
<b>A.4 Abrechnungssysteme – Begriffsdefinitionen</b> .....	<b>222</b>
<b>A.5: Industrielle Abrechnungssysteme</b> .....	<b>229</b>
<b>LITERATURVERZEICHNIS</b> .....	<b>240</b>

## Abbildungsverzeichnis

Abbildung 1: Aufbau der Arbeit.....	8
Abbildung 2: Einteilung des E-Commerce in Handelsbereiche.....	10
Abbildung 3: Arten des elektronischen Handels.....	11
Abbildung 4: Taxonomie digitaler Produkte.....	16
Abbildung 5: Klassifizierung digitaler Produkte anhand der Produktkategorie und -eigenschaften.....	19
Abbildung 6: Preisdeterminanten für digitale Produkte.....	26
Abbildung 7: Typen und Implementationsformen der Preisdifferenzierung.....	35
Abbildung 8: Klassifizierung der Preisdifferenzierung nach der Selbstselektion.....	36
Abbildung 9: Einführung von Preisdifferenzierungsstrategien.....	37
Abbildung 10: Partialmodelle eines Geschäftsmodell.....	44
Abbildung 11: Zweistufigkeit der Erlösentscheidung.....	45
Abbildung 12: Erlösformen im E-Learning.....	46
Abbildung 13: Erlösquellen und Preismodell des Penn State World Campus.....	51
Abbildung 14: Elementares Petri-Netz.....	57
Abbildung 15: CP-Netz eines einfachen Übertragungsprotokolls in CPN Tools- Notation.....	73
Abbildung 16: Codeblock für die Transition ReceivePacket.....	75
Abbildung 17: Ein einfaches CP-Netz.....	77
Abbildung 18: State-Space-Report für das CP-Netz aus Abbildung 17.....	79
Abbildung 19: Modellierungsgüte als Kriterium zur Evaluation von Petri-Netz- Typen.....	85
Abbildung 20: Modellierungsfähigkeit als Kriterium zur Evaluation von Petri- Netz-Typen.....	85
Abbildung 21: Metaebene durch zweifache extensionale Abstraktion, aus (Holten 2003b), S. 109.....	87
Abbildung 22: Methodologie zur Entwicklung von Modellierungstechniken für Fachkonzepte.....	89
Abbildung 23: Metamodell, Modell, Vorgehen.....	91
Abbildung 24: Rahmenmodell zur Entwicklung nach (Holten 2002), S. 18.....	96
Abbildung 25: Kundenorientierte Preispolitik aus Anbietersicht.....	97

Abbildung 26: CP-Netz zur Ermittlung von Tarifmerkmalen.....	105
Abbildung 27: Wertearray für drei Merkmalsdimensionen .....	111
Abbildung 28: Stark vereinfachte Struktur eines statischen Preismodells.....	114
Abbildung 29: konzeptuelles Modell der Beschreibungssprache für Preismodelle .....	123
Abbildung 30: Verbuchung eines einfachen Preises .....	126
Abbildung 31: Einfacher Preis als Relationenschema.....	127
Abbildung 32: Werteermittlung im komplexen Preis.....	128
Abbildung 33: Komplexer Preis als Relationenschema .....	129
Abbildung 34: Aggregationsregel als Relationenschema.....	130
Abbildung 35: Geschäfts-, Projekt-, System- und Entwicklungsarchitektur.....	141
Abbildung 36: Verwendung des Begriffs Architektur in der Arbeit.....	143
Abbildung 37: Berechtigungskonzept des dem E-Learning-System aus Kapitel 7.1 zugrunde liegenden Content-Management-Systems .....	145
Abbildung 38: Petri-Netz zur Ermittlung der Anwenderberechtigung für eine Funktion des E-Learning-Systems aus Kapitel 7.1.....	145
Abbildung 39: Anbindung eines SPS an ein Abrechnungssystem.....	146
Abbildung 40: Commercial-Exploitation-Komponenten eines Abrechnungssystems .....	149
Abbildung 41: Petri-Netz des Vorgehensmodells zur Entwicklung des Price- Modeler-Prototyps .....	151
Abbildung 42: Anwendungsfälle des Prototyps als CP-Netz.....	157
Abbildung 43: GUI des Price-Modelers.....	157
Abbildung 44: Anwendungsfall Modelle verwalten als CP-Netz.....	158
Abbildung 45: Datenbank-Verbindungen verwalten als CP-Netz .....	159
Abbildung 46: Mountpoint anlegen als CP-Netz .....	159
Abbildung 47: Maske zum Anlegen eines Mountpoints im Price-Modeler.....	160
Abbildung 48: Modellkomponente anlegen oder löschen als CP-Netz.....	160
Abbildung 49: Komplexen Preis anlegen als CP-Netz.....	162
Abbildung 50: Komplexen Preis anlegen - Beschreibung angeben .....	163
Abbildung 51: Komplexen Preis anlegen - SQL-String definieren.....	163
Abbildung 52: Komplexen Preis anlegen - Kunden identifizieren und Dimensionen anlegen.....	164
Abbildung 53: Komplexen Preis anlegen - Dimensionen skalieren.....	165



Abbildung 54: Komplexen Preis anlegen - Preiszähler erfassen.....	165
Abbildung 55: Komplexen Preis anlegen - AccLan-Ausdruck definieren.....	166
Abbildung 56: Einfachen Preis anlegen als CP-Netz.....	167
Abbildung 57: Eventbasierten Preis anlegen als CP-Netz.....	167
Abbildung 58: Zeitbasierten Preis anlegen als CP-Netz.....	168
Abbildung 59: Aggregationsregel anlegen als CP-Netz.....	169
Abbildung 60: Präzisierung durch Definition von Colour-Sets für Stellen.....	170
Abbildung 61: Java-Paketstruktur des Price-Modeler-Prototyps.....	171
Abbildung 62: Zusammenspiel des Price-Modelers mit einem LMS über eine Laufzeitumgebung.....	173
Abbildung 63: Anwendungsfälle aus der Sicht eines Administrators.....	174
Abbildung 64: LMS - GUI zur Anlage eines Benutzers.....	175
Abbildung 65: Anlegen eines Kurses.....	175
Abbildung 66: Hinzufügen einer Lektion zu einem Kurs.....	176
Abbildung 67: LMS - GUI zum Zuweisen eines Preismodells zu einem Kurs sowie zum Löschen von Preismodellen.....	177
Abbildung 68: Hinzufügen von Übungen und Downloads zu einer Lektion.....	177
Abbildung 69: Anwendungsfälle aus Sicht eines Lernenden.....	178
Abbildung 70: Kurs buchen.....	179
Abbildung 71: Preismodell auswählen für einen Kurs.....	179
Abbildung 72: Buchung von Lernobjekten.....	180
Abbildung 73: Berechnung eines Endpreises mit einem Preismodell.....	181
Abbildung 74: verkleinerte Anzeige eines Preismodells im LMS.....	182
Abbildung 75: Auswertung eines Preismodells für einen Lernenden.....	182
Abbildung 76: E-Learning- und Pricing-Prozesse.....	184
Abbildung 77: Korrektur eines Aufgabenblatts.....	185
Abbildung 78: Portal-Architektur eines verteilten LMS.....	187
Abbildung 79: Accounting Language AccLan als Erweiterte Backus-Naur-Form .....	202
Abbildung 80: XML-Schema von PriMoL in XML-Spy-Notation.....	203
Abbildung 81: XML-Schema-Repräsentation von PriMoL in Textform.....	205
Abbildung 82: Nebenläufigkeit in Petri-Netzen.....	208
Abbildung 83: Kausalität zweier Transitionen (Input-Concession).....	209
Abbildung 84: Kausalität zweier Transitionen (Output-Concession).....	209

Abbildung 85: Input- und Output-Konflikt .....	210
Abbildung 86: Kontakt in Elementaren Petri-Netzen .....	210
Abbildung 87: Dimensionen des Software-Engineerings und ausgewählte Literatur .....	218
Abbildung 88: Entwicklung von Software-Systemen mit Petri-Netzen – Einordnung von Petri-Netz-Konzepten .....	219

## Tabellenverzeichnis

Tabelle 1: Klassifizierungsschema für digitale Produkte .....	20
Tabelle 2: Framework zur Klassifizierung digitaler Produkte .....	21
Tabelle 3: Systematik der Preisdifferenzierung.....	32
Tabelle 4: Beispiele für Ausdrücke und deren Resultatstypen in CPN ML.....	68
Tabelle 5: Beispiele zur Definition von Colour Sets in CPN ML.....	70
Tabelle 6: Tarifmerkmal Jahresfahrleistung.....	106
Tabelle 7: Tarifmerkmal Geschlecht .....	106
Tabelle 8: Prämienermittlung bei zwei Tarifmerkmalen.....	106
Tabelle 9: Dienstprofile bei UMTS .....	108
Tabelle 10: arithmetische Bestimmung eines Preises .....	112
Tabelle 11: Definition der Orthosprache und Verknüpfung mit dem konzeptuellen Metamodell.....	115
Tabelle 12: Grafische Artefakte für Einfacher Preis, Komplexer Preis und Aggregationsregel.....	125
Tabelle 13: Events für einfachen Preis .....	127
Tabelle 14: Überführung von Tarifen in Preismodelle.....	132
Tabelle 15: Abbildung eines Mobilfunktarifs in Preismodellen .....	132
Tabelle 16: Kriterien „Konstruktivität“, „Analysierbarkeit“, „Adaptivität“ und „Implementierbarkeit“ als anwendungsnahe Qualitäten eines Modells ..	135
Tabelle 17: Kriterien „Konstruktionsadäquanz“, „Sprachadäquanz“, „Sprachrichtigkeit“, „Wirtschaftlichkeit“, „Klarheit“ und „Systematischer Aufbau“ zur Bewertung von Modellierungssprachen .....	136
Tabelle 18: Drei-Ebenen-Architektur des Price-Modelers mit Schichten.....	153
Tabelle 19: Ablaufstrukturen in Petri-Netzen .....	211

## Abkürzungsverzeichnis

Abs.	Absatz
ADD	Architecture-Driven Development
ADL	Advanced Distributed Learning Initiative
API	Application Programming Interface
AR	Aggregation Rule / Aggregationsregel
ASP	Application-Service-Providing
B/E-Netz	Bedingungs/Ereignis-Netz
BMBF	Bundesministerium für Bildung und Forschung
BNF	Backus-Naur-Form
BPEL4WS	Business Process Execution Language for Web Services
bspw.	beispielsweise
BSS	Base Station Subsystem
BSVC	Baltic Sea Virtual Campus
bzw.	beziehungsweise
C-Good	Conventional Good
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CBT	Computer-Based Training
CHF	Schweizer Franken
CP-Netz, CPN	Coloured Petri-Netz = Gefärbtes Petri-Netz
CRM	Customer Relationship Management
D-Commerce	Digital Commerce
DB	Datenbank
DOM	Document Object Model
DRM	Digital Rights Management
DSL	Digital Subscriber Line
DV	Datenverarbeitung
EBNF	Erweiterte Backus-Naur-Form
ECTS	European Credit Transfer and Accumulation System

E-...	Electronic ... , z. B. E-Commerce = Electronic Commerce, E-Learning = Electronic Learning, E-Mail = Electronic Mail usw.
engl.	Englisch
EOA	European Online Academy
ERD	Entity-Relationship-Diagramm
ERM	Entity-Relationship-Modell
ERP	Enterprise Resource Planning
etc.	et cetera
ETSI	European Telecommunications Standard Institute
EU	Europäische Union
FFHS	Fernfachhochschule Schweiz
FTP	File Transfer Protocol
ggf.	gegebenenfalls
GoM	Grundsätze ordnungsgemäßer Modellierung
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPD	http-Dämon = Webserver
ID	Identifizier
IDEF0	Integration Definition for Function Modeling
i. d. R.	in der Regel
i. e. S.	im engeren Sinne
IKT	Informations- und Kommunikationstechnologie
iMBA	Internet MBA
insb.	insbesondere
IS	Informationssystem
IT	Informationstechnologie
ITS	Intelligent Tutoring System
i. w. S.	im weiteren Sinn
JDBC	Java Database Connectivity
JDOM	DOM-Implementierung mit Erweiterungen für Java
JSP	Java Server Page

JSR	Java Standardization Request
J2EE	Java 2 Enterprise Edition
Kap.	Kapitel
KBit	kilobit, entspricht $2^{10} = 1.024$ bit
Kfz-HP	Kraftfahrzeug-Haftpflichtversicherung
K/I-Netz	Kanal/Instanzen-Netz
LCF	Logic for Computable Functions
LMS	Lern-Managementsystem
Math-ML	Mathematical Markup Language
MBA	Master of Business Administration
MBI	Master of Business Informatics
MMS	Multimedia Messaging Service
MSc	Master of Science
NSS	Network and Switching Subsystem
OASIS	Organization for the Advancement of Structured Information Standards
ODBC	Open Database Connectivity
OMG	Object Management Group
OSI	Open Systems Interconnect
OSS	Operation and Support System
PDF	Portable Document Format
PhD	Philosophical Doctor
PHP	Hypertext Pre-Processor
PREMIUM	BMBF-Forschungsprojekt "Preis- und Erlösmodelle im Internet – Umsetzung und Marktchancen"
PriMoL	Price Modeling Language
Proc.	Proceedings
Pr/T-Netz	Prädikate/Transitionen-Netz
PWC	Pennsylvania State University World Campus
s.	siehe
S.	Seite
SaaS	Software as a Service
SAX	Simple API for XML
SCORM	Sharable Content Object Reference Model

SML	Standard Meta Language
SMS	Short Message Service
s. o.	siehe oben
SOA	serviceorientierte Architektur
SP	Simple Price = Einfacher Preis
SPS	Service-Provisioning-System
SQL	Structured Query Language
SS	Sommersemester
S/T-Netz	Stellen/Transitionen-Netz
s. u.	siehe unten
SVG	Scalable Vector Graphics
u. a.	unter anderem, unter anderen
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
US\$	US-Dollar
usw.	und so weiter
vgl.	vergleiche
VGU	Virtual Global University
vs.	versus
WAN	Wide Area Network / Mobiltelefonnetz
WAP	Wireless Application Protocol
WfMS	Workflow-Managementsystem
WS	Wintersemester
WSDL	Web-Services Description Language
WSRP	Web-Services Remote Portlet Specification
WWW	World-Wide Web
W3C	World-Wide Web Consortium
XML	eXtensible Markup Language
XSDL	XML Schema Definition Language
z. B.	zum Beispiel
z. T.	zum Teil

# 1 Einleitung

E-Learning zeichnet sich als digitales Produkt dadurch aus, dass es sowohl über einen Produktcharakter (im Sinne herunterladbarer Dokumente), als auch einen Dienstleistungscharakter (Lehr- und Lernprozesse) verfügt.<sup>1</sup> Moderne Lern-Managementsysteme unterstützen Lernende und Lehrende bei der Durchführung von E-Learning-Aktivitäten und bieten die technischen Voraussetzungen, das Lernen hinsichtlich Ort, Zeit und der vermittelten Inhalte, z. B. durch Adaption der Inhalte an die Bedürfnisse der Lernenden oder durch gemeinsame Erstellung von Lernobjekten durch eine Gruppe von E-Learning-Providern (Content Syndication) zu flexibilisieren.<sup>2</sup> Die Vorteile des E-Learnings gegenüber traditionellen Lehrformen, die z. B. in der Orts- und Zeitunabhängigkeit gesehen werden können, haben dazu geführt, dass neben der Verbreitung von E-Learning im akademischen Bereich ein verstärkter Einsatz zur Unterstützung des „lebenslangen Lernens“ auch in der betrieblichen Fortbildung zu verzeichnen ist (Leithner und Back 2004), S. 2-4.

Nach einer Erhebung von (Allen und Seaman 2003) hat sich E-Learning in den USA unter den Studierenden als Alternative zum „Lernen im Klassenzimmer“ etabliert. Im Frühjahr 2002 nahmen ca. 1,6 Mio. Studierende an Online-Kursen teil. Von den Studierenden höherer Bildungseinrichtungen nahmen allein 11% an mindestens einem Online-Kurs teil. Für den europäischen E-Learning-Markt sind kaum Daten vorhanden.<sup>3</sup> Nach (Lutz und Pelka 2003), S. 108 ergeben die „insgesamt sehr uneinheitlich angelegten Studien zur Beschreibung des e-Learning-Markts nur ein lückenhaftes Bild. [...] Repräsentative Studien zur deutschen Marktsituation quer über alle Zielgruppen liegen [...] jedoch nicht vor.“

---

<sup>1</sup> (Bernecker 2005), Kap. 2 betrachtet Bildung aus gütertypologischer Sicht. Danach stellt „Bildung [...] trotz dinglicher Bestandteile [...] eine immaterielle Leistung dar.“ (S.13) Die Aussage gilt auch für elektronische Bildungsprodukte (E-Learning).

<sup>2</sup> Obwohl erste Prototypen für eine Workflowunterstützung der Lehr- und Lernprozesse existieren (Lin u.a. 2001) fehlt diese bisher in professionellen Lern-Managementsystemen (Cesarini u.a. 2004), (Gruene u.a. 2004), (Keferstein u.a. 2005).

<sup>3</sup> Eine Studie des Danish Technological Institutes kommt zum Schluss, dass in Europa statt eines einheitlichen Marktes nur nationale und technologie-bestimmte Teilmärkte existieren. (DTI 2005).



*E-Learning im Bereich der Hochschule* zeichnet sich in Deutschland „durch eine starke Finanzierung der Projekte mit Mitteln aus öffentlichen Förderprogrammen [aus].“ (Wache 2003).<sup>4</sup> Projektakteure stehen daher [bisher] nicht „unter dem Druck des schnellen Markterfolgs [...] [und haben] einen größeren Raum für experimentelle Erkundungen.“ Auf internationaler und europäischer Ebene ist dagegen ein verstärkter Fokus auf Kostenaspekte des E-Learnings und auf der Gestaltung von Umlageverfahren zur Umwälzung von Kosten festzustellen (Ernest&Young 2000; Bacsich u.a. 2001; Gerlach 2003; von Boettcher 2004).<sup>5</sup>

Erste Studien befassen sich im Rahmen von „E-Learning-Geschäftsmodellen“ auch mit der Generierung von Erlösen aus E-Learning im Hochschulbereich (vgl. z. B. (Bentlage 2002; Dohmen und Simons 2003; Euler und Seufert 2005)).<sup>6</sup> Dabei sehen (Dohmen und Simons 2003) im Hochschulbereich ein Erlöspotenzial gerade bei „abschlussorientierten Studienangeboten“. Zur Generierung von Erlösen bietet es sich für Hochschulen an, Produkte und Services nach Marktsegmenten zu differenzieren. Eine Quersubventionierung des traditionellen E-Learnings im Massenstudium ist dann durch erhöhte Erlöse im außeruniversitären Bereich, z. B. in Online-Executive Programmen oder in Online-Repetitorien, denkbar.<sup>7</sup>

Damit ein Anbieter von E-Learning seine E-Learning-Dienstleistung gegenüber den Kunden bepreisen und abrechnen kann, müssen geeignete Geschäftsmodelle identifiziert und umgesetzt werden. Dazu müssen in einem ersten Schritt die Bezugsobjekte (Erlösquellen), auf denen eine Gebührenberechnung erfolgen soll, festgesetzt werden. Ausgehend von der Wahl der Erlösquellen (Downloads, Dienstinanspruchnahme, Werbung etc.) und deren Dokumentation in einem

---

<sup>4</sup> Die Finanzierung von E-Learning-Aktivitäten an deutschen Hochschulen wird überwiegend durch Förderprojekte auf Hochschul-, Landes- und Bundesebene sichergestellt (vgl. Studien unter e-teaching.org, z. B. (Euler und Seufert 2005)). Analog (Schmees 2004), S. 185.

<sup>5</sup> Die Bewertung des E-Learning-Einsatzes mit Kosten ist die Voraussetzung für eine Kosten-Nutzen-Analyse. Diese ist gerade für Unternehmen, die den Erfolg ihrer E-Learning-Aktivitäten messen wollen sinnvoll. Bei der Bewertung können dann auch Einsparungen von Reisekosten und entgangener Arbeitszeit berücksichtigt werden. Eine Nutzenbewertung des E-Learning-Einsatzes ist für die Hochschule nahezu unmöglich. Nutzen, z. B. in Form von „Anwendbarkeit des Wissens im Berufsleben“ fällt nach Ansicht d. Verf. überwiegend bei den Kunden an, die als eigene „Wirtschaftseinheiten“ agieren.

<sup>6</sup> Zu Geschäftsmodellen im E-Learning s. auch (Seufert 2001; Hasebrook und Otte 2002; Scheffer und Hesse 2002; Hoppe und Breitner 2003; Reil und Appelrath 2004).

<sup>7</sup> Eine erste repräsentative Befragung im Projekt CALAMARES unter ca. 3000 Studierenden an der Universität Frankfurt ergab, dass Studierende auch bereit wären, für Online-Repetitorien Gebühren zu entrichten (Durmann 2006).

*Erlösmodell* können Preismodelle entwickelt werden, die den Bezugsobjekten Preise (Preisähler) zuordnen (vgl. z. B. (Skiera und Spann 2002)).<sup>8</sup> Die in Preismodellen beschriebenen Tarife müssen anschließend zur Abrechnung der digitalen E-Learning-Produkte in einem Abrechnungssystem implementiert werden.

## 1.1 Problemstellung

Obwohl für digitale Produkte, z. B. im Mobilfunk, eine Vielzahl unterschiedlicher Abrechnungssysteme am Markt vertrieben werden<sup>9</sup>, existiert kein System, das die Abrechnung von E-Learning-Produkten unmittelbar unterstützt. Darüber hinaus sind „existierende Abrechnungssysteme [für digitale Produkte] meist zu starr und nicht in der Lage, flexibel und schnell an neue Preismodelle oder personalisierte Produkt- bzw. Servicekonzepte im Internet angepasst zu werden.“ (PREMIUM 2003), S. 70.<sup>10</sup> Dies liegt zum einen daran, dass die aufgeführten Abrechnungssysteme für bestimmte Anwendungsdomänen und deren standardisierte Hardware (Router, Switches etc.) sowie Software (Operation-Support-Systems etc.) „optimiert“ bzw. in die Produkte selbst integriert<sup>11</sup> sind. Sie unterstützen daher i. d. R. nicht die wechselnden Anwendungslandschaften und den schnellen technologischen Fortschritt, die gerade im E-Learning anzutreffen sind.

---

<sup>8</sup> Die Gestaltung von Erlös- und Preismodellen kann sich dabei an anderen digitalen Produkten, z. B. in der Telekommunikationsbranche, orientieren. Neben Flat-Fee-Modellen, in denen Studierenden durch Zahlung einer monatlichen Gebühr Vollzugriff zum Lern-Managementsystem erhalten, sind zweiteilige Tarife denkbar, bei denen Gebühren in Abhängigkeit von der Dienstinanspruchnahme anfallen. Für die Definition einer effektiven Pricing-Strategie müssen aus Sicht des Marketings neben den Kosten der Erstellung, die Preisstrukturen der Konkurrenten sowie die Nutzenbewertung durch die Kunden berücksichtigt werden (vgl. (Nagle und Holden 1995), S. 1-35; (Monroe 2003), S. 11-12).

<sup>9</sup> Z. B. Danet, High Deal, Info Directions, Intec, Integris, Kenan Arbor, LavaStorm, Metasolv, Metratch, Mistral, Nazca, Netcracker, Nimbus Systems, nTel smart convergence, OmniOSS, Oracle Advanced Pricing, Orange Objects, Par3, Selcomm, Sigma Systems, Soprano, Subex Systems, Viziqor. S. auch Systeme im Anhang “A.5: Industrielle Abrechnungssysteme”, S. 229.

<sup>10</sup> Viele Abrechnungssysteme verlangen vom Anwender das Einpflegen von Preismechanismen in Tabellenform oder die Programmierung der Auswertungsregeln. Entscheider, die i. d. R. nicht über das nötige Expertenwissen verfügen, verlieren damit die Möglichkeit, eigenständig Tarifsysteme zu entwickeln und zu implementieren.

<sup>11</sup> „Die IT-Systeme in der Telekommunikationsbranche unterstützen [dagegen] die Herstellung von Basisprodukten der Telekommunikation (Anschlüsse, Ports etc.), die Bereitstellungsprozesse der Marktprodukte für die Kunden und die anschließende Abrechnung inklusive des Debitorenmanagements.“ (Külzer u.a. 2004), S.170.

Ein idealtypisches Abrechnungssystem für E-Learning-Produkte sollte über folgende Eigenschaften verfügen:

- Um eine Unabhängigkeit vom zugrundeliegenden Lern-Management-system zu erreichen sollte das Abrechnungssystem als *frei konfigurierbare Software-Komponente* konzipiert sein. Dies ermöglicht einen Austausch des Lern-Managementsystems z. B. bei technischen Neuerungen und erlaubt eine flexible Integration auch in „E-Learning-fremden“ Anwendungen.
- Das Abrechnungssystem sollte neben der Abrechnung von Preismodellen den Benutzer durch geeignete Beschreibungssprachen auch bei der Definition der Preismodelle unterstützen. (*Modellbasierte Betrachtung*)
- Neue Preismechanismen sollten schnell implementiert werden können. Eine Voraussetzung dafür ist das Vorhandensein einer *grafischen Beschreibungssprache*, mit der Preismodelle in einem „Rapid Design“ entworfen, verifiziert und diskutiert werden können. (*Schnelle Anpassbarkeit an sich ändernde Marktbedingungen*)
- Ein Abrechnungssystem sollte Schnittstellen zu gängigen Zahlungsverkehrssystemen (Electronic-Bill-Presentation & -Payment, EBPP) bereitstellen, um den Forderungseinzug bei großen Kundentstämmen zu automatisieren.

## 1.2 Zielsetzung der Arbeit

Die Arbeit leistet einen Beitrag zur Gestaltung von flexiblen Abrechnungssystemen für digitale Produkte im E-Learning. Dazu werden folgende Teilziele verfolgt:

- Es wird eine konzeptuelle, *formale Beschreibungssprache* (Syntax und Semantik) samt grafischer Notation definiert, die eine modellbasierte Beschreibung der Bepreisung von digitalen Produkten unterstützt.
- Der Beschreibungssprache wird durch einen Architektur-getriebenen *Entwurf eines Prototyps für die Definition von Preismodellen* mit anschließender Umsetzung „Leben eingehaucht“. Anwender werden so in die Lage versetzt, mit Hilfe von grafischen Modellen eigenständig Preismechanismen zu entwerfen und zu implementieren. (*Price-Modeler-Prototyp*)

- Die Einsatzfähigkeit der Beschreibungssprache und eine mögliche Umsetzung eines Abrechnungssystems im E-Learning werden mit Hilfe eines *prototypischen Lern-Managementsystems* im Sinne einer Fallstudie beschrieben.
- Durch Mapping der Sprachkonzepte mit Preismodellbestandteilen in den Anwendungsbereichen Mobilfunk und Versicherung wird in den Kapiteln 5.3.2 und 5.3.3 gezeigt, dass die Beschreibungssprache auch für die Modellierung von Tarifstrukturen in diesen Bereichen geeignet ist. (*Verallgemeinerbarkeit*)

### 1.3 Methodische Vorgehensweise

Das betrachtete Problemfeld des Pricings ist traditionell dem Marketing zuzuordnen. Dabei beschäftigt sich die Marketing-Literatur vor allem mit der Gestaltung von Differenzierungsstrategien für Preise und Produkte. Eine konkrete Handlungsanweisung, wie die beschriebenen Strategien in Softwaresysteme umzusetzen sind, bleibt jedoch aus. Im Gegensatz dazu widmet sich die Forschung in der Informatik der Konzipierung von Architekturen für Abrechnungssysteme und der Beschreibung sowie Umsetzung bestehender Abrechnungssysteme.

Im Rahmen der Arbeit wird aus Sicht der Wirtschaftsinformatik eine *Brücke zwischen einer strategischen Betrachtung des Pricings und einer reinen Fokussierung auf technische Umsetzungsmöglichkeiten* geschlagen. Dazu wurden neben dem Entwurf einer Beschreibungssprache Softwareprogramme entwickelt, die eine modellbasierte Steuerung von Tarifsystemen und eine Kopplung an industrielle Abrechnungssysteme erlauben.

Die *Definition der Modellierungssprache* (Entwurf einer Orthosprache und eines Metamodells) orientiert sich methodisch an Vorgehensweisen zur Definition von Modellierungssprachen für Führungsinformationssysteme (Holten 1999). Darüber hinaus wurde eine Mini-Programmiersprache (AccLan) als Teil der Sprache in einer Erweiterten Backus-Naur-Form (EBNF) spezifiziert.

Der Entwurfsprozess der Prototypen wurde durch die Definition eines *Vorgehensmodells* und durch die Konzipierung von Software-Architekturen strukturiert. Für die Darstellung von Prozessen wurden *Petri-Netze* (CP-Netze) verwendet, die neben der einfachen Erlernbarkeit im Gegensatz zu anderen

Prozessmodellierungssprachen über Formalismen zur Überprüfung der modellierten Prozesse verfügen.

Die Ergebnisse der Arbeit wurden mit Experten im E-Learning, im Vertrieb digitaler Produkte (Mobilfunk, Versicherung, Abrechnungssystem-Produzent) im Forschungsschwerpunkt „Internetökonomie“ und auf internationalen Konferenzen diskutiert. Erfahrungen wurden in die Arbeit integriert.

## 1.4 Gang der Untersuchung

Der Aufbau der Arbeit ist in Abbildung 1 auf Seite 8 dargestellt.

In einem ersten Block (Kapitel 2 bis 4) werden die methodischen Grundlagen beschrieben und die verwendete Begrifflichkeit erläutert.

Das zweite Kapitel analysiert die Begriffe des *digitalen Handels*, des *digitalen Produktes* und beschreibt anschließend den *Handel digitaler Produkte im E-Learning*. Neben einer Betrachtung der zentralen Bestandteile des E-Learnings aus informatischer Sicht – der Lernobjekte und Lernprozesse – werden insbesondere ökonomische Aspekte des E-Learnings beleuchtet. Hierzu gehören die Beschreibung von Preisdifferenzierungsstrategien und aktueller Geschäftsmodelle von Bildungsanbietern.

Kapitel 3 führt in die *Ablaufmodellierung mit Petri-Netzen* ein. Neben elementaren (einfachen) Petri-Netzen, die zur Illustration der grundlegenden Mechanismen und Elemente aller Petri-Netz-Klassen dienen, werden *Gefärbte Petri-Netze* (Coloured Petri Nets, CP-Netze) als eine Klasse höherer Petri-Netze eingeführt. Diese haben u.a. den Vorteil, dass sich mit ihnen die Manipulation beliebig komplexer Datentypen darstellen lässt und daher die prozessuale Logik von Software-Prototypen abbilden und verifizieren lässt, bevor diese implementiert werden. Das Kapitel bildet die Grundlage für das Verständnis der Prozessbeschreibungen mittels Gefärbter Petri-Netze (Coloured Petri-Nets) in den Kapiteln 6 und 7.

Im vierten Kapitel werden der *Modellbegriff*, *Methoden der Modellbildung* und ein *Rahmenmodell zur Informationssystem-Entwicklung* aus Sicht der Informatik betrachtet. Das Rahmenmodell dient dazu, die Entwicklung von sprachbasierten Informationssystemen in grobe Schritte (Fachkonzept, DB-Konzept und Implementierung) einzuteilen und somit eine gedankliche Strukturierung der Entwurfsprozesse für die in der Arbeit entwickelten Prototypen zu erreichen.

Durch die Konkretisierung des Rahmenmodells wurden in späteren Kapiteln Vorgehensmodelle für die Umsetzung der Prototypen entworfen. Daneben wird ein *Vorgehen zur Gestaltung von Modellierungssprachen* skizziert, das sich in Teilen im Rahemmodell wiederfindet und in Kapitel 5 bei der *Definition der Beschreibungssprache für Preismodelle – PriMoL* – angewendet wird.

Kapitel 5 bildet einen wesentlichen Teil der Arbeit. Zur Entwicklung der Modellierungssprache wurden neben dem E-Learning auch Versicherungstarife und Mobilfunktarife untersucht. Kapitel 5 umfasst neben einem konzeptuellen Überblick über die für eine Bepreisung benötigten Modelle aus informatischer Sicht (*Produkt-, Kundengruppen- und Preismodelle*) die *Sprachdefinition* (Kapitel 5.4). Das Kapitel schließt mit einer *Kritik der Beschreibungssprache*.

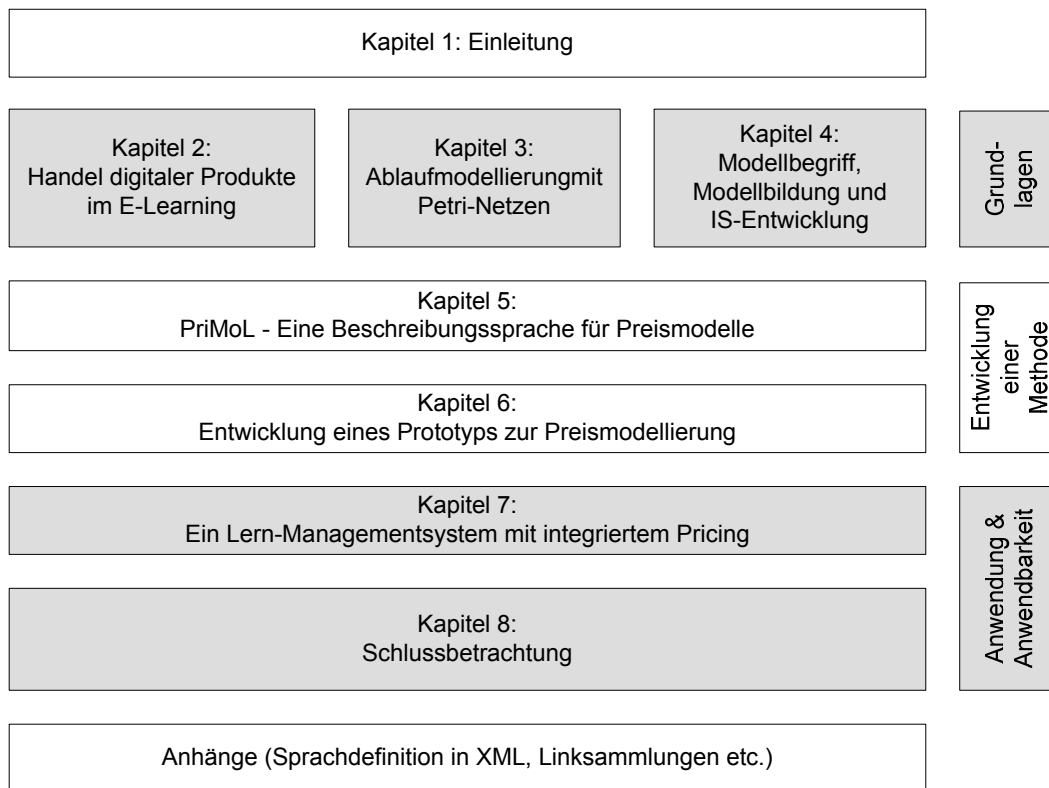
Das sechste Kapitel zerfällt in zwei Teile, in denen ein Vorgehen für die *Architektur-getriebene Entwicklung eines Software-Prototyps* zur Umsetzung der Beschreibungssprache beschrieben wird. In einem ersten Teil wird dazu eine allgemeine Informationssystem-Architektur für Abrechnungssysteme entworfen. Anhand der Informationssystem-Architektur wird der Funktionsumfang des entwickelten Software-Prototyps zur Modellierung von Preismodellen (Price-Modeler) eingegrenzt und durch die Überführung der Informationssystem-Architektur in eine Entwurfsarchitektur die Implementierung des *Java-Prototyps* beschrieben.

Der Hauptteil der Arbeit endet mit Kapitel 7. Anhand eines in PHP entwickelten *Lern-Managementsystems inklusive Laufzeitumgebung für Preismodelle* wird die Einsatzfähigkeit der Modellierungssprache bei der Abrechnung von digitalen Produkten im E-Learning demonstriert.

Die Schlussbetrachtung in Kapitel 8 fasst die *Ergebnisse der Arbeit* zusammen und gibt einen Ausblick auf *Weiterentwicklungen der vorgestellten Konzepte*.

Die Arbeit wird durch Anhänge abgerundet. Neben *Linksammlungen* zu virtuellen Studiengängen, virtuellen Universitäten, Suchmaschinen für E-Learning und Herstellern von Abrechnungssystemen, wird die *Grammatik der Beschreibungssprache PriMoL* und der integrierten Accounting-Language beschrieben. Der Anhang umfasst zudem eine umfassende Beschreibung *weiterer Aspekte der verwendeten Petri-Netz-Typen* und eine Betrachtung der *Eignung von Petri-Netzen im Software Engineering* sowie ein *Glossar wichtiger Begriffe*, die bei der

Beschreibung von Abrechnungssystemarchitekturen und Komponenten von Abrechnungssystemen in der Forschung und Praxis verwendet werden.



**Abbildung 1: Aufbau der Arbeit**

## 2 Handel digitaler Produkte im E-Learning

Im E-Learning werden digitale Produkte i. w. S. (Lernobjekte und Leistungen) an Lernende über das Internet vertrieben. Für viele E-Learning-Provider bietet es sich an, die erstellten Lerninhalte über das Internet an unterschiedliche Kundengruppen, z. B. in der Fortbildung oder an Universitätsstudierende zu vertreiben und durch geeignete Preisdifferenzierungsstrategien die angestrebten Erlöse aus E-Learning zu „maximieren“.<sup>12</sup> Die Digitalität der Produkte im E-Learning bietet – durch die sich daraus ergebende leichte Anpassbarkeit der Produkte – ideale Voraussetzungen für eine Preisdifferenzierung.

Das Ziel dieses Kapitels liegt darin, bestehende Vermarktungsstrategien für E-Learning zu sichten und eine Handlungsanweisung für die Einführung von Preismodellen als Teil der Vermarktungsstrategie / des Geschäftsmodells eines E-Learning-Anbieters aufzustellen.

In einem ersten Schritt werden dazu die grundlegenden Begriffe des elektronischen Handels (E-Commerce) beschrieben. Eine Vermarktung von E-Learning betrifft in der Regel nicht nur den in Kapitel 2.1 beschriebenen elektronischen Handel im Sinne eines „Direct E-Commerce“, sondern auch den „Indirect E-Commerce“, da vielfach E-Learning-Kurse durch konventionelle Lehrveranstaltungen flankiert werden und somit ein Teil der Leistung „offline“ erfolgt. Im Anschluss an die Darstellung des Handels, werden Klassen von digitalen Produkten voneinander abgegrenzt und deren Eigenschaften beschrieben (vgl. Kapitel 2.2-2.3). Mögliche Preisdifferenzierungsstrategien für digitale Produkte (Kapitel 2.4) werden in Kapitel 2.5 anhand eines Beispiels einer virtuellen Universität erläutert.

### 2.1 Elektronische Handelsbereiche

Der Begriff *E-Commerce* wird in der Literatur unterschiedlich definiert. Ein allgemeines Verständnis von E-Commerce als Prozess der elektronischen Abwicklung von Markttransaktionen über Online-Medien (Olbrich und Grünblatt 2001) wird z. B. durch (Lampe 1999), S. 3 detailliert. Dieser zählt neben der Abwicklung auch die Phasen der Anbahnung und Aushandlung zum

---

<sup>12</sup> Letztendlich kann nur ein möglichst hoher Erlös erzielt werden. Das Erlösmaximum kann nur unter Annahmen über den Markt, z. B. die Nutzenfunktionen der Abnehmer, angenähert werden.



E-Commerce. Unterschiede in den Auffassungen ergeben sich u. a. daraus, dass keine Einigkeit darüber herrscht, welche Technologien und welche Phasen von Markttransaktionen in die Definition des Begriffs einbezogen werden sollen. So meinen (Kalakota und Whinston 1996), dass E-Commerce einen Handel über Computernetzwerke voraussetzt. Dagegen vertreten z. B. (Adam und Yesha 1996) die Auffassung, dass E-Commerce auch dann vorliegt, wenn die Kommunikation digital erfolgt, z. B. über Telefon oder mittels Fax. Auch bezüglich der Anzahl der am Handel beteiligten Akteure herrscht keine Einigkeit. (Whinston u.a. 1997) vertreten sogar die Auffassung, dass auch unternehmensinterne Prozesse unter den Begriff fallen. In Anlehnung an (Whinston u.a. 1997) kann E-Commerce in die in Abbildung 2 dargestellten Teilbereiche zerlegt werden. Dabei ist lediglich der Handel von physischen Produkten durch physische Marktteilnehmer mittels physischer Prozesse nicht als Form des E-Commerce anzusehen. Bei vollständiger Digitalisierung der Produkte, Prozesse und Marktteilnehmer schlagen die Autoren die Verwendung des Begriffs *D-Commerce* vor.

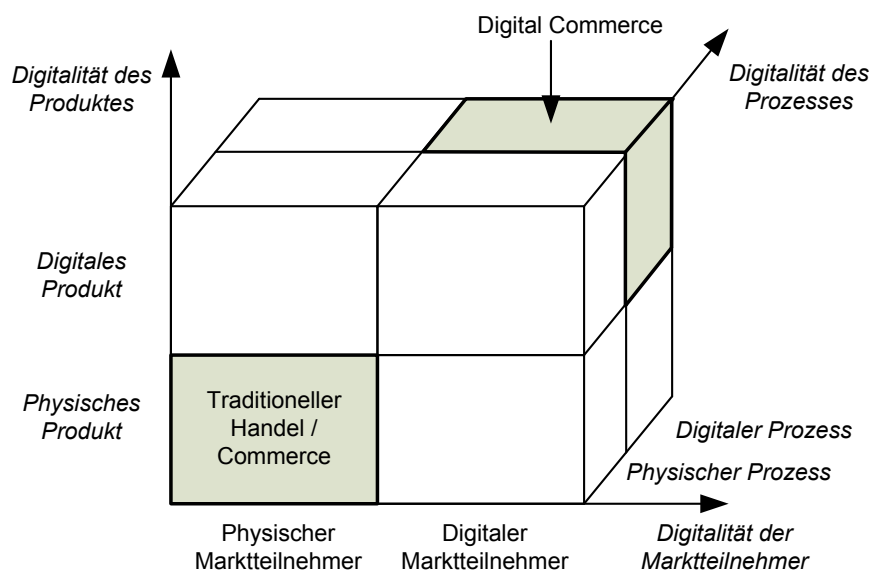


Abbildung 2: Einteilung des E-Commerce in Handelsbereiche<sup>13</sup>

Eine ähnliche Einteilung nimmt (Bar 2001) vor. Nach ihm lässt sich der E-Commerce in die in Abbildung 3 dargestellten Bereiche einteilen. *Net-aided Commerce*, *Indirect E-Commerce* und *Direct E-Commerce* lassen sich anhand der

<sup>13</sup> Quelle: modifiziert nach (Whinston u.a. 1997), S. 18.

zu Grunde liegenden Infrastruktur, der Gestaltung des Marktplatzes, der Art der Transaktionsdurchführung und des Bezahlsystems sowie der Beschaffenheit der gehandelten Güter voneinander und vom traditionellen Handel abgrenzen. Der *traditionelle Handel* (Conventional / Traditional Commerce) ist erst dann dem E-Commerce zuzurechnen, wenn zumindest der Bezahlvorgang über elektronische Netze abgewickelt wird und wird dann als *Net-aided Commerce* bezeichnet. Im *Indirect E-Commerce* treffen Angebot und Nachfrage für konventionelle Güter (C-Goods) auf einem *elektronischen Marktplatz* zusammen. Der Teilbereich des E-Commerce, in dem auch die Auslieferung des Produktes (E-Good) elektronisch erfolgt, wird als *Direct E-Commerce* bezeichnet (Bar 2001; Heffner und Mühlfeld 2001).

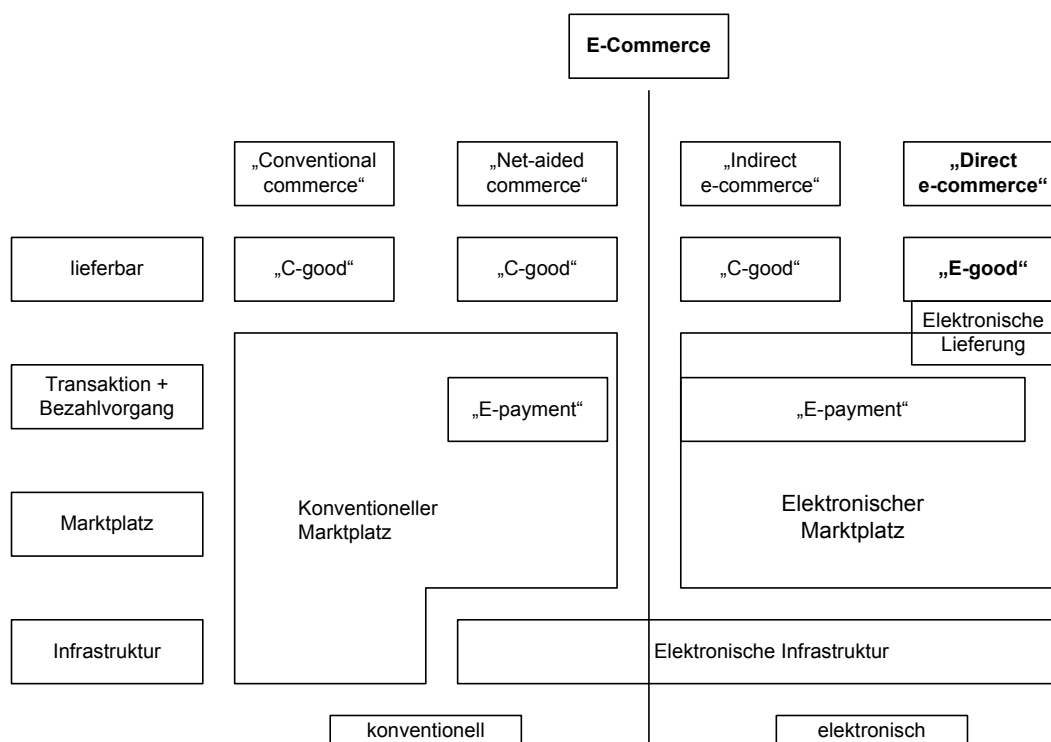


Abbildung 3: Arten des elektronischen Handels<sup>14</sup>

Die *Marktinfrastuktur* ermöglicht den Austausch von Informationen, z. B. Produkteigenschaften, Preisinformationen sowie Vertragsdokumenten zwischen den Marktakteuren. Zur *Marktinfrastuktur* zählt auch das verwendete Netzwerk, z. B. ein IP-basiertes Netzwerk wie das Internet.<sup>15</sup> Zu den *Marktakteuren* zählen

<sup>14</sup> Quelle: modifiziert nach (Bar 2001), S. 33.

<sup>15</sup> In frühen Artikeln zur Internetökonomie, z.B. (McKnight und Bailey 1997), wurde überwiegend der Internetzugang selbst als Ziel ökonomischer Untersuchungen betrachtet. „Developing more refined pricing models for the Internet that preserve the low transaction costs [...] is the common goal of studies of Internet economics“, S. 31.

Verkäufer, Käufer, Lieferanten, Zwischenhändler und andere durch den Handel eines Produktes an der Wertschöpfung beteiligte Akteure, beispielsweise Content-Ersteller für Mobilfunkanwendungen. Alle Teilbereiche des E-Commerce bauen auf einer elektronischen *Infrastruktur* wie dem Internet (WWW, E-Mail, FTP etc.) oder Mobilfunknetzen auf (Bar 2001), S. 32. Die Verständigung zwischen den beteiligten Wertschöpfungs- und Transaktionspartnern findet auf einem *Marktplatz* statt. Auf dem Marktplatz werden Angebots- und Nachfragepläne abgestimmt, Transaktionen vorbereitet und der Vertrag geschlossen, der die Übertragung der Verfügungsrechte zwischen den Marktteilnehmern vereinbart und rechtlich absichert. Der *Transaktionsprozess* und der *Bezahlvorgang* sind im Bereich des Direct und Indirect E-Commerce elektronische Prozesse, die über das Internet durchgeführt werden. Eine Transaktion beinhaltet hierbei den Versand sowie die Abwicklung von Aufträgen. Die Art des Handels wird durch die Güterart beeinflusst; nur E-Goods lassen sich aufgrund ihrer Digitalität über elektronische Netze verbreiten (E-Delivery). Indirect und Direct E-Commerce unterscheiden sich nur hinsichtlich ihrer Auslieferungsmöglichkeiten. *C-Goods*, die in den Bereichen des *Conventional Commerce*, *Net-aided Commerce* und *Indirect E-Commerce* die charakteristischen Produktarten bilden, können nur physisch ausgeliefert werden. Die *E-Goods* haben für die Internetökonomie besondere Bedeutung, da die in Kapitel 2.3 erwähnten ökonomischen Wirkungen mit den besonderen Eigenschaften dieser Produkte verbunden sind (Zerdick u.a. 2000). Insbesondere die Kostenstruktur digitaler Produkte bildet eine Voraussetzung dafür, dass eine Veränderung der Preissetzung im D-Commerce möglich ist.

## **2.2 Ein Framework zur Klassifizierung digitaler Produkte**

In diesem Kapitel werden in einem ersten Schritt Definitionen grundlegender Begriffe (digitales Produkt, digitale Dienstleistung usw.) aus unterschiedlichen Literaturquellen dargestellt. Die Aufführung der Definitionsversuche verfolgt den Zweck, eine Abgrenzung zwischen gängigen Begriffen zu erreichen und somit eine Basis für die nachfolgenden Unterkapitel zu schaffen. In den Unterkapiteln 2.2.3 und 2.2.4 wird anschließend das Framework zur Klassifizierung digitaler Produkte nach Hui und Chau sowie mögliche Erweiterungen dieses Frameworks beschrieben. Das Framework und dessen Erweiterungen führen wesentliche

Merkmale digitaler Produkte (Trialability, Granularität usw.) auf und liefern darauf basierende Hinweise, welche strategischen Aspekte bei der Bepreisung für unterschiedliche Arten digitaler Produkte zu beachten sind.

### 2.2.1 Produkte, Rechtstitel und Dienstleistungen

*Digitale Produkte* (i. w. S.) können nach (Whinston u.a. 1997; Loebbecke 1999) sowohl Produkte (i. e. S.), z. B. im Sinne eines auslieferbaren Software-Produkts, als auch elektronische Dienstleistungen (Services), z. B. im Rahmen eines Application-Service-Providing (ASP), umfassen. Voraussetzung für die Herstellung eines digitalen Produktes ist dessen Digitalisierbarkeit.

*Digitale Produkte i. e. S.:*

Zu den digitalen Produkten i. e. S. gehören nach (Loebbecke 1999) alle Güter, die sich online ausliefern lassen (Online-Delivered Content) und aus digitalkodierten Zeichenfolgen zusammengesetzt sind (Stickel u.a. 1997). Eine Definition von digitalen Gütern liefert (Stelzer 2000)<sup>16</sup>:

*„Unter digitalen Gütern [digitalen Produkten, d. Verf.] versteht man immaterielle Mittel zur Bedürfnisbefriedigung, die sich mit Hilfe von Informationssystemen entwickeln, vertreiben oder anwenden lassen. Es sind Produkte oder Dienstleistungen, die in Form von Binärdaten dargestellt, übertragen und verarbeitet werden können.“*

Diese sehr weit gefasste Definition ist zur Klassifikation digitaler Produkte nicht ausreichend. Nach (Whinston u.a. 1997), S. 62, 87 kann unter einem digitalen Gut alles verstanden werden, was über elektronische Netze versendet werden kann, digitalisierbar ist oder bereits in digitaler Form vorliegt.

*Digitale Rechtstitel und Zertifikate:*

Zu den digitalen Produkten i. e. S. gehören somit auch *Rechtstitel und Zertifikate* (s. Abbildung 4). Diese umfassen bspw. Lizenzen und Coupons. Im Falle von Coupons liegt weder ein Produkt vor, das direkt genutzt werden kann, noch handelt es sich um eine Dienstleistung. Ein *Coupon* ist ein Recht, das z. B. den vergünstigten Erwerb eines Produktes oder eine Zugabe durch den Produzenten im Erwerbsfall durch den Kunden verbrieft. Zu den digitalen Zertifikaten gehören

---

<sup>16</sup> Analog: (Varian 1998), S. 3.

weiterhin Sicherheitszertifikate, die zur Absicherung von Internetverbindungen benötigt werden, sowie digitales Geld (E-Cash).

*Digitale Dienstleistungen:*

Die Zuordnung der gehandelten Produkte (i. w. S.) zu Produkten (i. e. S.) und Dienstleistungen ist oft schwierig. Software, die als *Packaged Software* zum Download bereitgestellt wird, weist eher einen *Produktcharakter* auf, obwohl die Software selbst Dienste erbringt (Dokumente ausdrucken, E-Mails empfangen etc.). Dagegen ist Software, die im Rahmen eines Application-Service-Providing über das Internet bereitgestellt wird, eher als Dienstleistung zu betrachten.

Die folgenden Ausführungen zur Beschreibung des Begriffs „Dienstleistung“ orientieren sich an (Bodendorf 1999), S. 2 ff in Anlehnung an (Korschenewski 2005). Eine Dienstleistung verfügt über folgende Charakteristika:

- 1) *Immaterieller Charakter*: Betrachtet man das Ergebnis oder den Nutzen, aus einer Dienstleistung, so ist dieses / dieser i. d. R. nicht „unmittelbar fassbar.“
- 2) *Integration eines externen Faktors*: „Am Dienstleistungsprozess ist der Kunde oder ein Objekt aus seinem Besitz beteiligt. Dieser so genannte „externe Faktor“ ist als weiteres zentrales Dienstleistungsmerkmal weitgehend unstrittig und führt zu einer gewissen, allerdings auch nicht immer vorhandenen Auftragsindividualität. Der Kunde gestaltet dabei die Dienstleistung mit oder sie wird an seiner Person bzw. an einem ihm zugeordneten Objekt vollzogen.“ (Korschenewski 2005), S. 21
- 3) *Bedarfsdeckung durch Leistung*: Ein Nachfrager deckt seinen Bedarf durch das Konsumieren der Leistung. Dabei besteht der Nutzen der Dienstleistung darin, dass bei der Bedarfsdeckung der externe Faktor (s. o.) integriert werden muss. Sowohl die Erzeugung als auch die Konsumtion der Leistung haben einen Prozesscharakter.  
Da eine Dienstleistung im Vergleich zu Produkten nur schwer veranschaulicht werden kann, ist die Darstellung des Nutzens aus der Dienstleistung gegenüber einem Kunden schwierig. Der Vertrieb von Dienstleistungen geht daher i. d. R. mit einem hohen Beratungsbedarf seitens des Kunden einher.
- 4) *Hohe Verderblichkeit*: Dienstleistungen sind nicht lagerfähig. Der Produktionsprozess einer Dienstleistung ist i. d. R. mehrstufig. „Die

Beschaffung interner Produktionsfaktoren und die Produktionsprozesse finden teilweise vor [Vorkombination, d. Verf.] und teilweise nach dem Absatz [Endkombination, d. Verf.] statt.“ (Korschenewski 2005), S. 22.

Eine *digitale Dienstleistung* zeichnet sich dadurch aus, dass diese über elektronische Kanäle erbracht wird und meist durch die Aktion eines Abnehmers eine zugrundeliegende digitale Transaktion getriggert und automatisch durchgeführt wird. Zu den digitalen Dienstleistungen zählen u. a. von Providern im World-wide Web bereitgestellte Tarifvergleiche für Reisen und Versicherungen sowie Produktpreisvergleiche. Die vorgenannten Dienstleistungen wären in Abbildung 4 auf S. 16 z. B. dem aufgeführten Bereich „E-Commerce-Dienstleistung“ zuzurechnen. Nach der beabsichtigten Nutzung können die o. g. Dienstleistungen auch dem Bereich „Unterhaltung / Freizeit / Kommunikation“ zugerechnet werden.

Unter den Begriff der „digitalen Dienstleistungen“ werden auch „Infrastrukturdienste“ gefasst. Hierzu zählen bspw. die Bereitstellung von Internetzugängen (DSL, ISDN etc.) und die Weiterleitung von Datenpaketen über das Internet. Infrastrukturdienste stellen als „Low-Level“-Dienste die notwendige Plattform bereit, um einen Handel über das Internet überhaupt zu ermöglichen. Im Gegensatz zu den im vorigen Abschnitt genannten Diensten werden Infrastrukturdienste einmal eingerichtet und beliebig oft automatisiert ausgeführt. Infrastrukturdienstleistungen sind weitgehend standardisiert und werden oft mit Hilfe zeitbasierter oder volumenbasierter Tarife abgerechnet. Für eine zeitbasierte Bepreisung z. B. der Nutzung von DSL als Infrastrukturdienst spricht, dass diese keine Kenntnisse über die erbrachte Dienstleistung (Anzahl der übertragenen Pakete usw.) voraussetzt.

Neben digitalen Dienstleistungen, die nur durch den Anwender bzw. Abnehmer getriggert werden und voll automatisierten Dienstleistungen bei Infrastrukturdiensten, werden in der Arbeit auch solche Dienstleistungen als „digitale“ Dienstleistungen betrachtet, bei denen ein menschlicher Akteur bei der Leistungserstellung und / oder der Leistungsabnahme „mitarbeitet.“ Hierzu zählen auch E-Learning-Prozesse, die eine Kommunikation zwischen Lehrenden und Lernenden ermöglichen. Digitale Dienstleistungen, die durch eine intensive, asynchrone Kommunikation zwischen Produzenten und Abnehmern

gekennzeichnet sind, können durch spezifische Systeme unterstützt werden (z.B. Workflow-Managementsysteme und Groupware-Systeme)

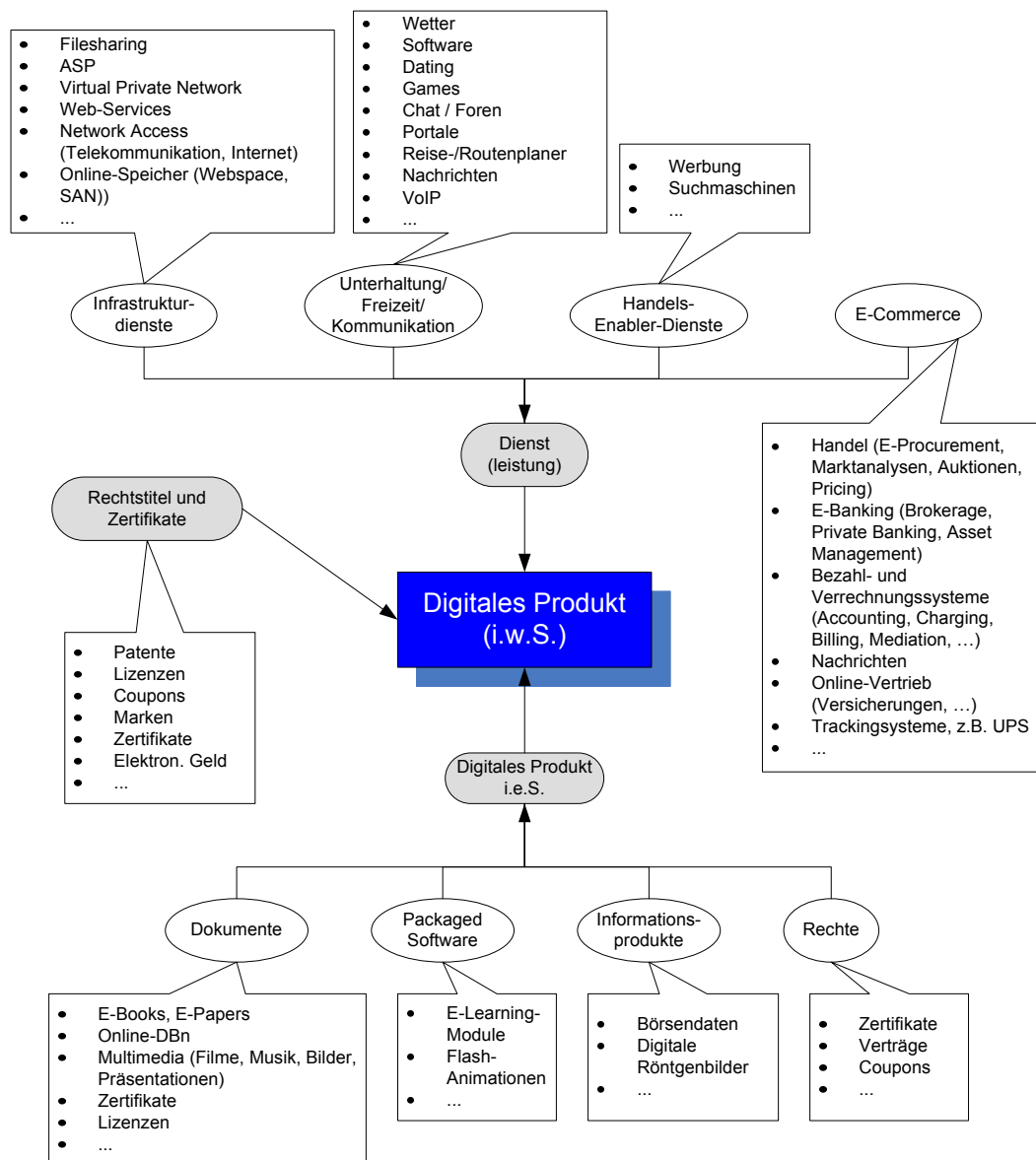


Abbildung 4: Taxonomie digitaler Produkte

## 2.2.2 Digitale Informations- und Netzeffektgüter

Digitale Produkte i. e. S. können nach (Friedrich 2003) in digitale Netzeffektgüter und digitale Informationsgüter unterteilt werden.

*Digitale Netzeffektgüter* sind den nicht digitalen Netzeffektgütern (Faxgeräten, Videorekorder etc.) ähnlich, da sie ebenfalls Netzeffekte aufweisen (Friedrich 2003), S. 3. Zu den digitalen Netzeffektgütern gehören bspw. Standardsoftware-Produkte. Der Begriff *Netzeffekt* leitet sich bei digitalen Netzeffektgütern daraus

ab, dass der Nutzen eines Netzwerkteilnehmers bspw. aus der Verwendung einer Standardsoftware, z. B. einem Textverarbeitungsprogramm, mit der Größe des Netzes der Teilnehmer, der so genannten installierten Basis, zunimmt. Dies liegt daran, dass mit zunehmender Teilnehmerzahl z. B. der Austausch von Dateien, die mit der Standardsoftware erstellt wurden, leichter möglich ist, da davon auszugehen ist, dass der Empfänger einer Datei diese auch lesen und verarbeiten kann. Zusätzlich zum eigentlichen Netzeffektgut entstehen bei einer großen installierten Basis komplementäre Dienstleistungen und Produkte, wie z. B. Add-Ins, Hilfeforen und Schulungen, die den Nutzen für den Anwender zusätzlich erhöhen. Es liegen dann (positive) *Netzeffekte* vor. Netzeffekte sind nachfragerseitige Skaleneffekte (Shapiro und Varian 1999), S. 179-181. Neben dem *originären Nutzen* eines Netzeffektgutes, der unabhängig von der gesamten Anzahl der Netzwerkteilnehmer und i. d. R. niedrig ist, ergibt sich der Nutzen für den Konsumenten vor allem durch den *Synchronisationswert* (ähnlich dem derivativen Nutzen), der durch die Interaktion mit anderen Teilnehmern des gleichen Netzwerkes bzw. durch kompatible Produkte entsteht (Friedrich 2003). Mit steigender Anzahl an Netzwerkteilnehmern oder kompatiblen Gütern steigt der Synchronisationswert für den Netzwerkteilnehmer. Im Zusammenhang mit Netzeffekten tritt oft ein Phänomen auf, das als *kritische Masse* eines Produktes bezeichnet wird. Die kritische Masse bezeichnet die Anzahl der Nutzer, die eine hinreichende Attraktivität für potenzielle Nutzer derart sicherstellt, dass ein sich selbst tragender Penetrationsprozess des Marktes in Gang gesetzt wird (Clement u.a. 2001), S. 102-104.

Bei *digitalen Informationsgütern* bilden im Unterschied zu Netzeffektgütern Informationen die eigentlichen Produkte, die für den Nachfrager einen Nutzen erzeugen.<sup>17</sup> Zeitungen und Fachbücher sind Beispiele für Informationsgüter, die sich zum Vertrieb als digitale Informationsprodukte, z. B. in Form eines E-Papers und E-Books, eignen (Whinston u.a. 1997). Digitale Informationsgüter zeichnen sich im Gegensatz zu Netzeffektgütern durch einen hohen originären Nutzen aus. Informationsgüter können *Zweck- und Unterhaltungsinformationen* kapseln (Hass 2002). Zu Informationsgütern, die Zweckinformationen kapseln, gehören Börsenkurse und Marktdaten. Unterhaltungsgüter können im Gegensatz zu



Zweckinformationen als Konsumgüter angesehen werden, die zur Produktion von Unterhaltung dienen und somit aufgrund emotionaler Kaufentscheidungen erworben werden. Sie werden meist mehrfach genutzt. Dagegen dienen Zweckinformationen der Findung von Entscheidungen (Investitionscharakter) und werden i. d. R. nur einmalig genutzt. Zweckinformationen unterliegen einer hohen Verderblichkeit.

### 2.2.3 Klassifizierung digitaler Produkte nach Hui und Chau<sup>18</sup>

(Hui und Chau 2002) schlagen ein Framework vor, das die Klassifizierung digitaler Produkte und Dienstleistungen unterstützt und so einen Grundstein für die Entwicklung von Plattformen zur Erstellung und zum Download von Produkten und Dienstleistungen legen soll. Dabei soll es Entscheidern helfen, durch Zuordnung von Produkteigenschaften zu Produktkategorien (Matching) gleichartige Produkte zu identifizieren, um so eine Wiederverwendung entwickelter Marketing-Strategien – z. B. die Wahl eines Erlösmodells für eine bestimmte Produktkategorie – zu ermöglichen. Darüber hinaus soll durch das Framework die Identifizierung kritischer Produkteigenschaften und -dimensionen, die eine „nähere Betrachtung“ durch den Produkt-Designer erfordern, erleichtert werden.

Die in der Abbildung 5 aufgelisteten Produktkategorien lassen sich wie folgt charakterisieren:

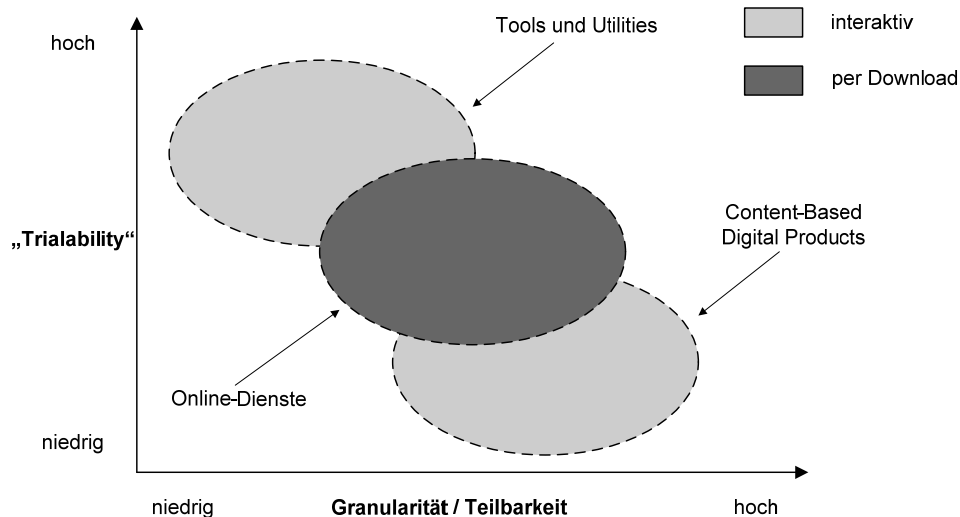
- *Tools und Utilities* sind Softwareprogramme, die Anwendern entweder helfen, bestimmte Funktionen auszuführen, z. B. Virenschannen oder die Erzeugung von PDF-Dateien, oder zusätzliche Fähigkeiten für bestehende Software bieten, z. B. das Abspielen von Musik im Internet-Browser.
- Der Wert von *Content-based Digital Products* liegt in den in ihnen gekapselten Informationen. Es handelt sich dabei um die in Absatz 2.2.2 beschriebenen Informationsgüter (Zeitschriften, Nachrichten, Musikdateien, Videos u. a.).

---

<sup>17</sup> Die in Form eines „Informationsgutes“ übertragenen Daten werden erst durch die Deutung des Anwenders zu Informationen. (Ferstl und Sinz 1993).

<sup>18</sup> Quelle: (Hui und Chau 2002).

- Als dritte Kategorie führen sie *Online-Dienste* auf, die einen Zugang zu „nützlichen“ Ressourcen bieten. Hierzu gehören u. a. Internet-Telefonie und Groupware-Dienste.



**Abbildung 5: Klassifizierung digitaler Produkte anhand der Produktkategorie und -eigenschaften<sup>19</sup>**

Als Klassifizierungsschema für die Produktkategorien werden Eigenschaften aufgeführt, die den Produkten „angeboren“ sind und somit nicht oder nur schwer geändert werden können. Zum einen unterscheiden sich die Produktkategorien hinsichtlich ihres *Übertragungsmodus* (Delivery Mode). Während Online-Dienste meist interaktiv angeboten werden, können Tools sowie Content per Download zur Verfügung gestellt werden. Produkte, die über eine hohe *Teilbarkeit / Granularität* (Granularity) verfügen, erlauben eine vertikale Differenzierung. So können Buchanbieter im Internet auch einzelne Kapitel ihrer E-Books zum Verkauf anbieten. Ebenso kann es für den Verkauf von Marktdaten (Börsenkurse etc.) interessant sein, nach deren Aktualität zu differenzieren und unterschiedliche Bündel zu vertreiben (Langhammer 2004). Als letzte Kategorie wird die *Trialability* (Möglichkeit des Testens durch den Anwender) eines Produktes aufgeführt. Informationsgüter wie Bücher besitzen eine geringe Trialability. I. d. R. können Bücher nicht „ausprobiert“ werden. Eine Möglichkeit, die Trialability zu erhöhen, besteht darin, eine Preview-Funktion bereitzustellen, in der ausgewählte Kapitel sowie das Inhaltsverzeichnis von Büchern zur Online-

<sup>19</sup> Quelle: modifiziert übernommen aus (Hui und Chau 2002).

Betrachtung zugelassen werden. Bei Software bietet sich die kostenfreie Nutzung von Testlizenzen zur Gewinnung von Kunden an.

Produkteigenschaft/ Produktkategorie	Transfer- modus	Granularität	Trialability
Tools und Utilities	per Download	niedrig *	hoch*
Online Services	interaktiv	mittel*	mittel*
Digital Content	per Download	hoch*	niedrig*

\* *relativ*

**Tabelle 1: Klassifizierungsschema für digitale Produkte<sup>20</sup>**

In Tabelle 1 werden die Eigenschaften der Produktkategorien abschließend gegenübergestellt. Die Aussagen sind als Tendenzaussagen zu bewerten, da sich innerhalb der Produktkategorien Verschiebungen ergeben können. So kann die Trialability von Digital Content wie zuvor beschrieben erhöht werden, indem für E-Books Probekapitel zum Download angeboten werden. Dabei eignen sich Produkte wie statistische Softwarepakete oder Internet-Telefonie tendenziell eher zu einer verbrauchsabhängigen oder zeitabhängigen Abrechnung (Pay-per-Period, Leasing, Licensing) als bspw. der Download von Content (Büchern etc.)

#### 2.2.4 Erweiterungen des Frameworks

Das Framework von (Hui und Chau 2002) wurde von (Larsson und Svensson 2004) zur Klassifizierung von Online-Newspaper-Content schwedischer Zeitungen im Rahmen einer empirischen Erhebung verwendet und erweitert. Als zusätzliche Klassifizierungsmerkmale wurden

- (1) die genaue *inhaltliche Beschreibung der Produktkategorien* (Content) und
- (2) die *Anpassbarkeit* (Customization) an die Kundenbedürfnisse eingeführt.

Auch die Produktkategorien selbst wurden verfeinert. Das angepasste Framework von (Larsson und Svensson 2004) ist in Tabelle 2 abgebildet. Die ursprüngliche Kategorie Tools und Utilities wurde z. B. um den Bereich der Mobile-Utilities erweitert.

---

<sup>20</sup> Quelle: modifiziert übernommen aus (Hui und Chau 2002).

Einteilung nach (Hui und Chau 2002)				Empirische Befunde		
Eigenschaft/ Kategorie	Delive- ry	Granu- larity	Trial- ability	Ähnliche Unterka- te- gorien	Zusätzliche Attribute	
					Inhalt	Custo- misa- tion
Tools und Utilities	per Down- load	niedrig *	hoch*	Mobile Utilities	z. B. Spiele, Klingeltöne, Hintergründe	nein
				Tools und Utilities	z. B. Acrobat Reader, Antivirus- software	nein
Online Services	inter- aktiv	mittel*	mittel*	Interpersonal Online Services	z. B. Online Hilfe, Astrologie	ja
				Online community services	z. B. Dating Commun., Spielge- meinschaften	ja
				Mediating Service	z. B. Travel Mediating, Used Car Mediating, Dating	ja
Digital Content	per Down- load	hoch*	niedrig*	Redaktionell er Inhalt	z. B. online Nach- richten, Reisetipps	nein
				Veränder- barer Informations- inhalt	z. B. Wetterdienst, TV-Programm, Sportergebnisse, Börseninformatio- nen, Events, Be- richte aus der Wirt- schaft	ja

**Tabelle 2: Framework zur Klassifizierung digitaler Produkte<sup>21</sup>**

Im Gegensatz zur Nennung allgemeiner ökonomischer Eigenschaften wie der Zuordnung von Gütern zu den Kategorien „Netzeffektgut“ vs. „Informationsgut“, liefern (Hui und Chau 2002) und (Larsson und Svensson 2004) eine erste Ansatzpunkt, wie digitale Güter bepreist werden können. Wesentlich für die Bepreisung von Produkten sind deren Attribute:

*„A key component in the process of formulating such [selling] strategies is to understand the implications behind the product attributes. This requires a good understanding of the nature and characteristics of different digital products.“ (Hui und Chau 2002), S. 73.*

<sup>21</sup> Quelle: modifiziert nach (Larsson und Svensson 2004) mit eigenen Erweiterungen.

## 2.3 Eigenschaften digitaler Produkte

Neben der Granularität und Trialability werden in der Literatur weitere Eigenschaften digitaler Produkte hervorgehoben, die im Folgenden kurz beschrieben werden.

### 2.3.1 Veränderbarkeit

Durch den Austausch von Modulen einer Software können z. B. neue Varianten des digitalen Produkts „Software“ erstellt werden (Whinston u.a. 1997), S. 69. Die Veränderbarkeit digitaler Produkte stellt neue Herausforderungen an die Verwaltung der Produkte in Repositorien. Einen Aufbau von Repositorien für die Verwaltung von CAD-Daten beschreibt z. B. (Katz 1990).

### 2.3.2 Reproduzierbarkeit

Der Aufwand und die Kosten der Vervielfältigung sind bei physischen Gütern wesentlich größer als bei digitalen Gütern (Whinston u.a. 1997), S. 69. Beim Download von Software veranlasst der Konsument sogar eigenständig die Reproduktion und wird zum „*Prosumenten*.“ Andererseits ist die einfache Reproduzierbarkeit digitaler Güter für die Anbieter nachteilig, wenn diese zur „[illegalen] Verbreitung“ (Albers 2001), S. 84 genutzt wird. Ohne die Existenz von Raubkopien ist aufgrund der Kostenstruktur digitaler Produkte eine im Gegensatz zu physischen Gütern verstärkte Realisierung von Skalenerträgen<sup>22</sup> möglich.

### 2.3.3 Kostenstruktur

Die Kosten für die Produktion digitaler Güter sind durch einen hohen Aufwand für die Erzeugung der ersten Einheit und durch sehr geringe Kosten für jede weitere Einheit (Reproduktion) gekennzeichnet (Skiera und Spann 2002), S. 271; (Shapiro und Varian 1999), S. 3, 20; (Picot und Heger 2003b), S. 20. Zwischen dem Original und der Kopie bestehen dabei keine Unterschiede. Auch niedrige Preise für ein digitales Produkt können einen positiven Deckungsbeitrag erzielen, da der Stückpreis die variablen Kosten<sup>23</sup> i. d. R. übersteigt. Für die Deckung hoher Anfangsinvestitionen bei der Produktion digitaler Produkte sind entweder bei

---

<sup>22</sup> Nach (Feess 1997), S. 774 ist ein Skalenertrag „[d]ie durch den infinitesimalen Mehreinsatz aller Faktoren hervorgerufene Produktionssteigerung.“

<sup>23</sup> Die variablen Kosten der Herstellung eines digitalen Produkts gehen gegen Null.

geringen Stückpreisen ein hohes Absatzvolumen oder hohe Einstiegspreise erforderlich. Bei digitalen Netzeffektgütern können durch die niedrigen variablen Kosten sowohl auf der Angebotsseite als auch auf der Nachfrageseite Skalenerträge realisiert werden. Bei nicht digitalen Netzwerkeffektgütern gibt es nur nachfrageseitige Skalenerträge (Shapiro und Varian 1999), S. 179.

#### **2.3.4 Nicht-Abnutzbarkeit und Kapazitätsbeschränkung**

Digitale Produkte unterliegen weder einer Abnutzung noch Kapazitätsbeschränkung im Sinne knapper Produktionsressourcen<sup>24</sup> (Shapiro und Varian 1999). Dies ist in der einfachen Reproduzierbarkeit dieser Güter begründet. Da sie nach (Shapiro und Varian 1999) darüber hinaus nicht verderblich sind, besteht für Anbieter auch keine Gefahr von Gewinnverlusten durch Lagerung. Die *Nicht-Verderblichkeit / Nicht-Abnutzbarkeit* muss bei vielen digitalen Produkten angezweifelt werden, da bspw. der Aktualitätsbezug von Informationsgütern aus dem Nachrichtenbereich (Nachrichtmeldungen, Börsenkurse etc.) oft preisbestimmend ist. Informationsgüter aus dem Unterhaltungs- und Bildungssektor (E-Books, Lehrvideos etc.) sind bei der Bepreisung weniger stark an ihre Aktualität gebunden, es sei denn sie unterliegen modischen Schwankungen wie z. B. Klingeltöne. Bei digitalen Netzeffektgütern kann eine *künstliche Alterung* durch Fortentwicklung der Programme erzeugt werden. Alte Versionen einer Software werden oft sogar als Freeware zum Download bereitgestellt. Vorteile für den Produzenten bestehen darin, sich mit der Freeware einen potenziellen Kundenkreis zu erschließen. Dies erleichtert bei Erfahrungsgütern die Kaufentscheidung.

#### **2.3.5 Lager- und Transportkosten**

Ein weiteres Merkmal digitaler Güter ist, dass Lager- und Transportkosten im Vergleich zu physischen Gütern entfallen (Picot und Heger 2003a). Die Übertragung bzw. die Auslieferung digitaler Güter an die Konsumenten als Bestandteil der Lieferkette ist zudem einfacher als bei physischen Waren. Für die

---

<sup>24</sup> Kapazitätsbeschränkungen bestehen bei der Produktion der ersten Einheit. Ist diese produziert worden, so bestehen keine Beschränkungen, solange der Prozess der Erzeugung von Kopien automatisiert wird und sich die Bereitstellung des Produktes/Dienstes nicht durch eine Reservierung von Speichermedien oder Prozessorleistung auszeichnet. Bei der individuellen Bereitstellung von ERP-Lösungen über ASP sind solche Kapazitätsbeschränkungen erkennbar. Die Kapazitätsbetrachtung fokussiert also eher auf Standardprodukte.

Verteilung digitaler Produkte können digitale Netze wie das Internet genutzt werden. Voraussetzung für eine Online-Auslieferung ist ein entsprechendes Ausgabemedium bei den Endverbrauchern. Im Gegensatz zu einer physischen Lieferung verkürzt sich bei einer Distribution durch den Produzenten über das Internet die Lieferkette, so gen. *Disintermediation* (Benninger und Grandjot 2001), S. 60 und Transaktionskosten sinken (Conrady 2002). Nachteilig sind nach (Albers 2001), S. 83 für den Kunden *Downloadgebühren* für den Bezug des digitalen Produkts und *Suchkosten*, die bei der Suche nach einem geeigneten Produkt im Internet entstehen. Diese Kosten fallen jedoch auch bei physischen Gütern in Form von Anfahrtkosten oder Frachtkosten an. Das Internet kann bei digitalen Gütern als zusätzlicher Distributionskanal gegenüber dem Offline-Vertrieb betrachtet werden bzw. den Offline-Vertrieb vollständig ersetzen.

### **2.3.6 Produkte als Eigenschaftsbündel und Produktlinien**

Neben einer ganzheitlichen Betrachtung von Produkten als Verkaufsobjekte können diese auch als *Eigenschaftsbündel* angesehen werden. (Brockhoff 1999; Albers und Herrmann 2002). (Brockhoff 1999) definiert ein Produkt als:

*„... eine im Hinblick auf eine erwartete Bedürfnisbefriedigung beim bekannten oder unbekanntem Verwender von einem Anbieter gebündelte Menge von Eigenschaften, die zum Gegenstand eines Tauschs werden soll, um mit der im Tausch erlangten Gegenleistung zur Erfüllung der Anbieterziele beizutragen.“*

Dabei wird der Begriff der Eigenschaft von Brockhoff weit gefasst. Neben *akquisitorischen Eigenschaften*, die vom Verwender wahrgenommen werden, sind auch technische oder aus Sicht der Produzenten „objektive“ *Eigenschaften* beobachtbar und messbar. In der Literatur werden weitere Klassifizierungen von Eigenschaften vorgenommen. So lassen sich nach (Crawford und Di Benedetto 2000) Eigenschaften in *Features*, *Funktionen* und *Benefits* unterteilen. Dabei entsprechen *Benefits* den wahrgenommenen Eigenschaften, wogegen *Funktionen* und *Features* eher den objektiven Eigenschaften zuzurechnen sind. Produktalternativen/-varianten können bei einer Beschreibung entlang ihrer Eigenschaften in einem *n-dimensionalen Eigenschaftsraum* dargestellt werden.

Dabei werden in den Dimensionen des Eigenschaftsraumes kontinuierliche und diskontinuierliche (diskrete) Produkteigenschaften abgetragen.

Die „Auffächerung“ der Eigenschaften eines Produkts kann als Basis für Preisverhandlungen genutzt werden. Auch der Preis selbst kann als Eigenschaft des Produktes angesehen werden:

*„Verhandlungsprozesse können über verschiedene Produkteigenschaften ermöglicht oder auf ein einzelnes Attribut wie den Preis beschränkt werden.“ (Weinhardt u.a. 2003)*

Die Beschreibung von Produkten als Eigenschaftsbündel findet auch in der Beschreibung von Software-Produktfamilien und im E-Learning Anwendung (Halmans und Pohl 2002; Oberweis u.a. 2005). In der Software-Entwicklung werden meist Produktlinien entwickelt,

*„die einen gemeinsamen Kern von Merkmalen [...] und die eine Reihe von Eigenschaften unterschiedliche[r] Ausprägungen besitzen und somit in diesen Eigenschaften variieren.“<sup>25</sup>*

Das zentrale Konzept der Produktfamilienentwicklung liegt darin, Variabilität in der Gestaltung eines Produktes ausgehend von einem Produktfamilienkern zu gewährleisten (Halmans und Pohl 2002; Obbink und Pohl 2005). Die Merkmale in der Produktlinienentwicklung werden dabei i. d. R. auf die Bedürfnisse unterschiedlicher Marktsegmente abgestimmt.<sup>26</sup>

### **2.3.7 Preisdeterminanten als Basis der Preisdifferenzierung**

Preisdeterminanten umfassen spezifische Eigenschaften eines Produktes bzw. einer Produktgruppe (Kosten etc.), die zur Bepreisung durch den Verkäufer eines Produktes genutzt werden.

In Abbildung 6 ist eine Auswahl dieser „Preisdeterminanten“ aufgeführt. Neben der Abnahmemenge eines Produktes sind zeitliche, räumliche und nutzerspezifische Aspekte für die Preisfestlegung wichtig. Unter dem Begriff „Lokation“ können neben der geografischen Verortung eines Nutzers, dessen rechtliche Rahmenbedingungen, z. B. die Höhe der anzuwendenden Mehrwertsteuer, betrachtet werden. Die Bepreisung hängt bei Diensten im

---

<sup>25</sup> <http://www-lufgi3.informatik.rwth-aachen.de/WORKSHOPS/ak-tools/>

<sup>26</sup> <http://www.sei.cmu.edu/productlines/index.html>



Telekommunikationsbereich, z. B. bei der Abrechnung von Telefonaten, überwiegend von der Lokation des Dienstenutzers bzw. des Gesprächsempfängers ab. Neben den produktgebundenen, nicht veränderlichen Eigenschaften wie Qualität und Quantität, Kompatibilität mit anderen Produkten und der Möglichkeit, dieses zu personalisieren, ist insbesondere die Art des Nutzers preisbestimmend. Preise variieren i. d. R. nach der Nutzergruppe, nach der Anzahl der Abnehmer, die sich zusammenschließen, um ein Produkt zu kaufen, nach der Art der Geschäftspartner (natürliche vs. juristische Person) usw. Die Arten der Preisdifferenzierung werden in Abschnitt 2.4 näher beschrieben. Preisdeterminanten bilden das Vehikel, auf dem eine Preisdifferenzierung aufsetzen kann.

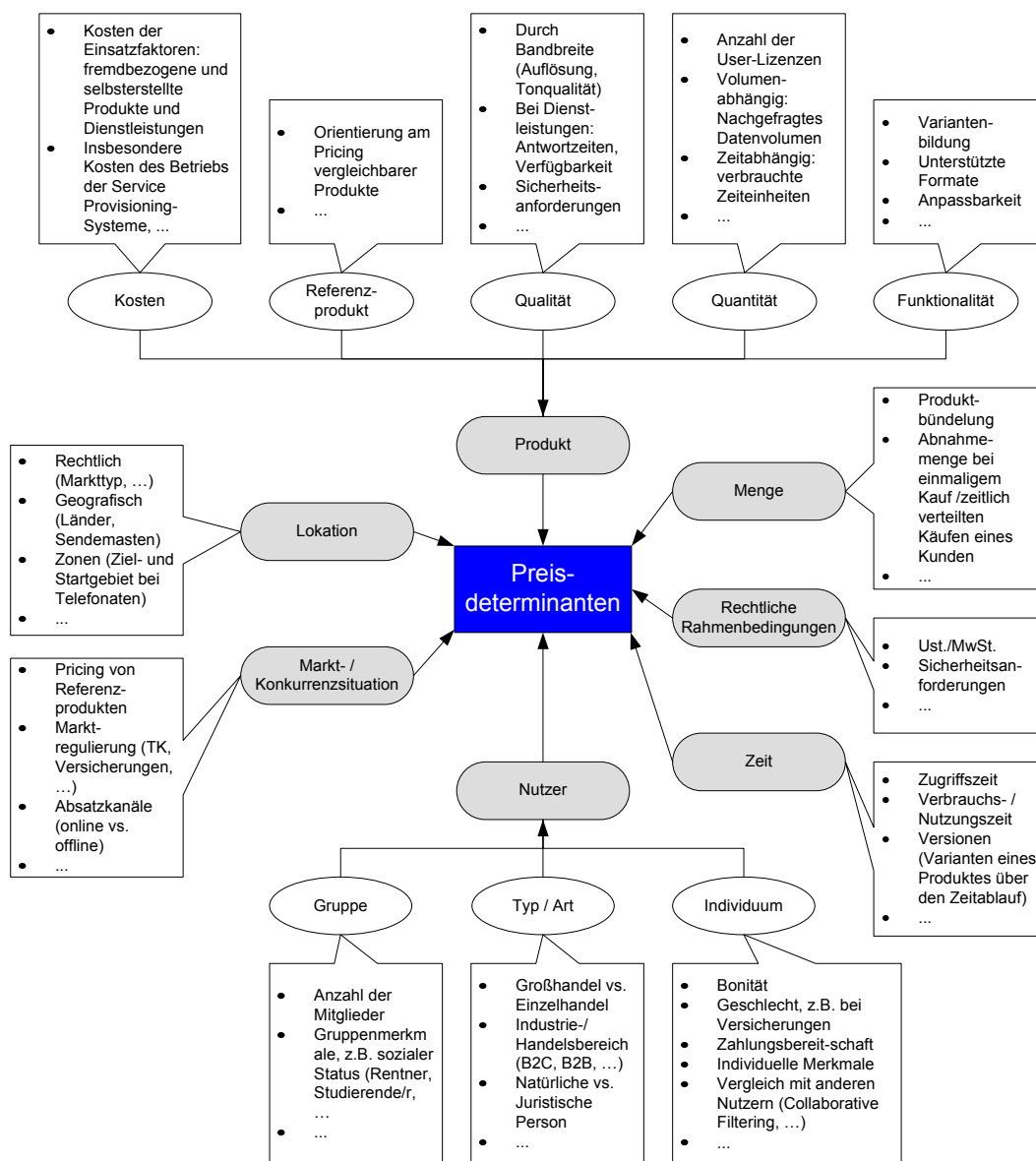


Abbildung 6: Preisdeterminanten für digitale Produkte

## 2.4 Preisdifferenzierung bei digitalen Produkten

Der Begriff der Preisdifferenzierung wird in der Literatur unterschiedlich definiert. Nach (Faßnacht 2003) liegt dann *Preisdifferenzierung* vor, wenn ein Produkt hinsichtlich der räumlichen, zeitlichen, leistungs- und mengenbezogenen Dimensionen identisch ist und zu unterschiedlichen Preisen angeboten wird. Dies trifft auch dann zu, wenn der Anbieter verschiedene Varianten des Gutes zu verschiedenen Preisen verkauft, ohne dass ein komplett neues Produkt entsteht. „Die unterschiedliche Behandlung von Kunden in Bezug auf den Preis [...]“ (Foldvary 2000), S. 264 führt dazu, dass in der Literatur die Preisdifferenzierung auch oft als *Preisdiskriminierung* bezeichnet wird (Geigant u.a. 1993; Faßnacht 1996; Daripa und Kapur 2001). Die Preisdifferenzierung ist nach (Tellis 1986), S. 147 neben der Marktpreisgestaltung und der Produktlinien-Preisgestaltung eine mögliche Preisstrategie.<sup>27</sup>

### 2.4.1 Preisdifferenzierung im Internet

Die Preisfindung und Preisdifferenzierung folgt im Internet den gleichen Regeln wie auf Offline-Märkten (Pohl und Kluge 2000). Aus Sicht des Anbieters bestimmen der „*wahrgenommene Nutzen, Preiselastizitäten, Preis-Absatz-Beziehungen sowie Kosten- und Wettbewerbsinformationen den optimalen Preis.*“ Wesentliche Unterschiede bestehen jedoch in der Dynamik und Komplexität der Preisfestlegung im Internet. Die folgenden Charakteristika beschreiben, wie sich die Komplexität im Pricing durch Nutzung des Internets ändert:

- Die hohe *Informationstransparenz* erleichtert Kunden die Durchführung von Preisvergleichen zwischen unterschiedlichen Anbietern. Aber auch Anbieter können sich über die Situation der Wettbewerber insbesondere deren Preissetzung einen einfacheren Überblick verschaffen. Beispiel: kayak.com für Flugbuchungen.
- Durch eine *globale Vernetzung* wird die Preisdifferenzierung erschwert. Konflikte zwischen dem Internet als Absatzkanal und bspw. Filialen eines Unternehmens können entstehen.

---

<sup>27</sup> Nach (Tellis 1986), S. 147 ist die Preisstrategie „a reasoned choice from a set of alternative prices (or price schedules) that aim at profit maximization within a planning period in response to a given scenario.“

- Elektronische Medien erleichtern die *Kommunikation* zwischen Produzenten/Anbietern und Kunden. Auktionen können mit geringem Aufwand durchgeführt werden und Nachfrager können durch Nachfragebündelung ihre Marktmacht erhöhen. Beispiele: priceline.com; ihrpreis.de
- *Preisänderungen* können schnell vorgenommen werden. Menu-Costs fallen nur für die Preisänderungen auf den Webseiten der Anbieter an.
- *Online-Merkmal*: Durch die Sammlung von Kundenprofilen und Vergleich der Kunden untereinander (Collaborative-Filtering), wird es möglich, individualisierte Produkte und Preise anzubieten. Beispiel: amazon.de.
- Die Leistungsabwicklung über das Internet ermöglicht eine zeitabhängige Preisabrechnung, z. B. bei Online-Rollenspielen (World of Warcraft, [www.wow-europe.com/de/](http://www.wow-europe.com/de/)).

#### **2.4.2 Ziele der Preisdifferenzierung**

(Kelly 1998; Zerdick u.a. 2000) zeigen, dass digitale Produkte im Internet den Endverbrauchern oft kostenlos oder sehr kostengünstig angeboten werden. Dabei stand bisher der schnelle Aufbau einer kritischen (Kunden-)Masse anstelle der Realisierung von Erlösen im Vordergrund. Eine Differenzierung der Preise für unterschiedliche Varianten digitaler Produkte oder nach Personengruppen bietet hingegen die Möglichkeit, identische und gleichartige Produkte parallel zu vertreiben und dabei unterschiedlich hohe Erlöse zu erzielen. So kann neben Freeware-Versionen einer Software eine bepreiste Premium-Variante vertrieben werden, die zusätzliche Funktionen bietet. Durch den Einsatz von Differenzierungsstrategien für digitale Produkte kann ein Produzent Erlöse erzielen bzw. seine Erlöse durch Zuordnung unterschiedlicher Preise zu Benutzergruppen erhöhen (Skiera und Spann 2002), S.282.

#### **2.4.3 Grade der Preisdifferenzierung nach Pigou**

Nach Pigou (1932) lassen sich drei Grade der Preisdifferenzierung unterscheiden: Bei einer *Preisdifferenzierung ersten Grades* zahlt der Käufer einen Preis, der seiner maximalen Zahlungsbereitschaft entspricht:

*„A first degree would involve the charge of a different price against all different units of commodity, in such wise that the price exacted for each was equal to the demand price for it, and no consumers' surplus was left to the buyers.“ (Pigou 1932), § II.XVII.5.*

Voraussetzung dafür ist, dass der Verkäufer die Zahlungsbereitschaft der Nachfrager kennt. Die Erhebung von Zahlungsbereitschaften soll in dieser Arbeit nicht diskutiert werden, erweist sie sich doch i. d. R. als schwierig, da der Kunde aus Gründen der eigenen Nutzenmaximierung einen Anreiz hat, seine Zahlungsbereitschaft zu niedrig darzustellen und somit seine Konsumentenrente aus dem Kauf zu erhöhen. Die Preisdifferenzierung ersten Grades ist – verglichen mit den anderen Formen der Preisdiskriminierung – in der Praxis nur bei Auktionen im Internet zu finden (Faßnacht 2003).

Bei der *Preisdifferenzierung zweiten Grades* erfolgt die Festlegung verschiedener Preise pro Einheit eines Gutes abhängig von der nachgefragten Menge:

Unterschiedliche Preise pro Einheit ergeben sich für die Konsumenten je nach der abgenommenen Stückzahl. *Nicht-lineare Preise* liegen dann vor, wenn sich die gesamten gezahlten Beträge nicht linear mit der nachgefragten Menge verändern. Beispiele für diese Form der Preisdifferenzierung sind Mengenrabatte oder Tarife, die sich aus einer Grundgebühr und einem konstanten Stückpreis zusammensetzen. Dem Kunden wird je nach abgenommener Menge ein unterschiedlich hoher Preisnachlass gewährt.

Unter *Preisdifferenzierung zweiten Grades* werden gemäß Pigou auch *Segmentierungsstrategien* gefasst:

*„A second degree would obtain if a monopolist were able to make  $n$  separate prices, in such wise that all units with a demand price greater than  $x$  were sold at a price  $x$ , all with a demand price less than  $x$  and greater than  $y$  at a price  $y$ , and so on.“ (Pigou 1932), § II.XVII.5.*

Bei der Segmentierung werden Konsumenten bezüglich ihrer maximalen Zahlungsbereitschaft in einzelne Gruppen, die Segmente, eingeteilt. Je nach Segment wird ein unterschiedlicher Preis für das Produkt festgelegt, wobei der Preis im jeweiligen Segment für alle Kunden gleich ist. Durch Selbstselektion ordnen sich nach der Segmentierung die Kunden den Gruppen zu. Es wird angenommen, dass sich Nachfrager mit einer hohen Zahlungsbereitschaft auch für

das entsprechende Segment mit dem hohen Produktpreis entscheiden.<sup>28</sup> Diese Art der Preisdifferenzierung ähnelt der Preisdifferenzierung ersten Grades, sofern die Differenzierung nach Käufergruppen und nicht nach einzelnen Käufern erfolgt.

Die Segmentierung wird durch das Angebot von *Varianten eines Produktes* möglich (z. B. Standard-Variante oder Premium-Variante). (Shapiro und Varian 1999), S.53. (Shapiro und Varian 1999; Daripa und Kapur 2001) bezeichnen diese Form der Preisdiskriminierung auch als *Versioning*.<sup>29</sup>

Eine *Preisdifferenzierung nach der Menge* (erste Interpretation) kann zusätzlich in den Segmenten angewendet und als spezielle Ausprägung dieses Typs der Preisdiskriminierung betrachtet werden.

Die Grenze zwischen Preis- und Produktdifferenzierung ist dabei fließend. Nach (Foldvary 2000), S. 267 handelt es sich bei der Variantenbildung bereits um eine Form der *Produktdifferenzierung*. Denn die Produktdifferenzierung ist durch „[geringfügige], aber subjektiv bedeutsame Abweichungen in den Eigenschaften von Produkten, die einander ansonsten sehr ähnlich sind, [...]“ gekennzeichnet. Auch die *Leistungsbündelung* (Bundling) kann nach (Shapiro und Varian 1999) als spezielle Form der Variantenbildung betrachtet werden. Bei geringen Grenzkosten, z. B. bei digitalen Produkten, kann eine Variantenbildung genutzt werden, um höhere Erlöse zu erzielen. Dabei kann die Zusammenstellung von Varianten durch Bereitstellung von Produktbaukästen auf den Konsumenten übertragen werden (Brandtweiner 2000).

Bei der *Preisdifferenzierung dritten Grades* erfolgt nach (Faßnacht 2003) die Segmentierung der Nachfrager hinsichtlich „beobachtbarer und ansprechbarer Kriterien“:

*„A third degree would obtain if the monopolist were able to distinguish among his customers  $n$  different groups, separated from one another more or less by some practicable mark, and could charge a separate monopoly price to the members of each group.“ (Pigou 1932), § II.XVII.5.*

Die Preise des Produktes sind direkt an die Gruppenzugehörigkeit gebunden. Ein Wechsel zwischen den Konsumentengruppen ist i. d. R. nicht möglich. Kriterien

---

<sup>28</sup> Dazu müssen die Produkte zumindest schwach differenziert sein, d.h. sich mindestens in einer Dimension voneinander unterscheiden.

<sup>29</sup> Bei Software werden unter Versionen eines Produktes Weiterentwicklungen verstanden, die sich in zeitlicher Folge ergeben.

für die Einteilung können bspw. die Zugehörigkeit zu einem Berufsstand, das Alter des Konsumenten oder die Mitgliedschaft in einem Verein sein. Ein Beispiel hierfür sind Forschungs- und Lehre-Lizenzen, bei denen eine Software zu einem geringen Preis an Mitarbeiter und Studierende von Hochschulen lizenziert wird.

Aus den Beispielen zur Preisdifferenzierung zweiten und dritten Grades wird die praktische Relevanz gegenüber der Preisdifferenzierung ersten Grades deutlich. Der Einsatz von Preisdifferenzierungsstrategien als preispolitisches Instrument und Teil des Marketing-Mixes (Borden 1964) muss durch den Produzenten und Verkäufer detailliert geplant werden. Insbesondere muss die Preisstruktur, die in den in Kapitel 5.3 beschriebenen Preismodellen definiert wird durch die vorhandenen Abrechnungssysteme des Unternehmens umgesetzt werden können bzw. neue Systeme eingeführt werden. Im folgenden Absatz soll der Begriff der Preis- und Produktdifferenzierung nach (Schmid 1965; Faßnacht 2003) anhand unterschiedlicher Bestimmungskriterien präzisiert werden.

#### **2.4.4 Preisdifferenzierung nach Güterart und Kostenstruktur**

Die Definition der Preisdifferenzierung ist nicht nur von der Preisgestaltung abhängig, sondern darüber hinaus ist die jeweils vorliegende Güterart (homogene vs. heterogene Güter) in die Betrachtung mit einzubeziehen, da sie durch eine bestimmte Kostenstruktur geprägt ist. Der Produktpreis, die Güterart und die Kostenstruktur der Güter sind die Bestimmungskriterien der Preisdifferenzierung (vgl. Tabelle 3, aus (Schmid 1965), S.41).

Der Begriff der *Preisdiskriminierung* wird in der Literatur meistens auf die unterschiedliche Preisfestlegung hinsichtlich eines homogenen Produktes<sup>30</sup> für verschiedene Konsumenten (bei gleichen Kosten) zurückgeführt (Albers 2001; Albers u.a. 2001; Faßnacht 2003), S.102; S.486 (vg. Fall B). Die Herstellung heterogener Produkte verursacht unterschiedliche Kosten, die bei der Betrachtung zur Preisdifferenzierung berücksichtigt werden müssen, da sie bei einem kostenbasierten Pricing Bestandteil der Preisforderung sind. Die Stückkosten sind für die Systematik von besonderer Bedeutung.<sup>31</sup> Aus der Übersicht geht hervor,

---

<sup>30</sup> Güter sind homogen, wenn sie sich im Urteil der Konsumenten nicht unterscheiden (Feess 1994).

<sup>31</sup> Vgl.(Schmid 1965), S. 39-40: Liegen für die Kostenkalkulation keine Stückkosten vor, werden die Grenzkosten berücksichtigt. Es werden aber nur die durchschnittlichen variablen Totalkosten

dass bei gleichen Preisen und Kosten (Fall A und F), unabhängig davon, ob es sich um homogene oder heterogene Güter handelt, keine Preisdifferenzierung vorliegt.

		Homogene Güter		Heterogene Güter	
		Gleiche Kosten	Verschiedene Kosten	Gleiche Kosten	Verschiedene Kosten
Preise	Gleiche Preise	<b>A</b> Keine Preisdifferenzierung	<b>C</b> Preisdifferenzierung möglich	<b>F</b> Keine Preisdifferenzierung	<b>H</b> Preisdifferenzierung möglich
	Verschiedene Preise	<b>B</b> Preisdifferenzierung	<b>D</b> Preisdifferenz = Kostendifferenz: keine Preisdifferenzierung  <b>E</b> Preisdifferenz ≠ Kostendifferenz: Preisdifferenzierung	<b>G</b> Preisdifferenzierung möglich	<b>I</b> Preisdifferenz = Kostendifferenz: keine Preisdifferenzierung  <b>K</b> Preisdifferenz ≠ Kostendifferenz: Preisdifferenzierung

**Tabelle 3: Systematik der Preisdifferenzierung**

Gerade bei digitalen Produkten haben die Fälle F und G, im Gegensatz zu physischen Produkten, eine große Bedeutung, da die Kosten der Reproduktion eines digitalen Produktes i. e. S. sich nicht von den Reproduktionskosten eines anderen digitalen Produktes unterscheiden.

Sind die Kosten homogener bzw. heterogener Produkte bei gleichen Preisen unterschiedlich (Fall C und H), sollte ein Anbieter über die Einführung von Preisdifferenzierung nachdenken. Bei unterschiedlichen Kosten homogener Güter, ist Preisdifferenzierung nur dann als preispolitisches Instrument anzusehen, wenn die Preisdifferenz nicht der Kostendifferenz entspricht (Fall E, sonst Fall D). Gleiches gilt bei heterogenen Erzeugnissen (Fall G und K). Eine Preisdifferenzierung mit verschiedenen Preisen wird als *offene Preisdifferenzierung* und mit gleichen Preisen als *versteckte Preisdifferenzierung* bezeichnet. Die Angabe nicht identischer Preise kann dabei als Hinweis für eine mögliche Diskriminierung gedeutet werden und wird deshalb als „offen“ angesehen. Es liegt in allen Fällen, in denen eine Preisdifferenzierung möglich ist

---

zugrunde gelegt, die als Durchschnittskosten der gesamten variablen Kosten für die Herstellung einer Einheit eines Produktes interpretiert werden.

(Fälle C, H, E, G und K), auch tatsächlich eine vor, wenn sie bewusst als preispolitisches Instrument eingesetzt wird. Bei heterogenen Gütern wird stets von Produktdifferenzierung gesprochen (Fälle F, G, H, I und K). Jedoch nur die Fälle F und I entsprechen einer Produktdifferenzierung in Reinform.

Die vorgestellte Systematik kann von Unternehmen genutzt werden, um einen Überblick über die Anwendungsmöglichkeiten der Preisdifferenzierung zu erhalten. Ist Preisdifferenzierung als Preisstrategie möglich und beabsichtigt, so muss sich der Anbieter für eine oder mehrere Implementationsformen der Preisdifferenzierung entscheiden, die im folgenden Absatz beschrieben werden.

#### **2.4.5 Implementationsformen der Preisdifferenzierung**

Die Implementationsformen der Preisdifferenzierung nach (Faßnacht 1996), S. 55 ff; (Faßnacht 2003) beschreiben die *konkrete Umsetzung von Preisstrategien* und umfassen

- die Preisdifferenzierung nach Konsumenten/Marktsegmenten,
- die Preisdifferenzierung nach der Menge und
- die Preisdifferenzierung nach den gehandelten Produkten.

Unter die *Preisdifferenzierung nach Konsumenten/Marktsegmenten* fallen die personenbezogene, räumliche, zeitliche und leistungsbezogene Preisdifferenzierung sowie die Mehr-Personen-Preisbildung, die sich an den maximalen Zahlungsbereitschaften der Konsumenten orientieren.<sup>32</sup>

Ist ein Produkt in seinen vier Dimensionen (räumlich, zeitlich, leistungs- und mengenbezogen) identisch und wird es verschiedenen Konsumenten (Konsumentengruppen) zu unterschiedlichen Preisen in Abhängigkeit von personenspezifischen Charakteristiken angeboten, liegt eine *personenbezogene Preisdiskriminierung* vor.

Analog dazu ist bei der *Mehr-Personen-Preisbildung* der Preis pro Einheit eines Produktes von der Anzahl der Personen abhängig.<sup>33</sup> Dies gilt auch für den Fall, dass Varianten eines Gutes angeboten werden, die sich hinsichtlich einer Dimension unterscheiden, aber kein neues Erzeugnis bilden.

---

<sup>32</sup> Der Maximalpreis kann durch den individuellen Nutzen oder die Preiselastizitäten ersetzt werden. Die Preiselastizität der Nachfrage ist die prozentuale Veränderung der Nachfrage nach einem Gut bei einer prozentualen Preisänderung.

<sup>33</sup> Die Lizenzkosten des Lern-Managementsystems WebCt sind bspw. von der Nutzeranzahl abhängig.



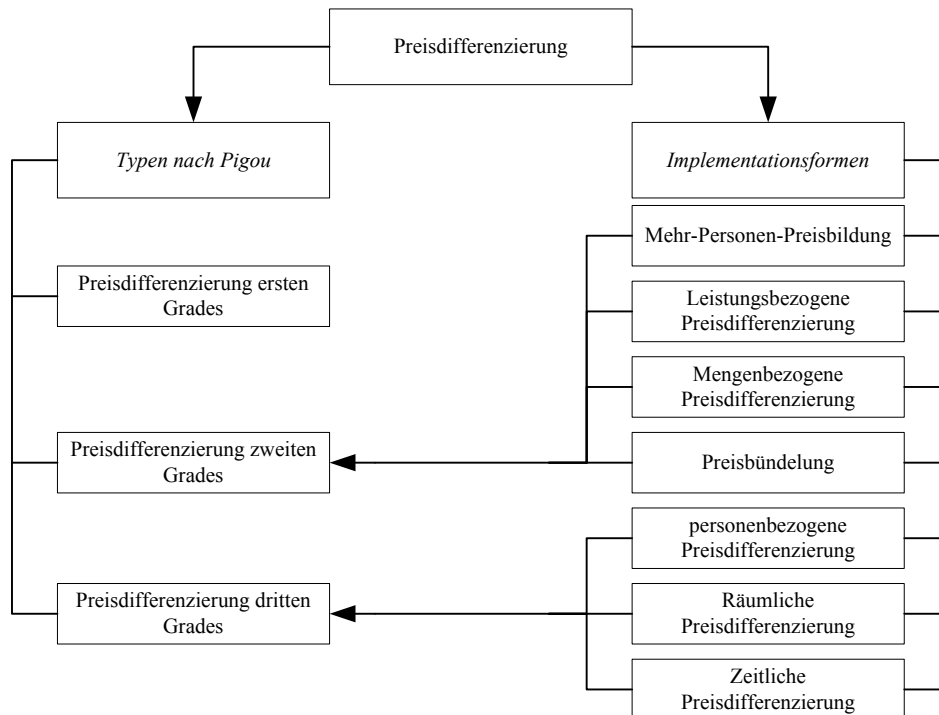
Zur *mengenbezogenen Preisdifferenzierung* wird z. B. die Preisdifferenzierung anhand von Mengenrabatten, durch Bonusprogramme, Blocktarife und zweiteilige Tarife gezählt. Die mengenbezogene Preisdifferenzierung beschreibt eine Situation, in der der Anbieter den durchschnittlichen Preis pro Einheit von Varianten eines Produktes mit der nachgefragten Menge verändert. Die anderen Dimensionen der Güter (Zeit, Raum und Leistung) bleiben dabei konstant.

Die *Preisdifferenzierung nach Produkten* orientiert sich an den unterschiedlichen Zahlungsbereitschaften der Käufer für die einzelnen Produkte eines Anbieters. Werden mehrere Erzeugnisse in einem Paket zu einem Gesamtpreis angeboten, liegt eine *Preisbündelung* vor<sup>34</sup>. Der Preis des Bündels ist dabei i. d. R. kleiner als die Summe der Einzelpreise der Produkte. Da die Grenzkosten der Produktion digitaler Produkte gegen Null gehen, lohnt sich oft eine *Produkt-/Preisbündelung* mehrerer digitaler Produkte (z. B. Vertrieb eines Office-Pakets statt Einzelvertrieb eines Textverarbeitungsprogramms). Ziel der Produktbündelung ist es aus Sicht des Anbieters, neue Kundengruppen zu erschließen, die zuvor die Einzelprodukte nicht erworben hätten oder nur einen Teil dieser Produkte (Olderog und Skiera 2000).

*Räumliche, zeitliche und leistungsbezogene Preisdifferenzierung* bedeuten, dass Varianten eines Produktes angeboten werden, die sich nur im Hinblick auf die entsprechende Dimension (*eindimensionale Preisdifferenzierung*) unterscheiden und zu verschiedenen Preisen verkauft werden. Eine *mehrdimensionale Preisdifferenzierung* liegt vor, wenn nach mehreren Dimensionen, z. B. nach Zeit und Lokation, unterschieden wird.

---

<sup>34</sup> Eine Übersicht über die Preisbündelung bei Informationsgütern bieten z. B. (Bakos und Brynjolfsson 1999).



**Abbildung 7: Typen und Implementationsformen der Preisdifferenzierung<sup>35</sup>**

Im Zusammenhang mit der Einteilung der Preisdiskriminierung nach Pigou werden die personenbezogene, die räumliche und die zeitliche Preisdifferenzierung der Preisdifferenzierung dritten Grades und die Mehr-Personen-Preisbildung, die Preisbündelung sowie die leistungs- und mengenbezogene Preisdifferenzierung der Preisdiskriminierung zweiten Grades zugeordnet (Faßnacht 2003), S. 497. Der Gesamtzusammenhang wird in Abbildung 7 dargestellt.

Die einzelnen Implementationsformen der Preisdifferenzierung lassen sich auch dahingehend unterscheiden, ob sich die Konsumenten selbst ein adäquates Produkt und den zugehörigen Preis suchen (*mit Selbstselektion*) oder durch den Anbieter einer Kundengruppe (*ohne Selbstselektion*) zugeordnet werden. Ein Klassifikationsschema nach Art der Selektion, das in Abbildung 8 aufgeführt ist, wird in (Skiera und Spann 2002), S. 274 beschrieben.

<sup>35</sup> Quelle: (Faßnacht 2003), S. 497.

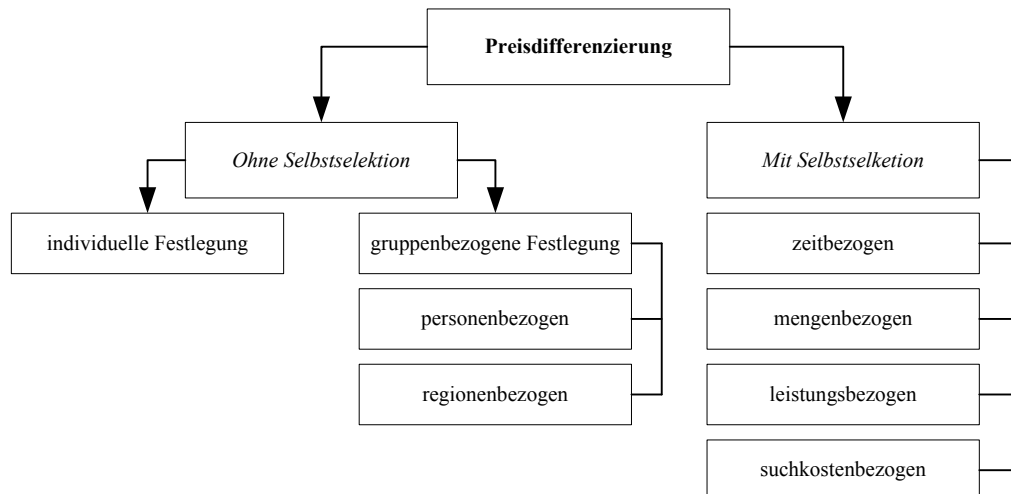


Abbildung 8: Klassifizierung der Preisdifferenzierung nach der Selbstselektion<sup>36</sup>

Die *individuelle Preisdifferenzierung* ist dabei der Preisdifferenzierung ersten Grades nach Pigou gleichzusetzen. Die *regionenbezogene Preisdiskriminierung* entspricht der Implementationsform der räumlichen Preisdiskriminierung und gehört zusammen mit der *personenbezogenen Differenzierung* zur *Preisdifferenzierung nach Gruppen*. Alle anderen Arten sind mit den oben beschriebenen, gleichnamigen Implementationsformen identisch. Die *Mehr-Personen-Preisbildung*, bei der die Anzahl der Kunden über den Preis des Produktes entscheidet, ist jedoch nicht in der Klassifizierung auf Basis der Selbstselektion aufgeführt, obwohl auch diese Form der Preisbildung im Internet zu finden ist ([www.letsbuyit.com](http://www.letsbuyit.com)). Bei der zeitlichen Preisdifferenzierung weisen Konsumenten verschiedene Zahlungsbereitschaften nach Tageszeiten auf. Eine *zeitliche Preisdifferenzierung* ist in der Regel bei digitalen Dienstleistungen, z. B. der Bereitstellung von Internetzugang, gegeben. Bei der Mehr-Personen-Preisbildung entsteht den Käufern ein Mehraufwand dahingehend, dass sie weitere Konsumenten für ihr Produkt finden müssen. Der Preis wird dann in Abhängigkeit der Käuferzahl und damit der Stückzahl des Produktes festgelegt. Der durchschnittliche Preis pro Einheit sinkt mit steigender Nachfrage (Rabattierung auf Basis der Personenzahl). Die *suchkostenbezogene Preisdifferenzierung* hängt direkt mit der Wahl des Distributionskanals durch den Anbieter zusammen. Dabei wird davon ausgegangen, dass unterschiedliche Vertriebskanäle verschiedene Suchkosten für die Konsumenten verursachen. Niedrigere Produktpreise sollten aus Sicht des Verkäufers mit einer intensiveren

<sup>36</sup> Quelle: (Skiera und Spann 2002), S.274.

Suche und damit mit größeren Opportunitätskosten für den Konsumenten verbunden werden. Eine Erleichterung bei der Suche günstiger Produkte versprechen (Meta-)Suchmaschinen für Produkte im Internet (z. B. kayak.com für Flugreisen oder geizkragen.de für Produkte der Unterhaltungselektronik). Da der Nachfrager über die Höhe der Suchkosten entscheiden kann, zählt auch die suchkostenbezogene Preisdifferenzierung zur Preisdifferenzierung mit Selbstselektion. Die suchkostenbezogene Preisdifferenzierung wird aufgrund der Fokussierung auf das Internet als Vertriebskanal in der Arbeit nicht weiter betrachtet.

Die Klassifizierung der Arten der Preisdifferenzierung mit oder ohne Selbstselektion nach (Skiera und Spann 2002) betrachtet nur den *Ein-Produkt-Fall*. Beruhen die Unterschiede der Varianten eines Gutes in der Veränderung nur einer Produktdimension, ist die Preisdiskriminierung *eindimensional*, in den anderen Fällen *mehrdimensional*. Die Implementationsform der Preisbündelung ist jedoch dem *Mehr-Produkt-Fall* zuzurechnen und kann nicht eindeutig zugeordnet werden.

#### 2.4.6 Preisdifferenzierung und E-Learning

In einem ersten Schritt muss ein E-Learning-Provider prüfen, ob die Einführung einer Bepreisung der E-Learning-Angebote sowie die Einführung von Preis(differenzierungs)strategien überhaupt zulässig und wünschenswert sind. Danach muss entschieden werden, welche Preisstrategien angewendet werden sollen (Preisdifferenzierung ersten bis dritten Grades). Erst im Anschluss an die Wahl der Preisdifferenzierungsstrategie wird eine passende Implementationsform gewählt (vgl. Abbildung 9).



Abbildung 9: Einführung von Preisdifferenzierungsstrategien

##### 2.4.6.1 Preisstrategien im E-Learning

Um eine *Preisdifferenzierung ersten Grades* nach Pigou (vgl. Kapitel 2.4.3, S. 28) einzuführen, müssen die Zahlungsbereitschaften der einzelnen Kunden /

Studierenden bekannt sein. Hierzu wären umfangreiche Erhebungen notwendig.<sup>37</sup> Die Preisdifferenzierung ersten Grades ist daher im E-Learning mit einer Vielzahl unterschiedlicher Kunden nicht praktikabel.

Eine *Preisdifferenzierung zweiten Grades* entlang der Zahlungsbereitschaften nach Pigou (Segmentierungsstrategie) setzt wie die Preisdifferenzierung ersten Grades voraus, dass Zahlungsbereitschaften in den Kundensegmenten durch den Anbieter erhoben werden. Dabei wird angenommen, dass sich die Kunden entsprechend ihrer Zahlungsbereitschaft selbstständig einem Segment zuordnen. Die Zuordnung kann durch das Anbieten unterschiedlicher Varianten eines Produktes erleichtert werden. Im E-Learning können neben einer Premium-Variante eines Kurses Standard-Varianten angeboten werden, die über zusätzliche Dienstleistungen oder einen höheren Service-Level verfügen.

Eine Preisdifferenzierung zweiten Grades bietet sich an, wenn der Anbieter keine Möglichkeit hat, Kunden anhand beobachtbarer Kriterien spezifischen Kundengruppen zuzuordnen, aber die Zahlungsbereitschaft der Kundengruppen im Ansatz kennt.

Die Berücksichtigung zweiteiliger Tarife (alternative Definition der Preisdifferenzierung zweiten Grades) ist z. B. durch Preisbündelung von Kursen oder durch Bepreisung einzelner Lernobjekte in einem einzelnen Kurs möglich. Im ersten Fall erhält der Kunde bei der Buchung eines weiteren Kurses einen Rabatt.

Im Fall der *Preisdifferenzierung dritten Grades* wird vorausgesetzt, dass der Anbieter von E-Learning-Produkten seine Kunden anhand messbarer Kriterien bestimmten Kundengruppen zuordnet. Eine Zuordnung und eine Erhebung von Unterscheidungskriterien erfolgen z. B. während des Immatrikulationsprozesses eines Studierenden an einer Hochschule.<sup>38</sup> Neben der Einteilung in Kundengruppen ist eine Einteilung der Studierenden in Studienphasen wichtig, da bspw. E-Learning-Produkte aus dem Hauptstudium nur bei bestandener Zwischenprüfung nachgefragt werden dürfen. Es handelt sich um eine Gruppenbildung über mehrere Stufen. Um Studierende unterschiedlichen Gruppen eindeutig zuordnen zu können, ist es wichtig, dass sich der einzelne Studierende

---

<sup>37</sup> Für die Erhebung von Zahlungsbereitschaften bietet sich u.a. die Conjoint-Analyse an (Blois u.a. 2000).

durch einen Bildungsprovider einer Kundengruppe zuordnen lässt (z. B. während der Immatrikulation). Standards, die eine internationale Vereinheitlichung der Leistungen eines Studierenden unterstützen, sind für die Zuordnung zu Gruppen sinnvoll und notwendig.<sup>39</sup> Neben der reinen Preisdifferenzierung dritten Grades (Fall B in Tabelle 3, S. 32) z. B. durch den Vertrieb der gleichen Leistung an reguläre Studierende und Studierende der „Universität des dritten Lebensalters“, kann eine Preisdifferenzierung in Kombination mit einer Produktdifferenzierung (Standard- und Premium-Variante, s. Preisdifferenzierung zweiten Grades) eingeführt werden (Fall G, sofern davon ausgegangen wird, dass die Kosten der Leistungserstellung gleich hoch sind, sonst Fall K).

#### **2.4.6.2 Wahl der Implementationsform**

Die *Wahl einer Implementationsform* der Preisdifferenzierung wurde in Kapitel 2.4.5 beschrieben. Zur Unterstützung von Preisdifferenzierung wird oft zeitgleich eine Produktdifferenzierung eingeführt. Dabei können Produkte als Eigenschaftsbündel beschrieben und variiert werden (vgl. Kapitel 2.3.6, 2.3.7).

Im folgenden Abschnitt werden daher grundlegende Komponenten (Lernobjekte und Prozesse) von „E-Learning“ als digitales Produkt beschrieben.

### **2.5 Vermarktung von E-Learning als digitales Produkt**

Unter *E-Learning* werden unterschiedliche Formen der Lehre über elektronische Medien zusammengefasst. Ein wesentlicher Aspekt des E-Learnings ist, dass das Lernen durch Lernplattformen und *Lern-Managementsysteme* (LMS) unterstützt wird (Beutner und Twardy 2003). Diese stellen Lehrenden und Lernenden unterschiedliche Dienste online zur Verfügung und ermöglichen die Sichtung und den Download von Lernobjekten. Hierdurch hebt sich E-Learning von anderen Formen der elektronischen Lehre und Vorläufern des E-Learning, z. B. *Intelligent Tutoring Systems* (ITS) und *Computer-based Training* (CBT), bei dem der Lernende datenträgerbasierte Lernobjekte bearbeitet, ab (Thimm 2005). Der Grad der Virtualisierung ist im E-Learning oft geringer als in CBTs, da der persönliche Kontakt zwischen Lehrenden und Lernenden in E-Learning-Veranstaltungen

---

<sup>38</sup> Die Kundengruppe ist dann z. B. „regulärer Studierender“. Als weitere Kundengruppen kommen „Executive Student“, „Teilzeitstudierender“, „Studierender an der Berufsakademie“ etc. in Frage.

<sup>39</sup> Durch die Einführung von zentralen Zertifikaten kann eine mehrfache Zertifizierung eines Studierenden und der dadurch entstehende Verwaltungs-Overhead vermieden werden.

einen wesentlichen Erfolgsfaktor des E-Learning ausmacht (Grüne u.a. 2006). In der Pädagogik wird daher oft der Begriff *Blended Learning* anstelle des eher technisch und ökonomisch geprägten Begriffs E-Learning verwendet (Kerres 2001; Sauter u.a. 2002). Blended Learning basiert auf konstruktivistischen Lerntheorien und setzt – im Gegensatz zum traditionellen Verständnis des E-Learnings – einen Wechsel zwischen Online- und Präsenzphasen voraus. In der Arbeit wird der Begriff E-Learning in Anlehnung an (Wilms 2004), S. 91 angesehen als:

*„[...] E-Learning umfasst computerunterstützte Lernprozesse [und Lehrprozesse], die insbesondere durch eine Interaktion zwischen dem Lernsystem [Lern-Managementsystem], den Lernenden und dem Lehrenden gekennzeichnet sind.“*

In *Lehr- und Lernprozessen*, die die digitalen Bildungs-Dienstleistungen im E-Learning abbilden, werden *Lernobjekte*, die digitalen Produkte des E-Learnings, verarbeitet.

E-Learning ist für den Vertrieb und das Pricing digitaler Produkte aus folgenden Gründen interessant:

- Zum einen weist E-Learning als digitales Produkt sowohl einen Produktcharakter (Lernobjekte), als auch einen Dienstcharakter (z. B. Lehr- und Lernprozesse) auf. Für eine Bepreisung von E-Learning kommen deshalb neben einem Flat-Fee-Pricing auch nutzungs- und volumenabhängige Preismodelle in Betracht. Das Pricing im E-Learning kann sich dann an andere serviceorientierten Industrien (Telekommunikation, ASP/SaaS<sup>40</sup>) anlehnen.
- Produkte im E-Learning sind modular aufgebaut. Ein Kurs besteht i. d. R. aus mehreren Lektionen. Durch Customizing können Lerninhalte an unterschiedliche Zielgruppen, z. B. in Executive-Studiengängen oder regulären Studiengängen angepasst werden. Eine Vielzahl möglicher Produktkombinationen muss durch das Pricing unterstützt werden: (1) Versionen von Preismodellen zu abgelaufenen und zukünftigen E-Learning-Angeboten müssen verwaltet werden. (2) Darüber hinaus können zeitgleich unterschiedliche Varianten von Preismodellen für ein

---

<sup>40</sup> SaaS = Software as a Service.

Produkt existieren, so dass Lernende aus unterschiedlichen Preismodellen ihren „optimalen“ Tarif wählen können.

- Die Bepreisung von E-Learning spielt in Deutschland bisher eine untergeordnete Rolle (Mendling u.a. 2005). Die wissenschaftliche Forschung hat sich auf die Kostenrechnung konzentriert, da E-Learning bisher eher als zusätzliche Leistung zu traditionellen Lehrformen angesehen wird, die durch vorgegebene finanzielle Budgets mit abzudecken ist (Ernest&Young 2000; Bacsich u.a. 2001; von Boettcher 2004).
- Zur Sicherung der Nachhaltigkeit von E-Learning-Angeboten ist eine Gegenfinanzierung der Angebote unumgänglich. Diese kann ganz oder in Teilen durch Erhebung von Nutzungsgebühren erfolgen. Dabei ist die Entscheidung über den Anteil der von den Lernenden zu tragenden Gebühren politisch brisant, sofern Kurse bepreist werden, die bisher „frei zugänglich“ waren (s. folgendes Kapitel). Andererseits zeigen professionelle Anbieter von E-Learning-Inhalten, dass eine Bepreisung von E-Learning möglich ist und unterschiedliche Geschäftsmodelle existieren. In Kapitel 2.5.3 werden unterschiedliche Preismodelle von Bildungsprovidern skizziert.
- Unabhängig von der Generierung von Erlösen kann E-Learning neben Vorlesungen und Übungen dabei helfen, gerade an einer Massenuniversität die Betreuungsleistungen zu verbessern. Neben der asynchronen Beantwortung von Fragen seitens der Lernenden z. B. in der Klausurenphase und der Bereitstellung von Software können Lernende miteinander Kontakt aufnehmen und gemeinsam Problemlösungen erarbeiten (Grüne u.a. 2006).

### **2.5.1 Lernobjekte als digitale Produkte im E-Learning**

Obwohl in der öffentlichen Diskussion Bildung oft als freies und damit „kostenfreies“ Gut angesehen wird, ist dies nur zum Teil zutreffend (Bernecker 2005). Insbesondere die Ausbildung an privaten Universitäten und Fachhochschulen unterliegt Kapazitätsrestriktionen und ist in der Regel kostenpflichtig. Auch öffentliche Hochschulen werden in Zukunft versuchen, durch die Verwertung von E-Learning-Inhalten Erlöse zu erzielen (Hesser 2006).



Bildungs-Dienstleistungen, die im E-Learning angeboten werden, sind nicht beliebig skalierbar, selbst wenn durch die Einführung von E-Learning-Aktivitäten die Betreuungsintensität für einzelne Studierende erhöht werden kann (Grüne u.a. 2006). Bildung im Allgemeinen und E-Learning verfügen sowohl über einen Dienstleistungs- als auch einen Produktcharakter (Bernecker 2005). Es bietet sich daher an, Prozesse zur Erbringung der Dienstleistung durch geeignete Software-Systeme zu unterstützen. In E-Learning-Kursen können *Workflow-Managementsysteme* zur technischen Umsetzung wiederkehrender Prozesse, z. B. bei der Anmeldung von Studierenden, der automatisierten Durchführung von Tests und der Bepreisung von E-Learning-Aktivitäten, eingesetzt werden und somit den Leistungserbringern eine Arbeitserleichterung bieten (Grüne u.a. 2003).

*Lernobjekte* bilden das zentrale *Produkt im E-Learning* und stellen Input oder Output von Aktivitäten in E-Learning-Prozessen dar. *Elementare Lernobjekte*, z. B. eine Übungsaufgabe, repräsentieren kleine semantische Einheiten (Knowledge-Units) im Lernobjekt-Management.<sup>41</sup> Sie können zu *komplexen Lernobjekten*, z. B. Übungsblättern oder ganzen Kursen, zusammengesetzt werden. Dabei ist die Frage, in welcher *Granularität* Lernobjekte gespeichert, verwaltet und bepreist werden sollen, schwer zu beantworten, weil einem erhöhten Nutzen aus Wiederverwendbarkeit kleingranularer Lernobjekte ein zunehmender Aufwand der Katalogisierung einer wachsenden Anzahl von Lernobjekten bei Zerlegung der Lernobjekte in immer kleinere Einheiten entgegensteht. Zur Unterstützung der Wiederverwendbarkeit können Lernobjekte durch *Metadaten* beschrieben werden. Die einheitliche Formulierung von Lernobjekt-Metadaten wird durch internationale *Standardisierungsinitiativen*, beispielsweise die Dublin Core Metadata Initiative ([www.dublincore.org](http://www.dublincore.org)), vorangetrieben.<sup>42</sup> Ein wichtiger Standard ist das *Sharable Content Object Model* (SCORM), das von der Advanced Distributed Learning Initiative ([www.adlnet.org](http://www.adlnet.org)) propagiert wird (ADL 2004a, 2004b, 2004c). Die Anreicherung von Lernobjekten mit Preisinformationen ist durch Erweiterungen des SCORM-

---

<sup>41</sup> Vgl. z. B. (Grüne u.a. 2003; Knolmayer und Montandon 2003; Redeker 2003; Vossen und Jaeschke 2003).

<sup>42</sup> Einen Überblick über Standards zur Auszeichnung und Definition von Lernobjekten bieten (Collier und Robson 2002).

Modells u. a. auf der Metadatenebene möglich und kann durch Preismodelle unterstützt werden (Gruene u.a. 2005; Gruene und Oberweis 2005).

(Wiley 2000) betont den *Objektcharakter* von Lernobjekten im Sinne der Objektorientierten Programmierung und betrachtet diese als *gekapselte Lerninhalte*, die interaktiv genutzt werden können, weil sie ein „Verhalten“ aufweisen. Hierzu gehören Simulationswerkzeuge wie Java-Applets zur Supply-Chain-Simulation, interaktive Tests und Web-based Trainings. Wird der Begriff des Lernobjekts weiter gefasst und auf nicht-interaktive Inhalte ausgedehnt, so können auch PDF-Dateien mit Vorlesungsfolien, Audiodateien, E-Papers und E-Books als Lernobjekte angesehen werden. Eine *Aggregation* (Bündelung) von Lernobjekten zu Lernobjekten höherer Ordnung kann aus technischer Sicht im Falle von SCORM durch Einführung von Metadatenstrukturen in Manifest-Files und Verpacken (Packaging) der Lernobjekte in Zip-Archiven ermöglicht werden (Gruene u.a. 2005).

### **2.5.2 Digitale Dienstleistungen im E-Learning**

Nach (Grüne u.a. 2003) können im E-Learning drei Klassen von Diensten (Prozessen) unterschieden werden. Hierzu gehören Lehr-, Lern- und Administrationsprozesse. Zu den *Lehrprozessen* zählen sowohl der Lernobjektentwurf und die -erstellung (Autorenprozess) als auch die Anleitung der Lernenden. *Lernprozesse* beinhalten neben dem eigentlichen Lernen auch den Informationsaustausch mit Lehrenden und anderen Lernenden. *Administrationsprozesse* gliedern sich in technische und organisatorische Prozesse: Technische Prozesse betreffen die Administration des Softwaresystems (z. B. Installations- und Wartungsarbeiten), organisatorische die Verwaltung von Lernobjekten, Lernenden oder die Zeitplanung. Lehr- und Lernprozesse sind selten fest vorgegeben, sondern hängen von den Präferenzen der Lehrenden und Lernenden ab. Um eine kostendeckende Bepreisung im E-Learning zu erreichen, müssen die bei der Durchführung von administrativen Prozessen entstehenden Kosten durch Umlageverfahren, z. B. durch eine Prozesskostenrechnung, auf die Leistungsprozesse umgeschlüsselt werden (Ernest&Young 2000; Bacsich u.a. 2001).

### 2.5.3 Geschäftsmodelle im E-Learning

Nach (Wirtz 2000) bildet ein *Geschäftsmodell* das Leistungssystem eines Unternehmens ab. Es spezifiziert, welche Ressourcen zur Leistungserstellung fließen und wie diese durch Prozesse in *vermarktbar Produkte* transformiert werden. In diesem Sinne kann es als *Architektur* angesehen werden, *in der Produkt- und Informationsflüsse sowie Akteure, Rollen und Erlösquellen festgelegt werden* (Timmers 1998). Das Geschäftsmodell setzt sich aus weiteren Partialmodellen zusammen.

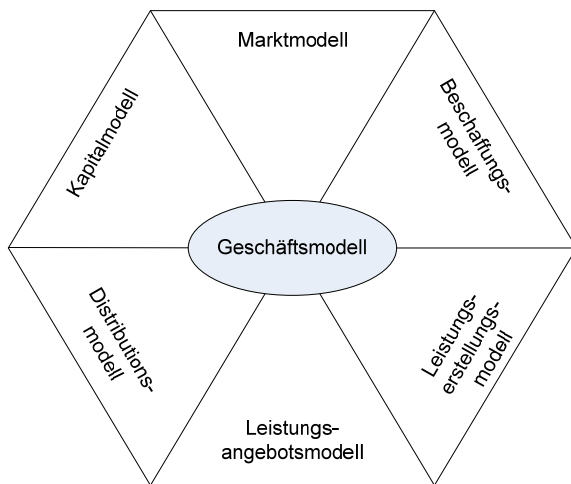
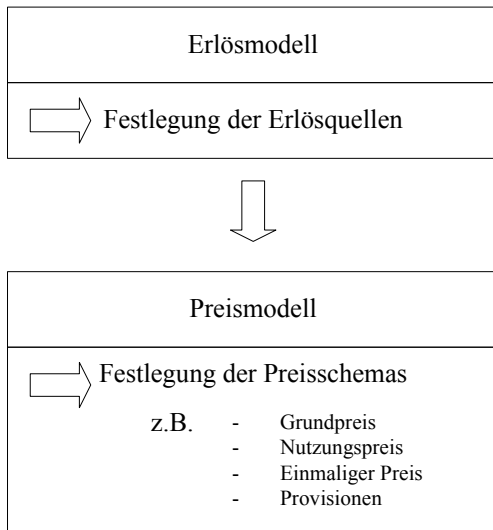


Abbildung 10: Partialmodelle eines Geschäftsmodell<sup>43</sup>

Zu den *Partialmodellen* gehören das Marktmodell, das Beschaffungsmodell, das Leistungserstellungsmodell, das Leistungsangebotsmodell, das Distributionsmodell und das Kapitalmodell (vgl. Abbildung 10). Partialmodelle können wie im Fall des Marktmodells weiter verfeinert werden. Zum Marktmodell gehören die Beschreibung der Struktur der Anbieter- und Nachfragermärkte sowie die Darstellung der Wettbewerbssituation.

Im Folgenden liegt das Hauptaugenmerk auf Erlös- und Preismodellen, die die Ergebnisse einer zweistufigen Erlösentscheidung im Kapitalmodell abbilden. In der zweistufigen *Erlösentscheidung* eines Unternehmens werden nach (Skiera und Lambrecht 2000) zuerst die Erlösquellen in einem Erlösmodell definiert, wonach die Zuordnung von Preisen zu den Erlösquellen in der Preismodellierung erfolgt (vgl. Abbildung 11).

<sup>43</sup> Quelle: modifiziert nach (Wirtz 2000).



**Abbildung 11: Zweistufigkeit der Erlösentscheidung<sup>44</sup>**

#### **2.5.4 Beispiel: Direktvertrieb von E-Learning als Geschäftsmodell**

Beim reinen *Direktvertrieb* von Bildungs-Dienstleistungen werden durch den E-Learning-Anbieter Lernobjekte selbst erstellt und die zugehörigen Prozesse eigenständig umgesetzt. Der Direktvertrieb bietet sich an, wenn ein Bildungsdienstleister bereits über eine große installierte Basis von Nutzern verfügt und Erfahrung im Umgang mit E-Learning-Technologien sowie in der Organisation von E-Learning-Veranstaltungen hat. Eine Vielzahl von Hochschulen versucht über den Direktvertrieb von E-Learning traditionelle Lehrformen wie Vorlesung und Übung zu erweitern. Dabei liegt zurzeit der Fokus eher auf der pädagogischen Anreicherung traditioneller Lehrformen als auf einer ökonomischen Verwertung der angebotenen Inhalte.

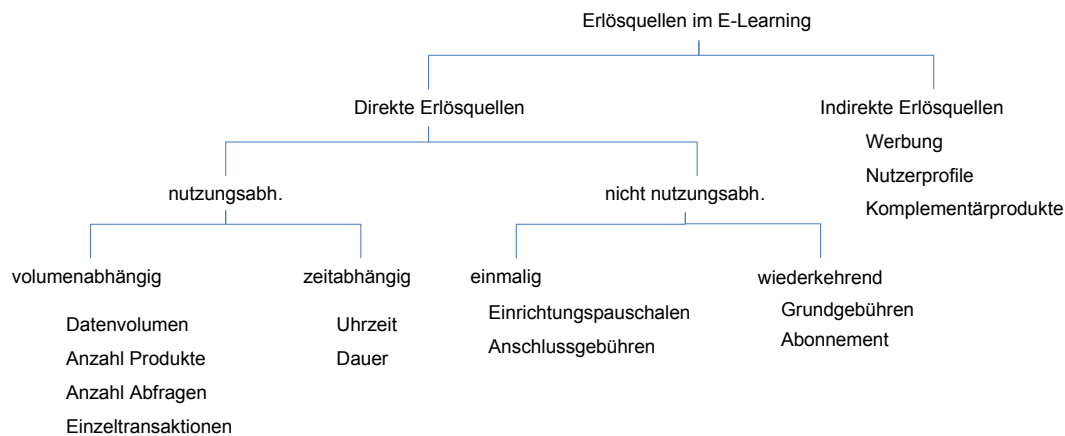
Für den Direktvertrieb von Bildungsleistungen durch Hochschulen sprechen folgende Punkte:

- Hochschulen bündeln fachliches und didaktisches Know-how und können somit den Lernenden ein umfassendes Spektrum an Kursen anbieten.
- Organisationsstrukturen zur Verwaltung hoher Studierendenzahlen sind an Hochschulen vorhanden. Die Studierendenverwaltung ist vielfach bereits online möglich.

---

<sup>44</sup> Quelle: (Skiera und Lambrecht 2000).

- Die Rechenzentren der Hochschulen verfügen i. d. R. über Erfahrungen in der administrativen Betreuung von E-Learning-Technologien, Computernetzwerken und dem Hosting von E-Learning-Applikationen.
- Bildungszertifikate, die durch Hochschulen ausgestellt werden, genießen eine hohe Reputation.



**Abbildung 12: Erlösformen im E-Learning**

Für Hochschulen stellt sich das *Problem der Kostendeckung* im E-Learning durch Generierung von Erlösen aus dem Einsatz von E-Learning (Mendling u.a. 2005). Zur Reduzierung von Entwicklungskosten für E-Learning-Angebote und zur Generierung von Erlösen ist es wichtig, auch auf bestehende Angebote anderer Hochschulen zurückzugreifen und Gebühren für die Nutzung (*Reselling*) der selbst erstellten Lernobjekte gegenüber Lernenden und anderen E-Learning-Providern zu erheben. Die technischen Voraussetzungen zur Bepreisung von E-Learning und zur Integration von bestehenden E-Learning-Angeboten können bspw. durch die Einführung Workflow-gestützter, Portal-basierter E-Learning-Systeme geschaffen werden (Gruene und Oberweis 2005)<sup>45</sup>.

(Ehrenberg 2001) nennt u. a. folgende Voraussetzungen, die vor der Einführung von E-Learning-Angeboten erfüllt werden müssen:

- Hochschulen und Fachbereiche müssen *Strategien* entwickeln, um Nutzungskonzepte zu schaffen, die die Nachhaltigkeit des Einsatzes neuer Medien einschließlich des *unternehmerischen Auftretens* der Hochschulen auf dem Bildungsmarkt sichern.

<sup>45</sup> Vgl. Kapitel 7.2.

- Eine *leistungsfähige IT-Infrastruktur* durch umfassende multimediafähige Ausstattung der Hochschule und der Studierenden ist für den reibungsfreien Unterricht erforderlich.
- *Kooperationen* mit spezialisierten Hochschulen und kommerziellen Multimedia-Firmen sollten bei der *Erstellung und Wartung von Bildungsmodulen* (Public-Private-Partnership) etabliert werden.
- Die *Wiederverwendung* und Mehrfachnutzung von multimedial aufbereiteten Bildungsinhalten muss durch eine konsequente Modularisierung unter Beachtung vorgegebener Standards gewährleistet sein.
- Ideal wäre die *Konzipierung und Realisierung von Forschungsschwerpunkten für die Nutzung neuer Medien* in Studium und Weiterbildung.
- Ein *effektives Hochschulmarketing* sollte die E-Learning-Angebote bekannt machen.
- Hochschulen sollten *international anerkannter Abschlüsse* vergeben, um die internationale Vergleichbarkeit der Abschlüsse zu ermöglichen. Hierzu gehört auch die Bewertung der Studienleistungen nach *Credit-Point-Systemen*.

Im Rahmen der Festlegung des Geschäftsmodells<sup>46</sup> muss ein E-Learning-Provider geeignete *Erlösquellen/-formen* für seine Produkte und Dienstleistungen definieren. Erlösquellen können wie in Abbildung 12 dargestellt in direkte Erlösformen, die mit der Nachfrage nach dem Produkt bzw. der Dienstleistung in Zusammenhang stehen, und indirekte Erlösquellen unterteilt werden. Erlösquellen geben nach (Spann u.a. 2005) an, „[...] was überhaupt bepreist werden soll.“ *Indirekte Erlösquellen* wie Werbung haben im E-Learning eine untergeordnete Bedeutung. Die *direkten Erlösquellen* können in *nutzungsabhängige und nicht nutzungsabhängige Erlösquellen* unterteilt werden. Nicht nutzungsabhängige Erlösquellen sind z. B. einmalig anfallende Einrichtungspauschalen für die

---

<sup>46</sup> Nach (Wirtz 2000) umfasst das Geschäftsmodell eines Unternehmens ein Kapitalmodell, das die Erlös- und Preismodelle sowie die Wahl der Kapitalgeber abbildet, ein Marktmodell (Nachfrager- und Wettbewerbermodell), ein Beschaffungsmodell (Einsatzfaktoren, Lieferanten), ein Modell der Leistungserstellung (Produktionsprozesse etc.), ein Leistungsangebotsmodell, das die Ausgestaltung der Angebote gegenüber den Kunden definiert und ein Distributionsmodell (Wahl des Absatzkanals), s. S. 44 / Kapitel 2.5.3.

Einrichtung des Arbeitsplatzrechners eines Lernenden oder wiederkehrende Grundgebühren für die Nutzung einer Lernplattform. Nach der Festlegung von Erlösformen/-quellen müssen diese durch *Preismodelle* für einzelne Nutzergruppen und Nutzungsverhalten implementiert werden.

### **2.5.5 Anwendungsfall: E-Learning am Penn State World Campus**

Die Pennsylvania State University ist an 24 Standorten im US-Bundesstaat Pennsylvania vertreten und betreute im Herbst 2005 ca. 80.000 Studierende, von denen ca. 69.000 in Undergraduate Programs und ca. 11.000 in Graduate Programs sowie First Professional Programs (Medizin und Recht) eingeschrieben waren. Seit dem Jahr 1998 bietet die Pennsylvania State University ([www.psu.edu](http://www.psu.edu)) 50 universitäre Ausbildungsprogramme als E-Learning-Kurse über den „World Campus“ ([www.worldcampus.psu.edu](http://www.worldcampus.psu.edu)), das universitätseigene E-Learning-System, an. Zu den Zielpersonen des Penn State World Campus gehören sowohl Studierende, die bereits an der Penn State University eingeschrieben sind, als auch Personen, die parallel zur Berufstätigkeit einen Abschluss erwerben möchten, z. B. Soldaten durch ein reguläres Studium oder Manager im Rahmen eines Executive Program. Darüber hinaus können in Zusammenarbeit mit Firmen individuelle Programme ausgearbeitet werden.

Das Geschäftsmodell des Penn State World Campus (PWC) basiert auf direkten Erlösquellen, die an die Bildungsprodukte gebunden sind.

#### **2.5.5.1 Produktmodell**

Der Penn State World Campus vertreibt *drei Arten von Bildungsprodukten*. Neben *Associate/Undergraduate Programs* und *Graduate Programs* bietet PWC *Certificate Programs* an, die mit einem Zertifikat abschließen. Zertifikate setzen auf unterschiedlichen Ausbildungsniveaus an. Neben Undergraduate Certificates werden Graduate Certificates und Professional and Personal Development Certificates angeboten. Professional and Personal Development Certificates dienen u. a. der Persönlichkeitsbildung und umfassen Kurse in Zeitmanagement, Organisation etc. Zu den Graduate Programs zählt auch ein Executive Program (iMBA).

### 2.5.5.2 Erlösmodell

Für die Teilnahme an den Programmen fallen unterschiedliche Gebühren an:

1. Anmeldegebühren für die Anmeldung zu einem Kurs.
2. Tuition: Hierunter sind Gebühren für die Teilnahme an einem Kurs zu verstehen. Die Höhe der Gebühr hängt von der Art des Kurses und der mit dem Kurs verbundenen „Credits“, d.h. Kreditpunkte, ab.
3. Weitere Kursgebühren fallen nach dem Lernmodus im Kurs an. Je nach Kurs kann dieser durchgeführt werden als:
  - *Online Group*, d.h. mehrere Teilnehmer im Kurs arbeiten gemeinsam an Themen und interagieren mit einem Betreuer;
  - *Online Individual*, nur ein Studierender interagiert mit seinem „Instructor“, dabei muss der Kurs entweder nach sechs Monaten abgeschlossen sein, wobei der Startzeitpunkt durch den Studierenden selbst wählbar ist (Online Individual: Six Months) oder der Kurs läuft während eines Semesters (Online Individual: One Semester);
  - *Independent Learning*-Kurs, der im Wesentlichen dem Online Individual: Six Months-Kurs entspricht, wobei ein Webzugang nicht erforderlich ist.
4. Neben den Kursgebühren (2. und 3.) fällt jährlich eine *Information Technology Fee* an, die für die Nutzung der IT-Infrastruktur erhoben wird.

Abhängig von Aktionen des Lernenden können weitere Gebühren erhoben werden. Hierzu zählen:

5. Gebühren für verspätete Kurs-Anmeldung und -Abmeldung
6. Gebühren für die Zulassung zu Abschlussprüfungen und
7. Gebühren für die Nichteinlösung von Schecks.

Zur Unterstützung von bedürftigen und hochbegabten Studierenden bietet der PWC eine Studierendenberatung an, die über die Teilnahme an Stipendienprogrammen und die Finanzierung des Studiums berät.

### 2.5.5.3 Preismodell

Studierende, die an einem Kurs teilnehmen möchten, können nur durch die Wahl des Kurstyps den zu entrichtenden Preis beeinflussen (Produktdifferenzierung). Der Preis für einen Kurs setzt sich aus den in Abbildung 13 dargestellten Einzelpreisen zusammen. Neben der Tuition Fee (Unterrichtsgebühr), die an die



Anzahl der Kreditpunkte eines Kurses gekoppelt ist und deren Höhe nach dem Studienabschluss gerichtet ist, werden Gebühren für unterschiedliche Arten der Online-Nutzung von Veranstaltungen, z. B. Online Groups, festgesetzt. Für die Anmeldung zu einem Studiengang wird eine Application Fee fällig. Auch für die Nutzung der technischen Infrastruktur wird eine jährliche Gebühr erhoben.

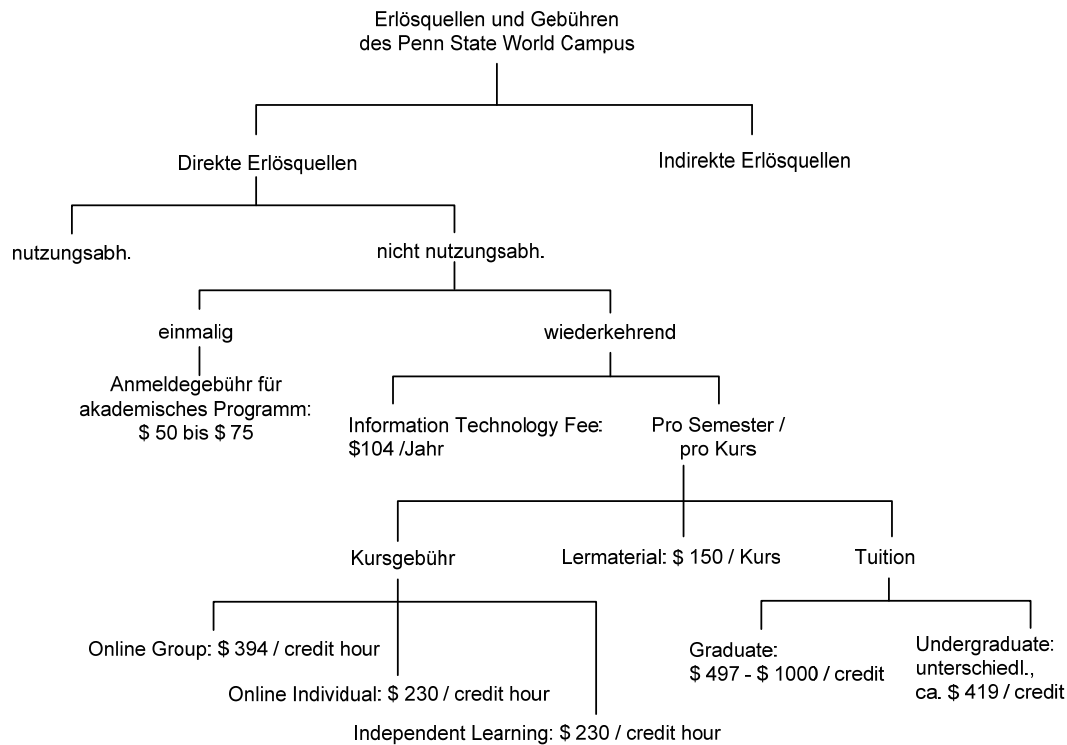
Die Teilnahme an einem Graduate-Kurs mit drei Kreditpunkten würde, ohne Berücksichtigung der anteiligen Information Technology Fee und der anteiligen Anmeldegebühr zum Studiengang, für einen Studierenden minimal  $3 \times \text{US\$ } 497,00$  (Tuition) +  $3 \times \text{US\$ } 394,00$  (Online Group) +  $\text{US\$ } 150,00$  (Lernmaterial) =  $\text{US\$ } 2.823,00$  kosten.

Das vorgestellte Preismodell umfasst folgende Implementationsformen der Produkt-/Preisdifferenzierung:

- Die Gestaltung unterschiedlicher Kurstypen fällt unter die Produktdifferenzierung.
- Sofern für einen Kurstyp unterschiedliche Preise abhängig z. B. von der Art der Online-Nutzung durch die Studierenden gefordert werden, handelt es sich um eine leistungsbezogene Preisdifferenzierung.<sup>47</sup>
- Eine gruppenbezogene Preisdifferenzierung erfolgt durch die Zuordnung der Lernenden zu unterschiedlichen Gruppen (Studierende, Army-Bedienstete etc.). Eine gruppenbezogene Differenzierung ist auch in der Preisdifferenzierung nach Studienfortschritt zu sehen.

---

<sup>47</sup> Auch als qualitative Preisdifferenzierung bezeichnet. Die leistungsorientierte Preisdifferenzierung „steht in enger Beziehung zur Produktlinienpolitik“. (Skiera und Spann 2002), S. 277-278.



**Abbildung 13: Erlösquellen und Preismodell des Penn State World Campus**

## 2.5.6 Beispiele für virtuelle Hochschulen und Studiengänge in Europa

Im Folgenden werden ausgewählte virtuelle Hochschulen und Studiengänge vorgestellt. Eine Auflistung von Förderprogrammen, virtuellen Universitäten sowie Suchmaschinen zu virtuellen Studiengängen und Universitäten ist im Anhang A1 auf Seite 196 zu finden.

### 2.5.6.1 European Online-Academy

Die European Online-Academy (EOA, [www.eu-online-academy.org](http://www.eu-online-academy.org)) wurde 2003 gegründet und bietet einen Online-Weiterbildungsstudiengang zu Integrationsfragen in Europa an, der mit einem „Certificate in European Studies“ abschließt. Die EOA wird durch die Europäische Union finanziell unterstützt. Zu den Zielgruppen gehören Angehörige von Ministerien, aus der öffentlichen und privaten Verwaltung und aus dem Forschungsumfeld. Der Studiengang findet in Online-Kursen und Präsenzphasen statt. Die Kursgebühr für die Nutzung der Online-Kurse und Teilnahme an den Präsenzphasen beträgt inkl. Übernachtungsentgelten € 3.000,00 pro Trimester.

### **2.5.6.2 Fernfachhochschule Schweiz**

Die Fernfachhochschule Schweiz (FFHS, [www.fernfachhochschule.ch](http://www.fernfachhochschule.ch)) bietet fünf Master- und vier Bachelor-Studiengänge in wirtschaftswissenschaftlichen und technischen Studienrichtungen an. Das Studium erfolgt weitestgehend online. Studierende werden einem Studienort zugewiesen, an dem sie sich mit Kommilitonen austauschen können. Für die Studierenden fallen Semestergebühren an, die nach Art des Studienprogramms zwischen CHF 1.800,00 und CHF 6.200,00 liegen. Hinzu kommen Gebühren für die Einschreibung, Nachprüfungen, das Fernbleiben von Prüfungen, Wiederholungsprüfungen und Anmelderückzug.

### **2.5.6.3 Virtual Global University und Winfoline**

Die Virtual Global University – School of Business Informatics (VGU, [www.vg-u.de](http://www.vg-u.de)) bietet einen Studiengang zum Master of Business Informatics (MBI) und Certificate Courses an. Die VGU ist eine virtuelle Universität, bei der sämtliche Kurse online stattfinden. Lediglich zur Abschlussprüfung treffen sich Betreuer und Studierende persönlich. Der Studiengang ist durch das Land Sachsen akkreditiert; Studierende erhalten bei erfolgreichem Studienabschluss ein Zertifikat der Europa-Universität Viadrina Frankfurt (Oder). Das englische Kursangebot wird von Lehrstuhlinhabern europäischer, deutschsprachiger Universitäten betreut und richtet sich an Interessenten, die zumeist einen berufsbegleitenden Abschluss anstreben. Studierende können das Studium in Vollzeit oder Teilzeit absolvieren. Die Tuition Fee beträgt für Vollzeitstudierende € 2.150,00 pro Semester. Teilzeitstudierende zahlen € 550,00 für den ersten Kurs im Semester und € 430,00 für jeden weiteren Kurs. Die Belegung von Zertifikatskursen kostet € 600,00 für den ersten Kurs eines Semesters, für jeden weiteren Kurs werden 20 % Rabatt gewährt. Studierenden aus „low standard-of-living countries“ werden „special rates“ gewährt.

Ein analoges Ausbildungskonzept verfolgt Winfoline-Konsortium der Universitäten Göttingen, Saarbrücken, Kassel und Leipzig, das einen Master of Science in Information Systems anbietet. Die Gebühren für das gesamte Studium betragen ca. € 12.500,00.

### 2.5.7 Herausforderungen bei der Umsetzung von E-Learning

Trotz der in Abschnitt 2.5.4 dargestellten Vorteile und Erfahrungen der Universitäten in der traditionellen Lehre werden diese bisher nur ungenügend für eigene E-Learning-Angebote genutzt. Dabei bieten politische Rahmenbedingungen zurzeit ideale Voraussetzungen zur Etablierung von E-Learning-Konzepten.<sup>48</sup> Die Forschungsförderung auf Ebene der EU (KOM 2002) ([www.elearningeuropa.info](http://www.elearningeuropa.info)), auf Ebene der Bundesländer (z. B. [www.e-learning-hessen.de](http://www.e-learning-hessen.de)) und universitäre Initiativen (z. B. [www.megadigitale.uni-frankfurt.de](http://www.megadigitale.uni-frankfurt.de)) stellen Mittel zur Förderung von E-Learning-Projekten bereit. Projekte wie der Baltic Sea Virtual Campus ([www.bsvc.org](http://www.bsvc.org)) und das BMBF-Leitprojekt „Virtuelle Fachhochschule“ (Oncampus, [www.oncampus.de](http://www.oncampus.de)) sollen zeigen, dass eine europäische bzw. regionale Zusammenarbeit bei der Durchführung von E-Learning-Angeboten möglich ist.

Die in den Kapiteln 2.5.5.3 und 2.5.6.1-2.5.6.3 vorgestellten Tarife für Kurse im E-Learning sind einfach gestaltet; der Preis für einen Kurs ist i. d. R. an die Anzahl der Kreditpunkte gekoppelt. Für Aktionen der Studierenden und Lernmaterial werden zusätzliche Gebühren erhoben. Studierende haben bisher keine Möglichkeit, nur Teile eines Kurses zu belegen, obwohl die individuelle Zusammenstellung von Kursen für institutionelle Kunden z. B. beim Penn State World Campus (s. Abschnitt 2.5.5) möglich ist. Erschwerend für die Preisgestaltung ist, dass eine Kostenbewertung der Bildungsangebote von Lehrveranstaltungen an europäischen Hochschulen z. B. im Rahmen einer Prozesskostenrechnung (Ernest&Young 2000; Bacsich u.a. 2001) meist nicht erfolgt und somit die Basis für ein *Cost-Plus-Pricing* und für ein Bildungscontrolling für E-Learning-Kurse fehlt. Auch für die wertbasierte Bepreisung (*Value-Based-Pricing*) der E-Learning-Angebote aus Kundensicht fehlt es an repräsentativen Erhebungen zur Zahlungsbereitschaft der Lernenden.<sup>49</sup>

---

<sup>48</sup> Es wurde bspw. ein europäisches Kreditpunktesystem (ECTS) im Rahmen des Bologna-Prozesses etabliert.

<sup>49</sup> Eine erste Erhebung der Zahlungsbereitschaft unter Studierenden für E-Learning-Kurse, die zusätzlich zu traditionellen Präsenzlehrformen angeboten werden, wurde im Rahmen einer Diplomarbeit an der Universität Frankfurt durchgeführt (Durmann 2006).

## **2.6 Zusammenfassende Betrachtung**

Das Ziel des zweiten Kapitels lag in der Darstellung von Vermarktungsstrategien für digitale Produkte im E-Learning (Kapitel 2.5). Dazu wurden als Grundlage unterschiedliche Arten (Kapitel 2.2, 2.5.1, 2.5.2) und Eigenschaften digitaler Produkte (Kapitel 2.3), die im E-Learning gehandelt werden, voneinander abgegrenzt. Zudem wurden in der Marketingliteratur gängige Differenzierungsstrategien beschrieben, die von Anbietern im E-Learning als Mittel zur Erlösgenerierung eingesetzt werden können.

Digitale Produkte im E-Learning können mittels Produktmodellen (Kapitel 2.5.5.1) beschrieben werden. Diese bilden die Basis für eine zweistufige „Erlösentscheidung“ eines E-Learning-Anbieters. In einer ersten Stufe werden die Erlösformen / -quellen identifiziert und in einem Erlösmodell (Kapitel 2.5.5.2) festgehalten. Erlösquellen lassen sich u.a. an den Parametern der Produkte festmachen (vgl. Eigenschaften digitaler Produkte in Kapitel 2.3). In einer zweiten Stufe können aus den Erlösmodellen durch Zuordnung von Preisschemas Preismodelle entwickelt werden (Kapitel 2.5.5.3).

Das zweite Kapitel versucht durch die strukturelle Beschreibung des „Aufsetzens“ von Preismodellen für E-Learning ein Instrumentarium zu schaffen, das E-Learning-Anbieter nutzen können, um Bepreisungsmöglichkeiten im E-Learning zu identifizieren, diese zu beschreiben und mit Hilfe der in den folgenden Kapiteln der Arbeit definierten Beschreibungssprache umzusetzen.

### 3 Ablaufmodellierung mit Petri-Netzen

Unter *Petri-Netzen* (Petri 1962) wird eine Klasse formaler und grafischer Beschreibungssprachen zur Modellierung von Prozessen in Informationssystemen (Ablaufmodellierung) verstanden<sup>50</sup>. Petri-Netze fassen als Begriff eine Vielzahl von Systemmodellen, Analysetechniken, grafischen Darstellungsvarianten und Notationen zusammen, die auf spezifischen Annahmen über die Art der Informationsverarbeitung in Systemen beruhen (Reisig und Rozenberg 1998), S. 1. Das Verhalten eines Systems wird in Petri-Netzen durch statische und dynamische Komponenten abgebildet. Durch das so genannte *Schalten* dynamischer Komponenten (Aktivitäten) wird der Zustand eines Petri-Netzes, der sich aus der Menge der Zustände seiner statischen Komponenten ergibt, in einen neuen Zustand überführt. Petri-Netze eignen sich aufgrund ihrer formalen Definition und grafischen Modellelemente insbesondere zur Erstellung von Systementwürfen in Teams. Durch die Verwendung unterschiedlicher Petri-Netztypen im Entwurf kann eine schrittweise Entwicklung von Prozessmodellen von einfachen, informalen Netzen hin zu Netzen, die eine Grundlage für die spätere Implementierung bilden, erreicht werden (Reisig 1985). Petri-Netze modellieren das Verhalten verteilter Systeme, bei denen Systemkomponenten lokal getrennt vorliegen können. Sie eignen sich daher insbesondere zur Darstellung von Prozessen in Lern-Managementsystemen (vgl. 2.5.2).

Petri-Netze können in *zwei grundlegende Typen* unterschieden werden. *Elementare Netze* (Low-level Nets), z. B. Bedingungs/Ereignis-Netze und Stellen/Transitionen-Netze erlauben eine einfache Interpretation der Netzkomponenten und sind daher leicht zu verstehen, aber im Falle der Spezifikation großer Systeme schwer zu verwalten (Oberweis und Sander 1996; Rozenberg und Engelfriet 1998). *Höhere Petri-Netztypen*, z. B. Prädikate/Transitionen-Netze (Genrich und Lautenbach 1981) und gefärbte Petri-Netze, vereinfachen die Darstellung komplexer Systeme durch ausdrucksstärkere Netzkomponenten. Insbesondere kann dadurch eine kompaktere Darstellung von

---

<sup>50</sup> Eine Übersicht über die Charakteristika verschiedener Petri-Netz-Klassen geben beispielsweise (Desel und Oberweis 1996), (Desel und Juhás 2001) und (Murata 1989).

Systemen als in elementaren Petri-Netzen erreicht werden.<sup>51</sup> Einen Überblick über höhere Petri-Netztypen geben z. B. (Gerogiannis u.a. 1997).

Die in Kapitel 3.3 vorgestellten CP-Netze eignen sich hervorragend für die Darstellung und Analyse von Prozessen, z. B. im Rahmen des Software-Engineerings. Neben der Strukturierung des Vorgehens beim Entwurf eines Software-Systems können sie zur Darstellung von internen Funktionsweise und der Schnittstellen eines Software-Systems und in speziellen Anwendungsgebieten auch zur Erzeugung von Programmcode verwendet werden. Ein wesentlicher Vorteil von CP-Netzen ist die Verfügbarkeit eines Modellierungswerkzeuges, mit dem CP-Netze erstellt und Eigenschaften der erstellten Netze untersucht werden können. Die Vorteile von Petri-Netzen im Software Engineering werden im Anhang „A.3 Petri-Netze und Software Engineering“ ab Seite 206 beschrieben. Gefärbte Petri-Netze werden in den Kapiteln 6 und 7 zur Beschreibung der Funktionsweise des Software-Prototyps verwendet.

### 3.1 Elementare Petri-Netze

Elementare Petri-Netze geben eine allgemeine Definition der Komponenten, die auch Höheren Petri-Netzen zugrunde liegt (Rozenberg und Engelfriet 1998). Ein Elementares Petri-Netz ist ein *gerichteter, bipartiter Graph mit zwei disjunkten Knotenmengen*, die als Stellen und Transitionen interpretiert werden, bei denen keine gerichtete Kante Knoten des gleichen Typs verbindet (Desel und Juhás 2001).<sup>52</sup>

#### 3.1.1 Formale Definition

Die folgende Definition von Elementaren Petri-Netzen orientiert sich an (Genrich und Lautenbach 1981; Rozenberg und Engelfriet 1998).

Ein Tupel  $N = (S, T, F)$  heißt *gerichtetes Netz*, falls gilt:

$$S \cap T = \emptyset,$$

$$S \cup T \neq \emptyset \text{ und}$$

$$F \subseteq (S \times T) \cup (T \times S).$$

---

<sup>51</sup> In der Petri-Netz-Domäne existieren zwei unterschiedliche Denkweisen. Nach (Girault und Valk 2003), S.81 sehen Vertreter des „direkten Ansatzes“ Petri-Netze als hilfreiche, benutzerfreundliche Methode zur Erzeugung grafischer Modelle an. Dagegen wird im „indirekten Ansatz“ die Meinung vertreten, dass Petri-Netze für einen geübten Anwender vor allem einen mächtigen Low-Level-Formalismus darstellen und damit zur Überprüfung anderer Modelle geeignet sind.

<sup>52</sup> (Rozenberg und Engelfriet 1998) fordern zusätzlich, dass T-Elemente (Transitionen) niemals isoliert auftreten dürfen.  $N$  kann auch das leere Netz mit  $N = (\emptyset, \emptyset; \emptyset)$  sein.

Für ein gegebenes Netz  $N = (S, T, F)$  sei

$X := S \cup T$  die endliche Menge der  $S$ - oder  $T$ -Elemente von  $N$  und

$F$  die Flussrelation, die die gerichteten Kanten von  $N$  umfasst.

Für ein Element  $x \in X$  sei

$\bullet x := y \mid (y, x) \in F$  der *Vorbereich* (Pre-Set) bzw.  $x \bullet := y \mid (x, y) \in F$  der

*Nachbereich* (Post-Set) von  $x$

Vorbereich und Nachbereich von  $x$  ergeben zusammen dessen *Umgebung*  $\bullet x \cup x \bullet$ , die auch mit  $\mathbf{nbh}(x)$  bezeichnet wird.<sup>53</sup>

### 3.1.2 Grafische Darstellung

In Abbildung 14 ist ein Elementares Petri-Netz abgebildet.  $S$ -Elemente (*Stellen*), werden durch Kreise,  $T$ -Elemente (*Transitionen*) durch Rechtecke und *gerichtete Kanten*  $(x, y) \in F$  durch Pfeile, die vom grafischen Symbol von  $x$  zum grafischen Symbol von  $y$  führen, dargestellt. Stellen und Transitionen eines Netzes werden durch Angabe von Beschriftungen (Labels) bezeichnet. Stellen, als statische Elemente eines Netzes, können außerdem markiert werden, indem eine *Marke* (Token) in der Stelle gesetzt wird. Im Beispiel ist die Stelle  $s_1$  markiert. Stellen dienen als Behälter für Marken und ermöglichen mittels Durchführung von *Markenspielen* die Simulation des Verhaltens eines einfachen Systems.<sup>54</sup>

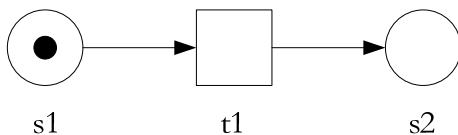


Abbildung 14: Elementares Petri-Netz

Die durch Rechtecke dargestellten Transitionen repräsentieren Veränderungsoperationen. Eine *Transition kann schalten*, d.h. ausgeführt werden, falls alle Vorbedingungen erfüllt, d.h. der Vorbereich belegt ist, und sämtliche Nachbedingungen nicht erfüllt sind (Nachbereich nicht belegt). Wenn eine Transition  $t$  schaltet, werden alle Vorbedingungen auf ungültig und alle Nachbedingungen auf gültig gesetzt. In der grafischen Repräsentation werden dazu alle Marken aus dem Vorbereich der Transition  $t_1$  gelöscht (im Netz aus Abbildung 14 die Marke aus  $s_1$ ) und im Nachbereich (in der Stelle  $s_2$ ) gesetzt.

<sup>53</sup> Die Abkürzung  $\mathbf{nbh}$  steht für engl. "Neighbourhood".

<sup>54</sup> In der Literatur wird zwischen Netzen und Netzsystemen unterschieden. Nach (Rozenberg und Engelfriet 1998) umfassen Netzsysteme neben dem Netz  $N$  zusätzlich eine Ausgangsmarkierung,



### 3.1.3 Strukturelle und dynamische Analyse von Petri-Netzen

Petri-Netze können hinsichtlich ihrer *Struktur* und ihrer *Dynamik* (Verhalten) untersucht werden. Die Struktur des Netzes wird durch die Verkettung von Stellen und Transitionen durch gerichtete Kanten definiert. Hierzu gehören auch z.B. unterschiedliche Arten von Verzweigungen.

Im Rahmen der strukturellen Analyse können bereits Netzteile (z.B. isolierte Stellen) identifiziert werden, die bei der Ausführung eines Netzes zu Problemen führen können. Eine strukturelle Betrachtung von Netzen ist insbesondere auch dann wichtig, wenn Netze in „Module“, so genannte stellen- und transitionsberandete Teilnetze, aufgeteilt werden sollen, um bspw. einen durch ein Netz beschriebenen Ablauf in Teilabläufe zu unterteilen.

Die *Dynamische Analyse* widmet sich der Untersuchung von Schaltvorgängen in Netzen. Dynamik entsteht durch die Abfolge von „Markierungen“<sup>55</sup> eines Netzes, die durch Schaltvorgänge von Transitionen entstehen. Erst durch die Belegung eines Netzes mit Marken und die Durchführung von Schaltvorgängen, können Probleme wie z.B. Verklemmungen (Deadlocks) und Konflikte sowie Kausalitäten und Nebenläufigkeiten in einem Netz identifiziert werden.

Eine ausführliche Darstellung der dynamischen Eigenschaften von Elementaren Petri-Netzen befindet sich im Anhang A.3.2 Dynamische Eigenschaften von Petri-Netzen.

## 3.2 Stellen/Transitionen-Netze

Grundlegende Konzepte von Petri-Netzen werden anhand der in Abschnitt 3.1 eingeführten Elementaren Petri-Netze in Anhang „A.3.1 Struktureigenschaften von Petri-Netzen“ und „A.3.2 Dynamische Eigenschaften von Petri-Netzen“ beschrieben. Da Elementare Petri-Netze zur Darstellung komplexer Systeme weniger geeignet scheinen, wurden diese in den 1970er Jahren zu *Stellen/Transitionen-Netzen* (Reisig 1986b; Desel und Reisig 1998) weiterentwickelt.<sup>56</sup> Stellen/Transitionen-Netze erlauben ebenso das Markenspiel

---

in der für alle Stellen angegeben wird, ob diese markiert sind. In der Arbeit werden Netze und Netzsysteme vereinfachend unter dem Begriff Netz betrachtet.

<sup>55</sup> Im Falle Elementarer Petri-Netze beschreibt eine *Markierung* als Untermenge von  $S$  die Gesamtheit aller gleichzeitig markierten Stellen.

<sup>56</sup> Es existieren zwei Auffassungen: Zum einen können S/T-Netze als Generalisierung von Elementaren Petri-Netze angesehen werden. Andererseits können S/T-Netze als „Kurzschreibweise“ und Abstraktion von Elementaren Petri-Netzen betrachtet werden, bei der

mit ununterscheidbaren Marken, können aber im Gegensatz zu Elementaren Petri-Netzen über mehrere Marken in den Stellen verfügen. Eine Darstellung von Stellen/Transitionen-Netzen findet sich in Anhang A.3.3 Stellen/Transitionen-Netze. Ein Stellen/Transitionen-Netz, dessen Stellen alle als Bedingungen interpretiert werden können, entspricht einem Bedingungs/Ereignis-Netz (Desel und Oberweis 1996). Obwohl Stellen/Transitionen-Netze komplexer als Elementare Petri-Netze sind, eignen sich diese nicht zur Modellierung von Software-Systemen. Ein wesentlicher Nachteil besteht darin, dass Marken als Objekte nicht unterscheidbar sind. In den folgenden Kapiteln werden Höhere Petri-Netze beschrieben, die die Definition individueller Objekte erlauben.

### 3.3 Gefärbte Petri-Netze

#### 3.3.1 Überblick

Inspiziert durch die von Genrich entwickelten Pr/T-Netze wurde von Jensen eine High-Level-Petri-Netz-Variante, die so genannten *gefärbten Petri-Netze* (*Coloured Petri Nets = CP-Netze*) (Jensen 1981, 1986; Jensen und Rozenberg 1991; Jensen 1994, 1996b, 1997), definiert.<sup>57</sup> Gefärbte Petri-Netze können als Weiterentwicklung von Pr/T-Netzen betrachtet werden. Sie unterscheiden sich von Pr/T-Netzen dadurch, dass nach (Jensen 1986)

- die *Stellentypisierung* explizit angegeben wird,
- die Anzahl der Token, die durch Schalten einer Stelle hinzugefügt oder aus dieser gelöscht werden, für zwei „*Occurence-Colours*“ unterschiedlich sein kann und
- die *Menge der zulässigen Ausdrücke und Prädikate nicht explizit definiert* ist.

Statt der Verwendung von prädikatelogischen Ausdrücken zur Definition von Schaltregeln, zur Belegung der Stellen mit Marken sowie zur Beschriftung der Elemente eines Netzes werden in gefärbten Petri-Netzen so genannte *Funktionen* im Sinne der funktionalen Programmierung verwendet. Funktionen werden mittels *CPN ML*, einer funktionalen, auf Standard ML (SML;

---

mehrere Bedingungen / Stellen zu einer Stelle und mehrere Ereignisse / Transitionen zu einer Transition zusammengefasst werden. (Reisig 1986b), S. 118.

<sup>57</sup> In der Literatur wird der Begriff „High-Level-Netz“ oft mit gefärbten Petri-Netzen gleichgesetzt, z. B. bei (Hummert 1989).

<http://www.smlnj.org>) basierender Programmiersprache, definiert.<sup>58</sup> Die Entwicklung von *CPN Tools* (Ratzer u.a. 2003), einem Editor und Analysewerkzeug für gefärbte Petri-Netze, hat dazu geführt, dass gefärbte Petri-Netze sowohl in der Wissenschaft, als auch in der Praxis weithin verbreitet sind.<sup>59</sup> Insbesondere finden gefärbte Petri-Netze Anwendung in der Spezifikation von Protokollen und Embedded-Systems, als auch im Software Engineering (Mortensen 1997; Kristensen u.a. 2003). Erweiterungen erlauben die Verfeinerung von CP-Netzen durch *hierarchische Dekomposition* von Transitionen und die Integration von Zeitaspekten.

### 3.3.2 Definition

Ein Tupel  $CPN = (\Sigma, P, T, A, N, C, G, E, I)$  heißt *Gefärbtes Petri-Netz* bzw. *Coloured Petri Net*, wenn gilt:

- (1)  $\Sigma$  ist eine endliche Menge nicht-leerer Typen (Types), auch *Colour-Sets*<sup>60</sup> genannt.
- (2)  $P$  ist die *Menge aller Stellen*.
- (3)  $T$  ist die *Menge aller Transitionen*.  
(2) und (3) bilden die Knoten des Netzes.
- (4)  $A$  ist die *Menge der Kanten* (Arcs), für die gilt:  
 $P \cap T = P \cap A = T \cap A = \emptyset$ .
- (5)  $N$  ist die *Knoten-Funktion*, die jede Kante mit einem Paar von Knoten verbindet, wobei der erste Knoten des Paares den Ursprung der Kante, der zweite den Zielknoten der Kante bildet.  $N : A \rightarrow P \times T \cup T \times P$ .
- (6)  $C$ , die *Colour-Funktion*, definiert zu jeder Stelle  $p$  einen zugehörigen Typ (Colour-Domain) im Sinne eines Datentyps. Datentypen entsprechen den Datentypen in Standard ML (Gansner und Reppy 2004):  
 $C : P \rightarrow \Sigma$

---

<sup>58</sup> Der Vorwurf von (Hummert 1989), S. 2-3 „Algebraische Netzspezifikationen sind übersichtlicher als [Gefärbte Petri-Netze], da die Funktionalität von Transitionen den mit Termen beschrifteten Kanten entnommen werden kann.“, ist unzutreffend, da gefärbte Petri-Netze auf einen getypten Lambda-Kalkül zurückgeführt werden können (und somit über eine algebraische Spezifikation verfügen).

<sup>59</sup> Die aktuelle Version von CPN Tools ist unter <http://wiki.daimi.au.dk/cpntools/cpntools.wiki> zu finden.

<sup>60</sup> Der Begriff der Farbe / Colour ist an die Physik angelehnt. Eine Farbmenge / Colour-Set kann unterschiedliche Farben umfassen. Analog dazu wird der Begriff eines Typs in Standard ML verwendet. Zu einem Typ gehören unterschiedliche Instanzen. In CPN ML wird ein eigener Typ für Colour-Sets definiert.

- (7)  $G$ , die *Guard-Function*, überwacht, dass in allen Transitionen  $t$  die Variablen einen Typ aus  $\Sigma$  haben. Der Typ einer Variablen kann durch die Funktion  $Type$  ermittelt werden. Bei der Programm-basierten Erstellung von CPN mit CPN Tools wird die Bedingung, dass sämtliche Transitionen einen Typ haben müssen, automatisch überprüft.

$$\forall t \in T : [Type(G(t)) = Boolean \wedge Type(Var(G(t))) \subseteq \Sigma] ,$$

wobei  $Var(ausdr)$  die Menge der Variablen in einem Ausdruck angibt. Es ist daher unerheblich, ob in der obigen Vorschrift statt  $Var(G(t))$  lediglich  $Var(t)$  aufgeführt wäre.

- (8)  $E$ , die *Arc-Expression-Function*, verknüpft jede Kante  $a$  mit einem Ausdruck des Typs  $C(p)_{MS}$ . Dies bedeutet, dass die Auswertung einer jeden Kante eine Multimenge von Marken/Farben des Typs der adjazenten<sup>61</sup> Stelle  $p$  ergeben muss und dass die Typen der Variablen in der Arc-Expression aus der Menge der zulässigen Colour-Sets des Netzes stammen.

$$\forall a \in A : [Type(E(a)) = C(p)_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$$

- (9)  $I$ , die *Initialisation-Function* definiert zu jeder Stelle  $p$  eine zugehörige Markierung, d.h. eine Multimenge von Marken des Typs  $C(p)_{MS}$ . Dies entspricht der Anfangsmarkierung  $M_0$  im Falle der Pr/T-Netze.

$$\forall p \in P : [Type(I(p)) = C(p)_{MS}]$$

Für alle Transitionen  $t$  aus  $T$  und alle Paare von Knoten  $(x_1, x_2) \in (P \times T \cup T \times P)$  gilt:

- (1)  $A(t) = \{a \in A \mid N(a) \in P \times \{t\} \cup \{t\} \times P\}$  als die Menge aller adjazenten Kanten zu  $t$ . (s. auch unter (3))
- (2)  $Var(t) = \{v \mid v \in Var(G(t)) \vee \exists a \in A(t) : v \in Var(E(a))\}$   
 $Var(t)$  umfasst alle Variablen, die in der Variablenmenge der Guard-Function einer Transition  $t$  oder in den ein- und ausgehenden Kanten der Transition  $t$  vorkommen.
- (3)  $A(x_1, x_2) = \{a \in A \mid N(a) = (x_1, x_2)\}$  ist die Menge aller Kanten zwischen den Knoten  $x_1$  und  $x_2$ . Die Funktion  $A$  ist „polymorph“ (vgl. Kapitel 0).

---

<sup>61</sup> Adjazent = angrenzend. Es handelt sich um die Stelle  $p$ , wenn  $a = (p, t)$  bzw.  $a = (t, p)$ .

Neben der hier zwei-parametrischen Funktionsdefinition ist  $A$  auch auf Transitionen  $t$  als Input-Parameter definiert (s. unter (1)). Dabei werden dann bei der Auswertung von  $A$  sämtliche Kanten betrachtet, die in eine Transition münden ( $P \times \{t\}$ ) bzw. von einer Transition ausgehen ( $\{t\} \times P$ ).

- (4)  $E(x_1, x_2) = \sum_{a \in A(x_1, x_2)} E(a)$  umfasst alle Ausdrücke an den Kanten, die  $x_1$  und  $x_2$  verbinden.

Ein Binding weist einer Variablen einen Wert zu (vgl. Kap. 3.3.5.1). Eine *Belegung* (ein *Binding*) einer Transition  $t$  sei eine Funktion  $b$ , die auf  $Var(t)$  wie folgt definiert ist:

- (1)  $\forall v \in Var(t) : b(v) \in Type(v)$ : Für alle Variablen  $v$  gilt, dass die Variablenbelegung (Wert der Variablen) dem Typ der Variablen genügen muss.
- (2)  $G(t) < b >$ : Auswertung der Guard-Function bei einem Binding  $b$ .
- (3)  $B(t)$ : Menge aller Bindings zu  $t$ .

Die Darstellung von Bindings erfolgt analog zur Darstellung von *Records* (einem speziellen Listentyp) in Standard ML. Gegeben eine Variablenmenge  $Var(t) = \{v_1, v_2, \dots, v_n\}$  kann ein mögliches Binding z. B. wie folgt aussehen:  $< v_1 = c_1, v_2 = c_2, \dots, v_n = c_n >$  mit  $c_i \in C(p)$ .

Ein *Token-Element* ist definiert als ein Tupel  $(p, c)$  mit  $p \in P$  und  $c \in C(p)$ .

Ein *Binding-Element* hingegen ist ein Tupel  $(t, b)$  mit  $t \in T$  und  $b \in B(t)$ .

Bindings sind also nur für Transitionen definiert.

Die Menge aller Token-Elemente wird mit  $TE$ , die Menge aller Binding-Elemente mit  $BE$  gekennzeichnet.

Eine *Markierung* ist eine *Multimenge* über  $TE$ .

Ein *Schritt* (Step) ist eine nicht-leere, endliche Multimenge über  $BE$ . D.h. in einem Schritt können mehrere Transitionen schalten.

Die *Ausgangsmarkierung* (*Initial-Marking*)  $M_0$  ist die Markierung, die durch Auswertung der *Initialisation-Expression* ermittelt werden kann. Die Ausgangsmarkierung wird für jede einzelne Farbe einer Stelle  $p$  evaluiert.

(  $\forall(p,c) \in TE : M_0(p,c) = (I(p))(c)$  )<sup>62</sup>.

Auch die Markierung eines Netzes kann als Funktion  $M^*$  über alle  $p \in P$  dargestellt werden. Es gilt:  $\forall(p,c) \in TE : M(p,c) = (M^*(p))(c)$ <sup>63</sup> mit dem Wertebereich  $C(p)_{MS}$  für  $M^*$ .

### 3.3.3 Eigenschaften

Ein *Schritt*  $Y$  ist genau dann bei gegebener Markierung  $M$  aktiviert bzw. kann schalten, wenn  $\forall p \in P : \sum_{(t,b) \in Y} E(p,t) \langle b \rangle \leq M(p)$  erfüllt ist. Dabei umfasst

$E(p,t) \langle b \rangle$  die Token, die aus  $p$  entfernt werden, wenn  $t$  unter  $b$  schaltet. Durch die Summe über alle Binding-Elemente erhalten wir alle Token, die aus  $p$  entfernt werden, wenn  $Y$  schaltet. Die Transition  $t$  ist dann unter  $b$  aktiviert, kurz  $(t,b)$  ist aktiviert. Und alle Elemente aus  $Y$  sind *nebenläufig* aktiviert.

Durch das Schalten von  $Y$  wird eine gegebene Markierung  $M_1$  in die Folgemarkierung  $M_2$  überführt. Es gilt:

$$\forall p \in P : M_2(p) = (M_1(p) - \sum_{(t,b) \in Y} E(p,t) \langle b \rangle) + \sum_{(t,b) \in Y} E(t,p) \langle b \rangle.$$

$M_2$  ist die Folgemarkierung von  $M_1$ , die durch Schalten von  $Y$  entsteht, kurz  $M_1[Y]M_2$ .

$[M_0]$  ist die Menge der *Folgemarkierungen* zu  $M_0$ , die durch Schaltvorgänge erreicht werden. Eine Markierung ist dann erreichbar, wenn sie zur Menge  $[M_0]$  gehört.

### 3.3.4 Annotation von CP-Netzen mit CPN ML

Zu jedem CP-Netz gehört eine Menge von Deklarationen (s. z. B. Abbildung 17, S. 77), die in der von (Jensen 1998) vorgeschlagenen grafischen Darstellung von

---

<sup>62</sup> Gemäß Jensen (E-Mail vom 9.5.2007): „ $I(p)$  is a multi-set of tokens, while  $(I(p))(c)$  is the coefficient for the colour  $c$  in the multi-set  $I(p)$ . Intuitively, the formula tells that the initial marking  $M_0(p)$  for a marking  $p$  is obtained by evaluating the initialisation expression  $I(p)$  for the place  $p$ .“

<sup>63</sup> Analog Definition zur vorherigen Fußnote.

CP-Netzen in einem an das Netz angrenzenden, gestrichelten Rechteck dargestellt werden.

Im Werkzeug CPN Tools werden die Deklarationen in einer Liste dargestellt. Zu den Deklarationen gehören:

- *Colour-Sets*,
- *Funktionen, Operationen, Variablen und Konstanten*, die für die Darstellung der Netzschriften (Kantenbeschriftungen, Guards, Initialisation-Expressions) verwendet werden, und
- *Markierungen*, die als *Funktionen* oder *Multimengen* (z. B. Listen und Records in Standard ML (s. u.)) dargestellt werden können.

Beispiel: Darstellung einer Markierung als Funktion:

$$M_0(p) = \begin{cases} 3 \cdot (q, 0) & \text{if } p = A \\ 2 \cdot (p, 0) & \text{if } p = B \\ 1 \cdot e & \text{if } p = R \\ 3 \cdot e & \text{if } p = S \\ 2 \cdot e & \text{if } p = T \\ \emptyset & \text{sonst} \end{cases}$$

Beispiel: Alternative Darstellung als Multimenge:

$$M_0 = 3 \cdot (A, (q, 0)) + 2 \cdot (B, (p, 0)) + 1 \cdot (R, e) + 3 \cdot (S, e) + 2 \cdot (T, e)$$

Dabei bedeutet der Ausdruck  $3 \cdot (A, (q, 0))$ , dass in Stelle  $A$ , dreimal ein Paar  $(q, 0)$  als Element vorkommt. Nicht unterscheidbare Marken, wie z. B. Marken in S/T- oder B/E-Netzen werden durch das leere Element  $e$  beschrieben<sup>64</sup>.

Für die Annotation von CP-Netzen wurde auf Basis von Standard ML die Beschreibungssprache *CPN ML* geschaffen, die im folgenden Absatz beschrieben wird.

### 3.3.5 Sprachbestandteile von CPN ML

In diesem Kapitel werden die zum Verständnis von CP-Netzen notwendigen Sprachbestandteile von CPN ML (vgl. Anhang) beschrieben. Hierzu gehören Colour-Sets, Variablen, Funktionen und Konstanten. Aus Gründen der

---

<sup>64</sup> Eine alternative Schreibweise für  $e$  ist das Nulltupel  $\langle \rangle$ . In der Arbeit werden u.a. auch  $t1$ ,  $t2$  für Variablennamen verwendet, die Nulltupel enthalten können.

Übersichtlichkeit werden die Kodierung von Zeitinformationen in CP-Netzen sowie dateigesteuerte Variablendeklarationen nicht beschrieben.<sup>65</sup>

### 3.3.5.1 Variablen

Variablen können nur in Kanteninschriften und *Guards* verwendet werden (vgl. Kapitel 3.3.2). Variablen selbst dürfen in CPN ML nur Colour-Sets als Typen verwenden.

Beispiel: `var x, y, z: AA` definiert die Variablen `x`, `y` und `z` des Typs `AA`.

Der Wert einer Variablen kann während der Ausführung eines CP-Netzes geändert werden.

Beispiel: `var id1, id2, ..., idn : cs_name;`

Erklärung:

`idi` Identifier der Variablen, der stets mit einem Buchstaben beginnen muss.

`cs_name` Name des Colour-Sets.

### 3.3.5.2 Referenzvariablen

Eine Referenzvariable zeigt auf einen Speicherbereich. Referenzvariablen dürfen nur in Codeblöcken verwendet werden. Der Gültigkeitsbereich einer Referenzvariablen entspricht dem CP-Netz. Referenzvariablen dürfen nicht zur Aktivierung einer Transition verwendet werden.

Beispiel: `globref id = exp;`

Erklärung:

`id` Identifier der Referenzvariablen

`exp` Ausdruck

Folgende Funktionen sind für die Verarbeitung von Referenzvariablen in CPN ML vorgesehen:

`!r;` Liefert den Wert der Variablen zurück.

---

<sup>65</sup> Die Beschreibung orientiert sich an (Jensen 1994) und der CPN Tools-Hilfe, die im World-Wide Web unter <http://wiki.daimi.au.dk/cpntools/> abgerufen werden kann.



`r := v;`            Weist der Referenzvariablen `r` die Konstante `v` zu.  
`inc / dec;`        In-/Dekrement einer Integer-Referenzvariable.

### 3.3.5.3 Funktionen

Der Ergebnistyp von Funktionen in CPN ML entspricht entweder einem Colour-Set, der Menge aller Multimengen über ein definiertes Colour-Set oder einem in CPN ML zulässigen Typ. Funktionen können verwendet werden, um

- Colour-Sets zu definieren,
- weitere Funktionen, Operationen (Infix-Funktionen) und Konstanten zu definieren und
- Arc-Expressions, Guards sowie Initialisation-Expressions zu beschreiben.

Der Rückgabe-Typ einer Funktion wird durch den CPN ML-Compiler bestimmt.

#### Beispieldeklaration:

```
fun modernize {make=m, built=year} =
{make m, built = year + 1}
```

Bei Aufruf der Funktion `modernize` wird `built` um 1 erhöht. Dabei setzt die Funktion ein zwei-elementiges Record als Parameter voraus. Wird der Typ der Übergabeparameter mit `pat` (für Pattern) abgekürzt, so kann eine allgemeinere Darstellung von Funktionen wie folgt aussehen:<sup>66</sup>

```
fun id pat1 = exp1
  | id pat2 = exp2
  ...
  | id patn = expn;
```

Bei der Durchführung der Funktion `id` wird der Typ des Übergabeparameters überprüft. Entspricht dieser `pat1`, so gibt die Funktion `exp1` zurück und so weiter (vgl. Harper 2005).

Es muss gewährleistet sein, dass `exp1` bis `expn` jeweils denselben Rückgabetypp haben.

---

<sup>66</sup> Mit „|“ ist ein programmatisches „Oder“ gemeint.

Als Kontrollstrukturen sind in Funktionen bedingte Verzweigungen zulässig. Diese können mittels `if-then-else` oder `case` definiert werden.

Beispiele für Kontrollstrukturen:

```
if bool-exp then exp1 else exp2;
```

```
case exp of
  pat1 => exp1
| pat2  => exp2
  ...
| patn => expn;
```

Hinweis: `exp1` bis `expn` müssen wie oben typkompatibel sein.

### 3.3.5.4 Konstanten

Konstanten können entweder zu einem Colour-Set gehören oder einen in CPN ML zulässigen Typ haben. Sie können zur Deklaration von Funktionen und anderen Ausdrücken in CPN ML verwendet werden. Konstanten werden durch Wertzuweisung definiert.

Beispiel:

```
val string_value = "Ein einfacher Stringwert";
```

Konstanten können auch auf Basis von Colour-Sets und als Multimengen definiert werden.

Beispiel für die Definition einer Konstanten auf Basis eines Colour-Sets:

```
val lastPacket = DATA("#####");
```

Beispiel für eine Konstante, die eine Multimenge enthält:

```
val intMS = 1`2++2`8;67
```

Ergibt eine Multimenge mit folgenden Werten: {2, 8, 8}

---

<sup>67</sup> „++“ bezeichnet den Mengenadditionsoperator, mit dessen Hilfe Tupel im Sinne einer Auflistung zu Multimengen verknüpft werden.

### 3.3.5.5 Ausdrücke / Net-Expressions

Alle in einem CP-Netz aufgeführten Ausdrücke können anhand eines Bindings ausgewertet werden. Ein *Binding* assoziiert dabei zu jeder in einem Ausdruck auftretenden Variable eine Konstante. Auf Basis der Bindings kann der Ausdruck dann ausgewertet werden und liefert einen Wert zurück. Jeder Ausdruck hat einen zugehörigen Typ, der durch die Menge aller möglichen „Auswertungswerte“ definiert ist. Der Typ einer Guard-Expression ist bspw. `Bool`, während der Typ einer Arc- oder Initialisation-Expression dem Wertebereich der angrenzenden Stelle  $p (C(p)_{MS})$  entspricht.

Beispiel	Variablen und Konstanten	Resultatstyp des Ausdrucks	Erläuterung
<code>x = q</code>	x: Variable q: Konstante	<code>Bool</code>	Der Ausdruck ist nur wahr, wenn <code>&lt;x = q&gt;</code> gilt.
<code>(x, i)</code>	x, i: Variablen	<code>P</code>	Die Belegung <code>&lt;x = p, i = 0&gt;</code> erzeugt ein Tupel <code>(p, 0)</code> .
<code>1` (x, i)</code>	x, i: Variablen	Multiset über Tupel	Wie zuvor.
<code>3` (q, 0)</code>	q, 0: Konstante	Multiset über Tupel	Da beide Elemente des Tupels Konstanten sind, folgt stets <code>3` (q, 0)</code> als Belegung.
<code>2`e</code>	e: Konstante	Multiset über e	Analog.
<pre>case x of   p =&gt; 2`e     q =&gt; 1`e</pre>	x: Variable e: Konstante	Multiset über e	Ergibt <code>2`e</code> , wenn <code>x = p</code> , falls <code>x = q</code> , dann <code>1`e</code> .

*Tabelle 4: Beispiele für Ausdrücke und deren Resultatstypen in CPN ML*

### 3.3.5.6 Colour-Sets

#### Definition eines Colour-Sets über vordefinierte Datentypen

Die einfachste Variante der Definition eines Colour-Sets besteht darin, die in der Standard ML Basis Library zur Verfügung stehenden Basis-Datentypen `int`, `real`, `string`, `bool` oder `unit` zu verwenden. Die Definition einer Farbe für

Integer-Werte namens INTCOL mit Hilfe des Typs `int` sieht dann wie folgt in CPN ML aus:

```
colset INTCOL = int .
```

Die Eingrenzung des Colour-Sets auf Untermengen der Datentypen, z. B. nur auf Integer-Zahlen zwischen 1 und 30, ist dabei möglich:

```
colset INTCOL_1_TO_30 = int with 1..30
```

#### Definition eines Colour-Sets durch Anpassung von Datentypen

Als zweite Möglichkeit können Colour-Sets für die Typen `bool` und `unit` auch mit alternativen Namen für die Farben versehen werden. Soll bspw. statt „false“ „no“ und statt „true“ „yes“ verwendet werden, so sieht die Definition eines entsprechenden Colour-Sets wie folgt aus:

```
colset BOOLCOL_NO_YES = bool with (no, yes)
```

Die Deklaration eines Colour-Sets für nicht unterscheidbare Token mit Hilfe des Typs `unit` sieht dann wie folgt aus:

```
colset TOKENCOL = unit with e
```

#### Definition eines Colour-Sets durch Enumeration bzw. Indexierung

Enumeration der im Colour-Set enthaltenen Farben:

```
colset FAMILYCOL = with man | woman | child
```

`colset CARCOL = index car with 3..8` ergibt sechs Farben von `car(3)` bis `car(8)`

#### Definition eines Colour-Sets basierend auf anderen Colour-Sets

Schließlich können Colour-Sets, wie in den folgenden Beispielen dargestellt, auch auf Basis bestehender Colour-Sets definiert werden:

Beispiel	Erläuterung
colset MM = product INTCOL * FAMILYCOL	Alle Tupel $(a, b)$ für die $a \in \text{INTCOL}$ und $b \in \text{FAMILYCOL}$
colset NN = record i:INTCOL * r:INTCOL * <sup>68</sup> s:INTCOL	Alle Records $\{i = a, r = b, s = c\}$ , für die $a, b, c \in \text{INTCOL}$
colset OO = union i1:INTCOL + i2:INTCOL	Alle Werte $i1(a)$ und $i2(b)$ , mit $a, b \in \text{INTCOL}$ . Dabei werden $i1$ und $i2$ <i>Selektoren</i> genannt.
colset PP = list INTCOL	Alle Listen mit Elementen des Wertebereichs INTCOL

**Tabelle 5: Beispiele zur Definition von Colour Sets in CPN ML**

*Stellen können drei Arten von Beschriftungen aufweisen:*

- (1) Das Colour-Set, das den Wertebereich der Stelle definiert.
- (2) Optional eine Ausgangsmarkierung der Stelle, die mit dem zuvor genannten Colour-Set kompatibel ist.
- (3) Optional die Zuweisung eines Namens zu der Stelle.

*Kantenbeschriftungen* bestehen aus CPN ML-Ausdrücken, die entweder bei ihrer Auswertung eine Multimenge oder ein einzelnes Element zurückgeben. Freie Variablen können an Output-Kanten entstehen, wenn diese nicht in einer Input-Kante oder der Guard der vorgelagerten Transition gebunden sind. Freie Variablen entstehen z. B. bei der sukzessiven Definition eines CP-Netzes durch einen Modellersteller, wenn die Transitionsinschrift noch nicht vorhanden ist.

Mit Hilfe von *Transitionsbeschriftungen* werden Transitionen Namen zugewiesen, Guards definiert und Codeblöcke hinzugefügt.<sup>69</sup> *Guards* werden zum Testen von Input-Kanten verwendet und können nur einen Booleschen Wert zurückgeben.

*Codeblöcke* umfassen ein Input-Pattern (optional), ein Output-Pattern (optional) und den Code zur Ausführung. Ein Input-Pattern beschreibt ein Variablen-Tupel. Nur die im Input-Pattern enthaltenen Variablen können im Code verwendet werden. Ein Output-Pattern definiert als Variable oder Tupel, welche Variablen

<sup>68</sup> „\*“: Bildung von Kreuzprodukten. Der Wertebereich von colset MM entspricht dem Kreuzprodukt von INTCOL und FAMILYCOL.

<sup>69</sup> In timed CPN können einer Transition auch Schaltzeiten zugeordnet werden.

als Folge der Ausführung des Codes manipuliert werden. Der Code selbst wird in CPN ML definiert und darf keine Variablen-Deklarationen enthalten. Lokale Funktionen können im Code definiert und genutzt werden. Ebenso kann im Code auf Konstanten zugegriffen werden.

### 3.3.6 Beispiel für ein CP-Netz im E-Learning

In Abbildung 15 wird ein CP-Netz für ein einfaches Protokoll beschrieben, mit dem eine Rechnung von einer virtuellen Universität für die Teilnahme an E-Learning-Kursen an den Studierenden „Alfred Neumann“ über das Internet versandt wird. Die Modellierung erfolgt mit Hilfe des Werkzeugs CPN Tools; das Beispiel ist angelehnt an (Jensen 1996a).

#### *CPN Tools-Notation*

Für das Verständnis des Coloured Petri Nets werden in im Folgenden die wesentlichen grafischen Elemente des Netzes beschrieben:

*Stellen:* Der Name der Stelle steht im grafischen Element der Stelle, d.h. dem Kreis bzw. der Ellipse. Grüne, bezifferte Kreise an den Stellen geben die Anzahl der Marken an, die in der jeweiligen Stelle vorhanden sind. So befinden sich in der Stelle *Send* zurzeit 8 Marken. Das Format der Marken, die Farbe, wird durch den Datentyp der Stelle angegeben, der neben der Stelle steht. Der Datentyp der Stelle *Send* ist INTxDATA, der Wertebereich der Marken entspricht somit dem Kreuzprodukt aus Integer- und DATA-Werten.<sup>70</sup> Die Ausgangsmarkierung der Stellen ist als Multimengennotation ebenfalls neben den Stellen zu finden. Für die Stelle *Send* sind insgesamt 8 Tupel aufgeführt. Die Anzahl, zu der das jeweilige Tupel in der Stelle vorkommt, wird entsprechend der Multimengennotation vor dem jeweiligen Tupel aufgeführt. So bedeutet  $1(1, \text{"Rechnung fuer Alfred Neumann ---"})$ , dass das Tupel  $(1, \text{"Rechnung fuer Alfred Neumann ---"})$  nur einmal in der Stelle vorliegt.

*Kanten:* Die gerichteten Kanten (Pfeile) werden mit Hilfe von Variablen beschriftet. Die an der Stelle *Send* anliegende Kante ist mit dem Tupel  $(n, p)$ , bestehend aus der Integer-Variablen  $n$  und der DATA-Variablen  $p$  beschriftet. Die Kante kann, da sie das Format der Stelle *Send* „unterstützt“, Werte aus *Send* lesen

---

<sup>70</sup> DATA entspricht z.B. dem Variablentyp String in der Java-Programmierung. Das Kreuzprodukt wurde hier in der Definition des Datentyps INTxDATA mit „x“ bezeichnet. In CPN ML-Notation:  $\text{colset INTxDATA} = \text{product INT * DATA}$ ;

und zur Verarbeitung an die Transition *SendPacket* weiterreichen.<sup>71</sup> Kanten können auch mit Funktionen belegt werden. Beispiele für Funktionen werden in den nachfolgenden Absätzen erläutert.

*Transitionen*: Transitionen, dargestellt als Vierecke, nehmen die Marken aus den eingehenden Kanten entgegen und schalten, sofern die in der Transition bzw. an den Kanten durch Funktionen formulierten Bedingungen erfüllt sind. Durch das Schalten werden die Stellen an den Ausgangskanten entsprechend der Kantenbeschriftung belegt. Ein einfacher Fall einer Transition ist in *ReceiveAck* zu finden. Beim Schalten der Transition wird der in der Stelle *NextSend* liegende Integer-Wert durch die Belegung der aus der Stelle ausgehenden Kante mit der Beschriftung *k* gelöscht. Gleichzeitig wird der aus der Stelle *D* stammende Wert der Variablen *n* in die Stelle *NextSend* geschrieben. Durch die Beschriftung der Kanten ist die Transition *ReceiveAck* vollständig beschrieben. In CP-Netzen können Transitionen selbst auch Codeblöcke enthalten. Ein Beispiel für eine solche Transition ist in Absatz 3.3.6.1 beschrieben. In Codeblöcken können auch Operationen zur Manipulation der eingehenden Marken enthalten sein.

Die *Farbgebung im Netz* trägt keine Bedeutung. Sie dient lediglich der Hervorhebung von Netzbestandteilen. Aktivierte Transitionen und Markenbelegungen des Netzes werden in CPN Tools grün hervorgehoben.

---

<sup>71</sup> Kanten mit doppelten Pfeilenden sind als nur-lesend zu interpretieren, da sie gleichzeitig lesen und löschen sowie den gelesenen Wert / die gelesene Marke wieder zurückschreiben.





*NextSend* enthält die Nummer des nächsten (aus der Sicht des Senders) zu übertragenden Tupels. Bei Erhalt einer Bestätigung einer erfolgreichen Übertragung wird die Marke in der Stelle inkrementiert.

*ReceivePacket* nimmt nur dann Pakete an, wenn die Paketnummer des eintreffenden Pakets  $n$  der Nummer  $k$  des Paketes in der Stelle *NextRec*, das als nächstes empfangen werden muss, entspricht. Falls die Paketnummern übereinstimmen, d.h. der Empfang erfolgreich ist, wird die Nummer des nächsten zu empfangenden Paketes um eins erhöht und der im Paket enthaltene String-Wert dem String in der Stelle *Received* durch Concatenation hinzugefügt. Falls der String-Wert des eingehenden Paketes das Stop-Wort "#####" enthält, so endet die Übertragung<sup>72</sup>.

Die Transition *SendPacket* kopiert das in der Stelle *Send* vorkommende Paket, das die Nummer  $n$  der Stelle *NextSend* enthält, in die Stelle  $A$ . Der Zähler in *NextSend* wird erst dann erhöht, wenn ein *ReceiveAck* schaltet. *TransmitPacket* überträgt eine Marke vom Sender an den Empfänger, indem es diese aus der Stelle  $A$  löscht und in die Stelle  $B$  einfügt. Die Funktion  $\text{Ok}(s, r)$  an der Kante, die von *TransmitPacket* nach  $B$  verweist, simuliert den Verlust von Paketen während der Übertragung im Internet. Sie liefert einen Wahrheitswert für den Vergleich der Zufallsvariablen  $r$ <sup>73</sup> mit der in der Stelle  $SP$  enthaltenen Marke  $s$  zurück (wahr, falls  $s \geq r$ ). Je höher  $s$  bei der Initialisierung gewählt wird, desto unwahrscheinlicher ist eine fehlerhafte Übertragung der Marke von  $A$  nach  $k$ .

Die Schaltung von *TransmitAck* erfolgt analog zur Transition *TransmitPacket*.

### 3.3.6.1 Codeblöcke als Transitionsinschriften in CP-Netzen

In CP-Netzen können Transitionsinschriften wie unter 3.3.4 beschrieben aus Codeblöcken bestehen. Ein Codeblock für eine Transition setzt sich zusammen aus:

- (1) einer Menge von Input-Variablen und
- (2) einem Action-Block, der die Verarbeitungsanweisungen enthält und beliebigen CPN ML-Code umfassen kann..

---

<sup>72</sup> Im Netz steht nur der Wert „stop“. Dieser wurde in den Deklarationen zum Netz wie folgt definiert: `val stop = „#####“;`

<sup>73</sup> Eine Zufallsvariable ist immer dann gegeben, wenn für eine Variable kein Binding vorliegt.

Der in Abbildung 16 stehende Codeblock zur Transition *ReceivePacket* bewirkt, dass bei einem  $n = 1$  durch die Verwendung der SML-Funktion `openOut` aus dem Modul `TextIO` eine Ausgabedatei namens „SimpleProtocol.SimRes“ geöffnet und für  $n > 1$  der Inhalt des jeweiligen Pakets in die Ausgabedatei geschrieben wird. Die Eingangsvariablen im Input-Block stammen aus den Stellen *B* und *NextRec*.

```
% input-Codeblock
input (n, p, k);
% ab hier Action-Codeblock
action
if n=k
then
    (if n=1
    then
outfile:=TextIO.openOut("SimpleProtocol.SimRes")
    else ();
    INTxDATA.output(!outfile, (n,p));
    if p=stop then TextIO.closeOut(!outfile) else ())
else ();
```

**Abbildung 16:** Codeblock für die Transition *ReceivePacket*

### 3.3.7 Analyse von CP-Netzen

Die Analyse von CP-Netzen wird durch CPN Tools-Funktionen zur *Simulation* von CP-Netzen (Simulation Tools) und zur *Erreichbarkeitsanalyse* (Statespace Tools) unterstützt (Ratzer u.a. 2003). Mit Hilfe des Simulationstools kann durch Folgemarkierungen durch manuelles bzw. automatisches Ausführen von Schaltvorgängen in einem CP-Netz iteriert werden. Simulationsergebnisse können in textueller Form abgespeichert werden, Schritte der Simulation werden im CP-Netz grafisch hervorgehoben.

Um eine Erreichbarkeitsanalyse von CP-Netzen durchzuführen, wird durch CPN Tools ein netzspezifischer Code für einen Erreichbarkeitsraum generiert, der für jede erreichbare Markierung im Netz einen eigenen Knoten enthält und die Basis für weitere Analysen bildet. Standard-Reports der Analysen mit Auswertungen

des CP-Netzes, z. B. hinsichtlich Beschränktheit, Lebendigkeit und Fairness, können gespeichert und mit einem Texteditor sichtbar gemacht werden. Eine Darstellung von Erreichbarkeitsgraphen, die mit der Vorgängerversion des Werkzeugs (Design/CPN) möglich war, kann mit CPN Tools leider nicht durchgeführt werden.

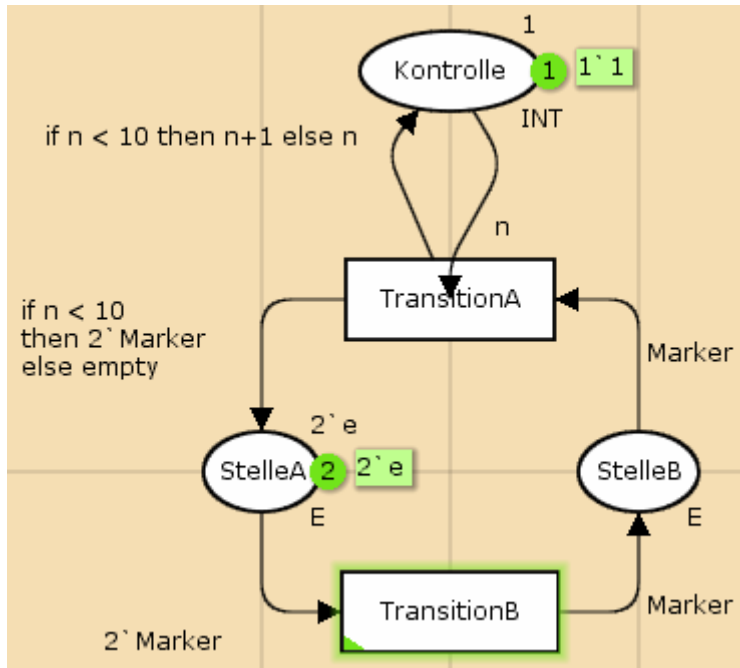
Abfragen an das Ergebnis der Erreichbarkeitsanalyse können innerhalb von so genannten „Auxiliary Text“-Blöcken erfasst und ausgewertet werden. Tote Markierungen in einem Netz können z. B. durch die folgende Abfrage ermittelt werden: `ListDeadMarkings()` ;

### 3.3.7.1 Erreichbarkeitsanalyse mit CP-Netzen

Die Erreichbarkeitsanalyse von CP-Netzen durch Generierung von State-Space-Reports wird anhand eines einfachen CP-Netzes in Abbildung 17 dargestellt.<sup>74</sup> Das CP-Netz umfasst zwei Stellen, die nicht unterscheidbare Marken aufnehmen können: *StelleA* beinhaltet als Ausgangsmarkierung zwei Marken (leere Tupel, Nulltupel), *StelleB* ist zu Anfang nicht besetzt. Die Stelle *Kontrolle* simuliert einen Zähler, der bei Schalten der Transition *TransitionA* um eins erhöht wird, solange er kleiner als 10 ist. Die Kanteninschrift der Kante zwischen *TransitionA* und *StelleA* sagt aus, dass *TransitionA* nur dann zwei Marken in *StelleA* ablegen kann, wenn der Zähler kleiner als 10 ist. Reduziert man die Betrachtung auf das Teilnetz, das aus den Stellen *StelleA* und *StelleB*, den Transitionen *TransitionA* und *TransitionB* sowie den Kanten, die diese Stellen und Transitionen verbinden, so wird ersichtlich, dass es sich um einen Zyklus handelt, in dem *TransitionA* und *TransitionB* beliebig oft hintereinander schalten können.

---

<sup>74</sup> Eine detailliertere Darstellung der Erreichbarkeitsanalyse findet sich bei (Jensen 1998).



```

▼DedStad
  ▼colset E = with e;
  ▼colset INT = int;
  ▼colset BOOL = bool;
  ▼colset STRING = string;
  ▼var Marker : E;
  ▼var n, k : INT;

```

Abbildung 17: Ein einfaches CP-Netz

Ein *State-Space* (s. o.) kann durch Anwendung des Werkzeugs „State-Space“ erzeugt werden. Das Ergebnis der Auswertung des Netzes aus Abbildung 17 ist in Abbildung 18 dargestellt und besteht aus fünf unterschiedlichen Blöcken, die neben statistischen Informationen Eigenschaften, z. B. die Beschränktheit, des CP-Netzes beschreiben.

Der erste Block der Auswertung enthält statistische Informationen zum Umfang des State-Space. Für das vorliegende CP-Netz enthält der State-Space 21 Knoten und 20 Kanten. Die gesamte Auswertung hat weniger als 1 Sekunde (0) gedauert und deckt den kompletten State-Space ab (full). Daneben werden im ersten Block auch Informationen zu einem *Strongly-Connected Components Graph* (SCC Graph) aufgelistet.<sup>75</sup> Der zweite Teil der Auswertung (*Boundedness-Properties*) enthält Informationen zur Beschränktheit der Knoten. Dabei wird unter Integer-Bounds die maximale bzw. minimale Anzahl der Marken, die ein Knoten aufnehmen kann, geführt. Die Stelle Kontrolle enthält z. B. immer nur

eine Markierung, deren Wert sich aber ändern kann. Ein *Upper Multi-Set Bound* gibt für die jeweilige Stelle die kleinstmögliche Multimenge an, die alle möglichen Markierungen, die die Stelle „durchlaufen“ kann, umfasst. Im Falle der Stelle *Kontrolle* sind dies die Markierungen  $1^1$  bis  $1^{10}$ , in Standard ML-Schreibweise für Multimengen:  $1^1 ++ 1^2 ++ 1^3 ++ 1^4 ++ 1^5 ++ 1^6 ++ 1^7 ++ 1^8 ++ 1^9 ++ 1^{10}$ .

Ein *Home-Marking* ist eine Markierung, die von sämtlichen anderen Markierungen im Netz aus, d.h. immer wieder, erreichbar ist. Eine *tote Markierung* ist eine Markierung des Netzes, bei der keine Transition mehr aktiviert ist. Im Falle des CP-Netzes handelt es sich um die Endmarkierung  $M_{21}$ , bei der  $n = 10$  gilt. Da Markierung  $M_{21}$  eine Home-Marking und tote Markierung ist, kann von jeder Markierung aus der Endzustand erreicht werden.<sup>76</sup>

Das CP-Netz enthält weder lebendige Transitionen, die immer wieder (irgendwann) aktiviert werden können, noch tote Transitionen, die nie schalten können. Das bedeutet, dass jede Transition mindestens einmal im Netz schalten kann. Bei Vorliegen einer toten Markierung kann es keine lebendigen Transitionen geben.

---

<sup>75</sup> Eine *Strongly-Connected-Component* ist ein maximaler Subgraph, bei dem es möglich ist, von jedem beliebigen Knoten einen Pfad zu jedem beliebigen anderen Knoten herzustellen.

<sup>76</sup> Eine Markierung ist per Definition immer von sich selbst aus erreichbar.

Statistics		
-----		
State Space		
Nodes:	21	
Arcs:	20	
Secs:	0	
Status:	Full	
Scc Graph		
Nodes:	21	
Arcs:	20	
Secs:	0	
Boundedness Properties		
-----		
Best Integers Bounds	Upper	Lower
MeinNetz'Kontrolle 1	1	1
MeinNetz'StelleA 1	2	0
MeinNetz'StelleB 1	1	0
Best Upper Multi-set Bounds		
MeinNetz'Kontrolle 1	1`1++1`2++1`3++1`4++1`5++1`6++1`7++1`8++1`9++1`10	
MeinNetz'StelleA 1	2`e	
MeinNetz'StelleB 1	1`e	
Best Lower Multi-set Bounds		
MeinNetz'Kontrolle 1	empty	
MeinNetz'StelleA 1	empty	
MeinNetz'StelleB 1	empty	
Home Properties		
-----		
Home Markings:	[21]	
Liveness Properties		
-----		
Dead Markings:	[21]	
Dead Transitions Instances:	None	
Live Transitions Instances:	None	
Fairness Properties		
-----		
No infinite occurrence sequences.		

**Abbildung 18: State-Space-Report für das CP-Netz aus Abbildung 17**

### 3.4 Zusammenfassende Bewertung

In den vorigen Abschnitten wurden ausgewählte Petri-Netz-Typen beschrieben. Dabei wurde die Betonung auf Gefärbte Petri-Netze (CPN) gelegt, die im Verlauf der Arbeit eine Rolle u. a. bei der Modellierung von Prozessen der zu entwickelnden Software spielen. CPN vereinen viele Vorteile (beliebig komplexe Markierungen, Vorhandensein von Analyseverfahren etc.) sind jedoch noch nicht für den kommerziellen Einsatz geeignet. Gründe hierfür liegen in der mangelnden Unterstützung zur Manipulation XML-Dokumenten<sup>77</sup> und in der fehlenden Bereitstellung von Schnittstellen zu relationalen Datenbanken.<sup>78</sup> Zudem fehlt eine Laufzeitumgebung, in der CPN ohne großen Aufwand ausgeführt werden können<sup>79</sup> und spezielle Operatoren / Transitionen, die im Sinne von Workflows eine Interaktion mit Benutzern erlauben.

Durch die Entwicklung einer Vielzahl von Petri-Netz-Typen wird es zunehmend für den Anwender schwierig, einen geeigneten „Dialekt“ zu identifizieren, der zur Lösung spezifischer Probleme geeignet erscheint. Ein erster Schritt zur Schaffung von Transparenz ist in der standardisierten Beschreibung von Petri-Netz-Typen mit der Petri Net Markup Language (PNML, vgl. (Jügel u.a. 2000; Weber und Kindler 2003)) zu sehen, die XML als Austauschformat nutzt. PNML wurde im Juli 2005 in Teilen als ISO-Working-Draft veröffentlicht (ISO/IEC WD 15909-2:2005(E)). Durch PNML wird auch ein Austausch von Petri-Netzen zwischen unterschiedlichen Werkzeugen möglich. Die Definition von PNML-Dokumenttypen für CPN wäre wünschenswert.<sup>80</sup>

---

<sup>77</sup> Eine XML-Unterstützung kann z. B. durch das Einbinden von weiteren Modulen in SML erreicht werden.

<sup>78</sup> Die Unterstützung des Datenbankzugriffs muss über Umwege erfolgen, z. B. durch Generierung von Textdateien durch das DB-System, die dann mit CPN Tools eingelesen werden können.

<sup>79</sup> Es kann nach Auskunft der CPN-Mailingliste jedoch auf die Laufzeitumgebung von SML NJ zurückgegriffen werden.

<sup>80</sup> Fraglich ist, inwieweit Konstrukte aus SML in der Sprachbeschreibung von CPN in PNML integriert werden können. In (Kemper 2000) wird mit APNN ein Austauschformat für Petri-Netze vorgestellt, das auch CP-Netze unterstützen soll.

## 4 Modelle und Systementwicklung

In diesem Kapitel werden die Grundbegriffe, die zum Verständnis des Entwurfs einer Beschreibungssprache zur konzeptuellen Modellierung von Preisstrukturen (Tarifen) benötigt werden, geklärt. Neben der Darstellung des „Modellbegriffs“ in Kapitel 4.1 werden in Kapitel 4.2 Kriterien beschrieben, die eine Bewertung von Modellen und Modellierungssprachen erlauben. Nach Überlegungen zur Genese von Modellierungssprachen in Kapitel 4.3 wird darauf aufbauend in Kapitel 4.4 ein Framework beschrieben, mit dessen Hilfe die Entwicklung sprachbasierter Informationssysteme strukturiert werden kann.

### 4.1 Modellbegriff

Nach (Stachowiak 1973), S. 131-132 besteht der Modellbegriff aus drei *Hauptmerkmalen*: Modelle verfügen über ein Abbildungsmerkmal, sind eine verkürzte Abbildung eines Realitätsausschnitts und weisen ein pragmatisches Merkmal auf.<sup>81</sup> (Kowalk 1996), S. 30 variiert die Definition von Stachowiak, indem er Modelle über zugrunde liegende Systeme, d. h. räumlich abgeschlossene und zeitlich begrenzte Einheiten, definiert. Modelle bilden bei ihm Merkmale eines realen oder künstlichen Systems ab und legen Beziehungen zwischen den Einheiten fest.<sup>82</sup> Im Folgenden wird die Definition von Stachowiak näher beschrieben (Stachowiak 1973), S. 131-132.

#### *Abbildungsmerkmal*

Modelle sind *Repräsentationen* natürlicher oder künstlicher *Originale*. Dabei können Originale wiederum selbst Modelle sein.

„[Originale] können dem Bereich der Symbole, der Welt der Vorstellungen und der Begriffe oder der physischen Wirklichkeit angehören.“ (Stachowiak 1973), S. 131.

Die Wahrnehmung von Originalen (Perzeption) erfolgt durch menschliche oder maschinelle Informationsverarbeiter oder, „unabhängig von momentaner Perzeption, in zentral-operationalen Prozessen [z. B. menschlichen

---

<sup>81</sup> Ein analoges Modellverständnis findet sich bspw. bei (Lee 1999).

<sup>82</sup> (Wedekind u.a. 1998), S. 267 kritisieren diese Ansicht, indem sie die Fähigkeit eines Modellierers in Frage stellen, ein System „zu erkennen.“ Dies liege insbesondere daran, dass ein



Denkprozessen].“ Alle von einem kognitiven Subjekt erfahrbaren Entitäten können als Original eines Modells angesehen werden.

„*Originale und Modelle werden [...] ausschließlich als Attributklassen gedeutet, die oft die Gestalt attributiver Systeme erlangen.*“ (Stachowiak 1973), S. 131.

Unter *Abbildung* ist dabei die Zuordnung von Modell-Attributen zu Original-Attributen zu verstehen. Dem Abbildungsbegriff kann der mathematische Abbildungsbegriff<sup>83</sup> zugrunde gelegt werden.

### ***Verkürzungsmerkmal***

Modelle abstrahieren, indem sie nur die Attribute des durch sie repräsentierten Originals erfassen, die den jeweiligen Akteuren, z. B. Modelldesignern und Modellnutzern, wichtig erscheinen. Auch mathematische Modellformulierungen basieren auf der Wahrnehmung des Modellierers. Durch die Verkürzung sollten jedoch die subjektive und intersubjektive Eindeutigkeit von Modellen gewahrt bleiben.

### ***Pragmatisches Merkmal***

„*Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für bestimmte – erkennende und / oder handelnde, modellbenutzende – Subjekte, b) innerhalb bestimmter Zeitintervalle und c) unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen [Zweck].*“ (Stachowiak 1973), S. 132.

Nach (Ludewig 2002) können Modelle in *deskriptive Modelle*, Abbilder von etwas, und *präskriptive Modelle*, die Vorbilder für etwas darstellen, unterschieden werden.<sup>84</sup> *Deskriptive Modelle* verfolgen das Ziel, den Zugriff auf relevante Informationen so einfach wie möglich zu gestalten. (Ludewig 2002) führt als Beispiel Organigramme an, die die Erkundung von Personalstrukturen durch ein

---

Modellierer nicht in der Lage ist zu beurteilen, wann die Abgeschlossenheitsbedingung für ein System gegeben ist.

<sup>83</sup> Z. B. mengenorientiert oder algebraisch.

<sup>84</sup> Obwohl nahe liegt, dass „Abbilder stets nach, Vorbilder stets vor dem Original existieren“, trifft diese Annahme nicht immer zu. *Prognosemodelle*, z. B. in Form von Wahlprognosen, treffen Aussagen bezüglich zukünftiger Phänomene und Objekte. Dennoch sind prognostische Modelle deskriptiv, da das Modell „am Original hängt“ und nicht umgekehrt.

Individuum ersparen. Jedes *präskriptive Modell* hat einen Ursprung, den es deskriptiv repräsentiert.

## 4.2 Bewertung von Modellen und Beschreibungssprachen

Die qualitative Bewertung von Modellen als alternative Beschreibungen eines Originals kann z. B. mit Hilfe der in (Schütte 1998), S. 119-175 vorgestellten Grundsätze ordnungsmäßiger Modellierung (GoM) erfolgen. Schütte nennt dabei folgende *Qualitätskriterien* für Modelle:

1. *Konstruktionsadäquanz*: Nutzen eines Modells für seine praktische Anwendung,
2. *Sprachadäquanz*: Spracheignung, semantische Mächtigkeit, Formalisierungsgrad, Sprachverständlichkeit,
3. *Sprachrichtigkeit*: korrekte Syntax, Vollständigkeit gegenüber einem Metamodell,
4. *Wirtschaftlichkeit*: ökonomische Bewertung der Anpassungsfähigkeit, Amortisationszeit usw.,
5. *Klarheit*: Verständlichkeit und Eindeutigkeit und
6. *Systematischer Aufbau*: Inter-Modellkonsistenz zwischen Struktur- und Verhaltensmodellen, Unterstützung von Sichten und Vergleichbarkeit (semantische Deckungsgleichheit zweier Modelle).

Dabei stehen viele der o.g. Ziele in Konflikt zueinander. Ein hoher Grad an Konstruktionsadäquanz ist wirtschaftlich nicht unbedingt sinnvoll, wenn dadurch in der Kommunikation zwischen Modellierern hohe Kosten verursacht werden. Die Klarheit eines Modells hingegen stellt i. d. R. ein zur Konstruktionsadäquanz komplementäres Ziel dar.

Die Beurteilungskriterien nach (Schütte 1998) können auch als *Qualitätskriterien für Modellierungs- / Beschreibungssprachen* herangezogen werden (vgl. Kapitel 5.5.4, S. 135 ff.). Beschreibungssprachen werden genutzt, um Modelle zu definieren. Durch die Wahl einer Beschreibungssprache (grafisch, textuell, mathematisch-formal etc.) wird insbesondere auch die praktische Umsetzbarkeit von Modellen in Informationssystemen bestimmt (s. Gliederungspunkt 1.). Um Aussagen über die *Angemessenheit* (Adäquanz) einer Beschreibungssprache zu treffen, werden in der Literatur für unterschiedliche Klassen von Beschreibungssprachen Kriterienkataloge definiert. Ein Kriterienkatalog mit Anforderungen an

Beschreibungssprachen für Geschäftsprozesse ist z. B. bei (Frank und van Laak 2003) zu finden. (Zelewski 1996) definiert Kriterien, mit denen die Eignung unterschiedlicher Petri-Netz-Varianten zur Modellierung von Realsystemen untersucht werden können. Abbildung 19 und Abbildung 20 listen die Kriterien auf. Analog zu den von Schütte aufgestellten Kriterien, können die Kriterien nach Zelewski zur Beurteilung einer Modellierungssprache adaptiert werden (vgl. Kapitel 5.5.1, S. 131 ff.).

Kriterienkatalog zur Beurteilung von Petri-Netz-Typen (Teil 1) – Modellierungsgüte
<p>1. Modellierungsgüte</p> <p>a. Pragmatische, „anwendungsnahe“ Qualitäten</p> <p><i>Abbildung von Modellierungsphasen</i></p> <p>i. Konstruktivität</p> <ul style="list-style-type: none"> <li>○ Ausmaß, in dem die <i>systematische Konstruktion</i> von Modellen durch Entwurfshilfen unterstützt wird (z.B. durch Hierarchisierung von Modellen)</li> <li>○ Modularität (Baukastenprinzip)</li> </ul> <p>ii. Analysierbarkeit</p> <ul style="list-style-type: none"> <li>○ Modelleigenschaften, die für Anwender „von Interesse“ (Anzahl und Unterschiede der verfügbaren Analysetechniken für Modelle)</li> <li>○ Sonstige Eigenschaften (Lebendigkeit etc.)</li> </ul> <p>iii. Adaptivität (Anpassung an Modellierungsbedürfnisse)</p> <ul style="list-style-type: none"> <li>○ Verfeinerung / Vergrößerung (vertikal)</li> <li>○ Erweiterung / Restriktion (horizontal)</li> <li>○ Parametrische Modellformulierung (durch Einführung von Variablen)</li> </ul> <p><i>Phasenunspezifische Aspekte</i></p> <p>iv. Anwendungseffizienz</p> <ul style="list-style-type: none"> <li>○ Bei Konstruktion und Auswertung eines Modells</li> </ul> <p>v. Benutzerfreundlichkeit</p> <ul style="list-style-type: none"> <li>○ Grafische und/oder formale Modellierung</li> </ul> <p>vi. Implementierbarkeit</p> <ul style="list-style-type: none"> <li>○ Realisierung von Modellierungskonzepten in Informationssystemen</li> <li>○ Automatisierbarkeit</li> </ul>

Kriterienkatalog zur Beurteilung von Petri-Netz-Typen (Teil 1) – Modellierungsgüte
<p>1. Modellierungsgüte (Fortsetzung)</p> <p>b. Theoretische, eher „anwendungsferne“ Qualitäten eines Modellierungskonzepts, u.a.</p> <p>Konsistenz, Eindeutigkeit, Formalisierungsgrad, Interpretierbarkeit, Operationalität, Realitätsadäquanz, Einfachheit, Einheitlichkeit, Vollständigkeit, Integrationsqualität, Fruchtbarkeit</p>

Abbildung 19: Modellierungsgüte als Kriterium zur Evaluation von Petri-Netz-Typen<sup>85</sup>

Kriterienkatalog zur Beurteilung von Petri-Netz-Typen (Teil 2) – Modellierungsfähigkeit
<p>2. Modellierungsfähigkeit</p> <p>a. Allgemeine /Universalität</p> <ul style="list-style-type: none"> <li>○ Vergleich mit der Ausdrucksmächtigkeit von Turing-Automaten durch Einordnung in die folgenden Klassen <ul style="list-style-type: none"> <li>• Einfache arithmetische Ausdrücke: Addition und Subtraktion im Bereich der nichtnegativen Ganzzahlen (nichtnegative Pressburger Arithmetik)</li> <li>• arithmetische Ausdrücke</li> <li>• Ausdrücke aus einer sortierten Algebra</li> <li>• Ausdrücke der konventionellen Prädikatenlogik 1. Stufe</li> <li>• Ausdrücke einer sortierten Prädikatenlogik 1. Stufe</li> <li>• Ausdrücke, die sich von Turing-Automaten bewältigen lassen</li> </ul> </li> </ul> <p>b. Spezielle (kein weithin akzeptierter Kriterienkatalog), u.a.</p> <p>Darstellung von Alternativen, Konkurrenz um Ressourcen, Eigenschaften der Objekte (Menge, Qualität, ...), Individuelle Betrachtung der ObjektZeitaspekte abbildbar, Priorisierung von Arbeitsabläufen</p>

Abbildung 20: Modellierungsfähigkeit als Kriterium zur Evaluation von Petri-Netz-Typen<sup>85</sup>

### 4.3 Modellbildung – Überblick und Definitionen

Ein Ziel dieser Arbeit liegt darin, eine *Beschreibungssprache* (kurz Sprache) zur grafischen Modellierung<sup>86</sup> des Pricings digitaler Produkte zu entwickeln. Eine zu entwickelnde / zu untersuchende Sprache bezeichnet (Holten 1999), S. 11 als *Objektsprache*. *Metasprachen* hingegen dienen der Analyse und Beschreibung

<sup>85</sup> Nach (Zelewski 1996).

<sup>86</sup> Grafische Modelle sind anschaulich-räumliche Abbildungen eines „Originals“ (Stachowiak 1973), S. 159.

von Objektsprachen.<sup>87</sup> Sprachen werden dazu verwendet, „Begriffe und Sachverhalte mitteilen zu können“ (Strahinger 1996), S. 17. Dabei umfasst eine Sprache neben einer Menge von Zeichen / Symbolen auch die Regeln zur Verwendung der Zeichen / Symbole. Mit Hilfe der in einer Modellierungssprache verfügbaren Zeichen und Symbole können Modelle konstruiert werden. Sprachliche Abbildungen sind dann als Modelle eines Gegenstandsbereiches zu betrachten, wenn Abbildung und Gegenstandsbereich übereinstimmen. (Strahinger 1996), S. 24-28. Wird eine Sprache, mit deren Hilfe Modelle gebildet werden, selbst in einem Beschreibungsmodell abgebildet, so wird letzteres als *Metamodell* bezüglich des mit Hilfe eines Modells beschriebenen *Gegenstandsbereichs* bezeichnet. Insofern kann ein Metamodell als mittelbares Modell eines Modells angesehen werden (vgl. Abbildung 21).

(Holten 2003b), S. 81-110 entwickelt eine Methodik zur „Formalisierung von Modellsprachen-Systemen“ im Rahmen einer umfassenden „Theorie der Informationssysteme und der IS-Integration“, die auf der Logischen Propädeutik von (Kamlah und Lorenzen 1996) aufbaut.

Die Theorie konkretisiert die Ausführungen von Strahinger, indem zwischen *intensionaler Abstraktion* auf einer Modell-Ebene, z.B. der Typebene, und *extensionaler Abstraktion* zwischen unterschiedlichen Ebenen, z.B. von der Instanz- zur Typebene, unterschieden wird.

#### 4.3.1 Intensionale Abstraktion

In Abbildung 21 wird die zweistufige Abstraktion von der Rede<sup>88</sup> zur intensionalen Bedeutung auf der Typebene nach (Kamlah und Lorenzen 1996) dargestellt: Zur Entwicklung einer Modell[ierungs]sprache M „müssen [...] zunächst eine Menge von Zeichenschemas (Symbole) für M und anschließend die intensionale Bedeutung<sup>89</sup> der Zeichenschemas vereinbart werden. Das Gleiche gilt für jede syntaktische Regel [der Modellierungssprache].“ (Holten 2003b), S. 82.

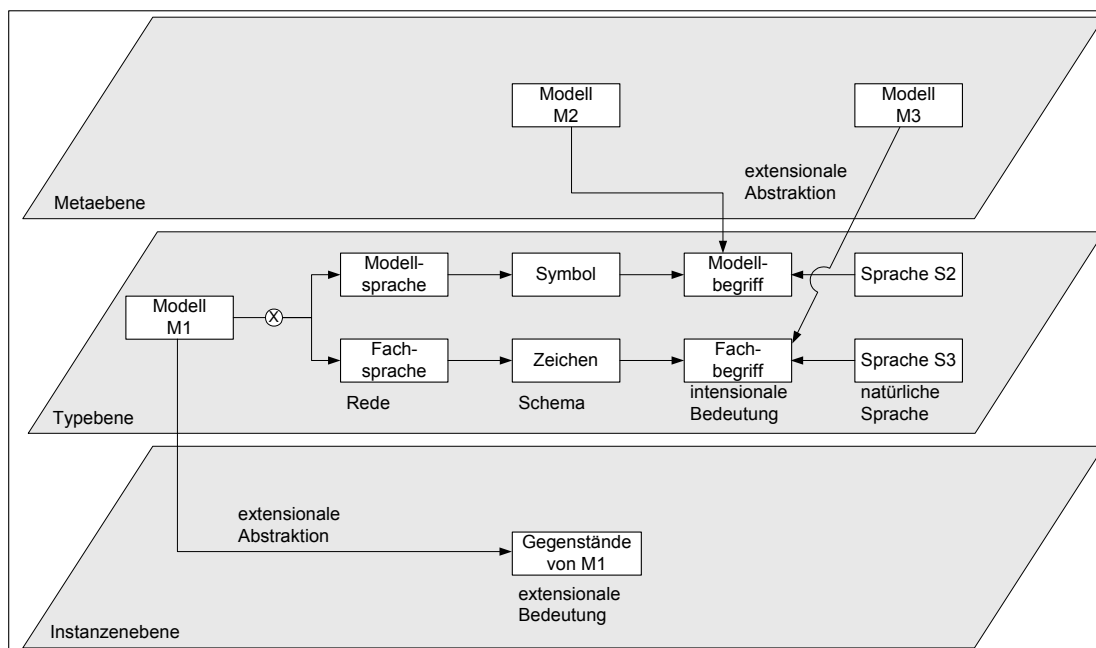
---

<sup>87</sup> Objektsprachen können selbst wieder Metasprachen für Sprachen bilden (Sprachstufentheorie). Vgl. (Strahinger 1996), S. 18, 24 bzw. (Holten 1999), S. 11.

<sup>88</sup> Unter „Rede“ ist in diesem Zusammenhang auch das „Modellieren“ durch Verwendung einer Modell- und einer Fachsprache zu verstehen.

<sup>89</sup> Die Bedeutung eines Wortes [grafischen Symbols] ist dasjenige, was das Wort aufgrund von expliziter oder impliziter Vereinbarung zwischen den an der Kommunikation Teilnehmenden zu verstehen gibt. Vgl. (Holten 2003b), S. 40.

Das Ziel der fachkonzeptionellen Modellierung liegt darin, Sachverhalte fachsprachlich auszudrücken und in eine formale, modellsprachliche Repräsentation (Modell) zu überführen. Voraussetzung dafür ist, dass ein Symbolvorrat definiert wird, der bei der grafischen Modellierung verwendet werden kann. Zudem müssen die fachsprachlichen Begriffe im Sinne einer Orthosprache durch Festlegung von Wörtern / Zeichen definiert werden. Die Definition der zuvor beschriebenen Sprachbestandteile und ihrer Bedeutungen erfolgt mit Hilfe natürlicher Sprache (S2, S3 in der Abbildung), die „[m]it Bezug auf die Sprachstufentheorie der Logik [...] [dann] als Metasprache[n] bezüglich der Modellsprache und der Fachsprache“ anzusehen ist. (Holten 2003b), S. 106.



**Abbildung 21: Metaebene durch zweifache extensionale Abstraktion, aus (Holten 2003b), S. 109**

Die fachkonzeptionelle Modellierung, z. B. die Erstellung von Preismodellen, kann also als „komplexe Sprachhandlung beschrieben werden, welche im Kern fachsprachliche Sachverhalte in eine modellsprachliche Repräsentation überführt (Übersetzung im Sinne KAMLAHS und LORENZENS).“ (Holten 2003b), S. 88-89.

Begriffe als „intensionale Bedeutungen der modellsprachlichen Symbole und der fachsprachlichen Zeichen werden in der Regel mit Beispielen und umfangreichen Erläuterungen eingeführt (Holten 2003b), S. 105 (vgl. Kapitel 4.4 für ein Vorgehen zur Einführung einer Modellierungssprache)

### 4.3.2 Extensionale Abstraktion

„Ein fachkonzeptionelles Modell als komplexe fachsprachliche Aussage in formaler Strukturierung ist [...] eine Darstellung einer Menge.“ (Holten 2003b), S. 105. Die Mengeneigenschaft von Modellen besagt, dass es für jedes Modell als Menge eine Vielzahl von Instanzen / Gegenständen als Elemente der Menge geben kann. Das Modell (M1) bildet somit den Typ der zugehörigen Instanzen / Gegenstände, indem es von den Extensionen (Instanzen) abstrahiert (vgl. Instanzenebene in Abbildung 21). Ebenso können Modellbegriffe der Typebene als Extension des Metamodells M2 angesehen werden.

Ein fachkonzeptionelles Modell hat demnach eine Bedeutung (Intension) und umschreibt als abstraktes Objekt / Menge eine Vielzahl von Elementen (extensionale Bedeutung), d.h. Gegenstände, für die die fachsprachliche Aussage, die durch das Modell repräsentiert wird, wahr ist. (Holten 2003b), S. 105

## 4.4 Vorgehen zur Gestaltung von Modellierungssprachen

### 4.4.1 Entwurf einer Orthosprache (Schritt 1)

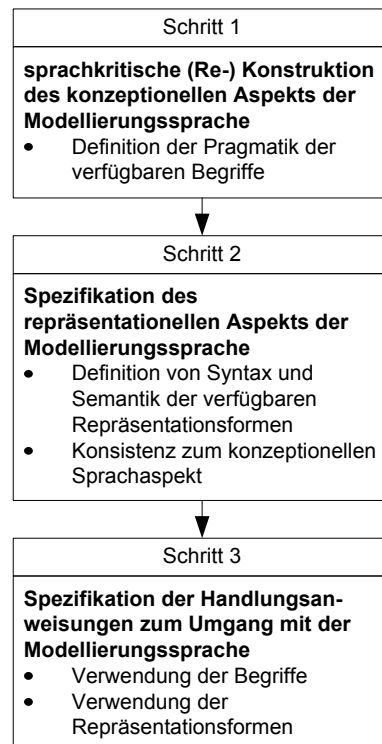
Soll eine Sprache für einen Anwendungsbereich, z. B. das Aufstellen von Tarifen, „entwickelt“ werden, so müssen in einem ersten Schritt die für den Anwendungsbereich üblichen Begriffe sowie Relationen zwischen den Begriffen definiert werden (vgl. Abbildung 22).

Die *semantische Beschreibung* der Begriffe verfolgt dann den Zweck der *Normierung* bzw. der Herausbildung einer Orthosprache (Holten 2001), S. 2. (Ortner 2005), S. 248 beschreibt den Begriff der *Orthosprache* wie folgt:

“Eine rationale Zwischensprache, die als eine Art Ideal- oder Universalsprache (auch Norm- oder Rationalsprache genannt) eingesetzt wird, wurde von Lorenzen [...] Orthosprache (vom griechischen orthos = richtig) genannt. Eine Orthosprache ist eine objekt- und metasprachlich aufgebaute (materiale) Sprache, in der jedes Wort oder jedes Zeichen ausdrücklich und zirkelfrei in seiner Verwendungsweise aus der Praxis (d. h. empraktisch) rekonstruiert festgelegt ist. Daher ist auch jede Fach-Normsprache eine Orthosprache.”<sup>90</sup>

---

<sup>90</sup> (Lehmann 1998) gibt dem Begriff „Normsprache“ gegenüber der Verwendung des Begriffs „Orthosprache“ den Vorzug, „um nicht den falschen Eindruck zu erwecken, [die] Normsprache sei die einzig *richtige* Sprache, etwa im Sinne einer Idealsprache.“ Der Begriff der Orthosprache geht auf (Lorenzen 1973) zurück.



*Abbildung 22: Methodologie zur Entwicklung von Modellierungstechniken für Fachkonzepte<sup>91</sup>*

Im Rahmen der Arbeit wird der *konzeptuelle Aspekt der Beschreibungssprache* mit Hilfe eines *Entity-Relationship-Diagramms* (Chen 1977) als Metamodell<sup>92</sup> dargestellt (vgl. (Holten 1999), S.24-28). Dabei werden die in der Orthosprache gebildeten bzw. normierten Begriffe mit Hilfe von Entity-Typen und die Beziehungen zwischen den Begriffen mit Relationship-Typen abgebildet. Zur *semantischen Beschreibung* der Begriffe der Orthosprache wird in der Arbeit die Textform gewählt. Die Zuordnung von Elementen aus dem Metamodell zu den Begriffen erfolgt tabellarisch. (vgl. Kapitel 5.4.3)

#### **4.4.2 Entwurf der Notationen (Schritt 2)**

Nach (Karagiannis und Kühn 2002), Abs. 2 enthält eine Beschreibungssprache „[...] the elements, with which a model can be described.“ Nachdem die Semantik der Modellierungssprache in einem Metamodell fachkonzeptuell beschrieben wurde, muss der repräsentationale Aspekt der Sprache in einem zweiten Schritt entworfen werden, indem den Begriffen / Bausteinen der Sprache *Repräsentationselemente* im Sinne einer *Notation der Sprache* zugeordnet

<sup>91</sup> Modifiziert nach (Holten 2001), S.12.



werden. Dabei wird die „Semantik der Repräsentationselemente [...] durch die Zuordnung zu den Begriffen der Orthosprache [...] festgelegt.“ (Holten 2001), S. 14. Die *Syntax* beschreibt die Elemente und die Regeln, die für das Aufstellen von Modellen gelten. Sie ist in einer *Grammatik* definiert. Es ist darauf zu achten, dass die Konsistenz zum konzeptuellen Modell gewahrt bleibt (vgl. Schritt 2 in Abbildung 22). Neben einer relationalen Notation, im Sinne eines relationalen Datenbankschemas, die Voraussetzung für die Speicherung der Modelle ist, wird eine grafische Notation beschrieben, mit der Modelle visualisiert werden können.

#### **4.4.3 Entwurf der Prozessunterstützung für den Anwender (Schritt 3)**

Eine *vollständige Methode der Systementwicklung* muss neben der Bereitstellung von Beschreibungssprachen auch den *Prozess der Modellbildung* (Vorgehen / Modellierung) beschreiben (Greiffenberg 2003), S. 958; (Strahinger 1996), S. 26 und dem Nutzer der Sprache Handlungsanweisungen zum Umgang geben (vgl. Schritt 3 in Abbildung 22). Der Modellierungsprozess beschreibt die Schritte, die zur Erzielung von Resultaten mit einer Beschreibungssprache notwendig sind (Karagiannis und Kühn 2002), Abs. 2. In Analogie zur sprachbasierten Metaisierung, können auch Prozesse als Gegenstand der Metaisierung betrachtet werden. Prozesse zur Aufstellung von Preismodellen, die in einem Prototyp implementiert wurden, werden in Kapitel 6.6.3 der Arbeit als Anwendungsfälle Gefärbte Petri-Netze beschrieben.

#### ***Zwischenfazit***

Abbildung 23 fasst die Erläuterungen grafisch zusammen und skizziert den Bezug zur Arbeit. Der *Gegenstandsbereich* der Arbeit liegt im Pricing von digitalen Produkten. Dabei soll eine *Objektsprache nebst einer zugehörigen grafischen Notation* entwickelt werden, die die Definition von Tarifen in Modellform ermöglicht. Die Konzepte der Sprache werden mit Hilfe von Entity-Relationship-Diagrammen dargestellt.<sup>93</sup> Um den Entwurfsprozess für Modelle und die *Prozesse*

---

<sup>92</sup> Nach (Ferstl und Sinz 1998) werden in einem *Metamodell* die Arten von Modellbausteinen, die Arten von Beziehungen zwischen den Bausteinen und die Regeln zur Verknüpfung der Bausteine durch Beziehungen abgebildet.

<sup>93</sup> Alternativ könnten Klassendiagramme der Unified Modeling Language (UML) verwendet werden. Die Beschreibung mit ER-Diagrammen bietet den Vorteil, dass durch Überführung des in ER-Notation beschriebenen Metamodells in Relationen eine direkte Abbildung der Sprachkonstrukte in einer Datenbank erfolgen kann. Die Entwicklung einer Metasprache als ER-

in Abrechnungssystemen zu spezifizieren werden Gefärbte Petri-Netze als Prozessbeschreibungssprache verwendet. Diese bilden eine Weiterentwicklung von Prädikate/Transitionen-Netzen und bieten durch die Einführung von Datentypen die Möglichkeit, Prozesse zum Lesen, zur Manipulation und zum Löschen von Datenobjekten zu beschreiben (vgl. Kapitel 3.3). Datenobjekte können beliebig komplex sein. Ihre Struktur wird in Datentypen definiert.

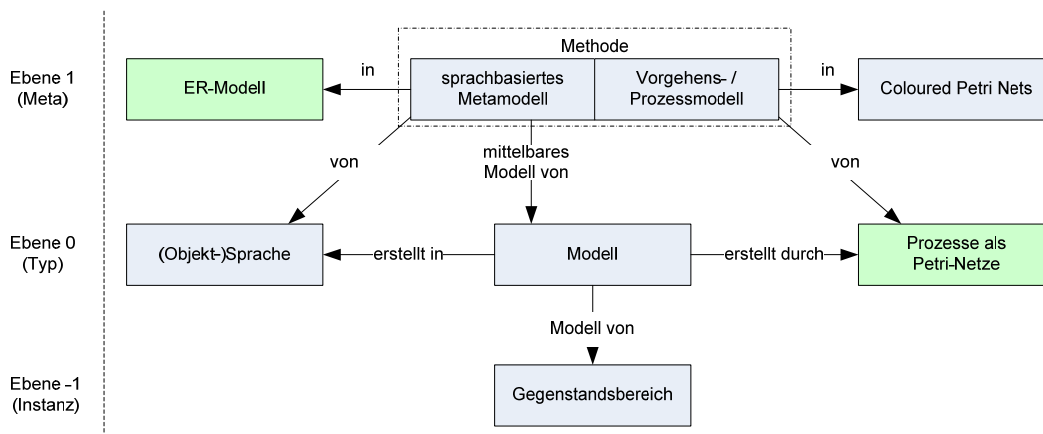


Abbildung 23: Metamodell, Modell, Vorgehen<sup>94</sup>

## 4.5 Rahmenmodell zur IS-Entwicklung

Um den Entwurf der Sprache zur Definition von Preismodellen und eines Software-Prototyps für die Sprache zu strukturieren, wird das Vorgehen zur Umsetzung der sprachlichen Konzepte in einem Informationssystem, einem Abrechnungssystem, in diesem Abschnitt in Anlehnung an (Holten 2002, 2003b) beschrieben und in Abbildung 24, S. 96 skizziert.

### Abstraktionsebenen

„Für die Integration von Informationssystemen [bzw. die Erstellung von Abrechnungssystemen, d.Verf.] sind drei Abstraktionsebenen relevant: Die *Instanzenebene* betrachtet konkrete Vorfälle im Rahmen der Durchführung von Geschäftsprozessen [...]. Die *Typeebene* spezifiziert, wie der Bestellprozess [bzw. Geschäftsprozess, d.Verf.] generell durchzuführen ist [...]. Die *Metaebene* spezifiziert das Schema der Elemente der Typebene [...].“ (Holten 2003a) Neben dieser prozessualen Betrachtung des Anwendungsbereichs muss zur Entwicklung

Diagramm geht also mittelbar mit der Entwicklung eines Datenbankschemas zur Speicherung der Modelle, die mit einer Objektsprache erzeugt werden, einher.

<sup>94</sup> Quelle: modifiziert nach (Greiffenberg 2003), S. 958 und (Strahinger 1996), S.26.

sprachbasierter Informationssysteme außerdem eine Objektsprache definiert werden. Die Beschreibung einer solchen Sprache erfolgt ebenfalls durch Abstraktion über Instanzen-, Typ- und Metaebene und wurde in Kapitel 4.3, Abbildung 21 sowie Abbildung 23 dargestellt.

Im Rahmen der Bepreisung und Abrechnung digitaler Produkte und Dienstleistungen besteht der oben erwähnte Geschäftsprozess auf Typebene aus mehreren Teilprozessen, die voneinander unterschieden werden können: Für die Bepreisung ist es unerlässlich, dass in einem ersten Teilprozess der Konsum digitaler Produkte und Dienstleistungen durch einen Kunden messbar gemacht wird. Auf Basis dieses *Verbrauchsprozesses* bzw. der Verbrauchsprozesse muss ein zweiter *Teilprozess für die Bepreisung* entworfen werden.<sup>95</sup> Vereinfachend wird im Folgenden davon ausgegangen, dass auf der Instanzenebene nur der zugrundeliegende Verbrauchsprozess betrachtet wird.

„Die Entwicklung einer einheitlichen Terminologie ist zwingende Voraussetzung der Integration von Geschäftsprozessen und Informationssystemen. Die Bedeutung dieser terminologischen Integration liegt in der Notwendigkeit, zusätzlich zur Korrektheit der Strukturen (mittels Grammatiken und Datentypen) auch die Korrektheit der Inhalte (mittels der Fachterminologie) [die für die betrachtete Domäne des Pricings in der Sprache PriMoL kodifiziert ist] prüfen zu müssen.“ (Holten 2003a) Die Entwicklung der Beschreibungssprache PriMoL (Kapitel 5.3), basiert in dieser Arbeit auf der Analyse bestehender Bepreisungsmechanismen im E-Learning, im Versicherungsbereich und in der Telekommunikationsindustrie (vgl. Kapitel 5.3).

### ***Phasen der Systementwicklung***

Um den Entwicklungsprozess eines Informationssystems zu strukturieren, kann dieser in einzelne Phasen unterteilt werden. (Scheer 1997), S. 14 teilt den Prozess der Systementwicklung in drei Phasen (Fachkonzeption, DV-Konzeption und Implementierung). „Aus methodischer Sicht sind die[se] Entwicklungsphasen [...] mit den Abstraktionsebenen zu kombinieren.“ (Holten 2003a) Durch die Kombination ergibt sich für die vorliegende Arbeit das in Abbildung 24

---

<sup>95</sup> Der Entwurf der Teilprozesse beeinflusst sich gegenseitig. Oft werden erst Verbräuche in einem Abrechnungssystem erst „festgehalten“, wenn diese zur Bepreisung genutzt werden sollen. Die Entwicklung eines Informationssystems für die Abrechnung digitaler Produkte führt oft zu einer Adaption bestehender Service Provisioning-Systeme, mit deren Hilfe digitale Produkte vertrieben werden.

dargestellte Rahmenmodell mit neun Feldern. Die einzelnen Phasen werden im Folgenden beschrieben.

### ***Fachkonzeption***

In der Phase der *Fachkonzeption* werden die Anforderungen an das Informationssystem aus fachlicher Sicht spezifiziert (Fachspezifikation). Hierzu gehört die Spezifikation der Funktionsweise des Systems. Das zu entwickelnde System soll den *Entwurf sowie die Umsetzung von Preismodellen* in einem Abrechnungssystem prototypisch demonstrieren. Die Fachspezifikation besteht aus mehreren Teilen:

- (1) Den wesentlichen Teil der Fachspezifikation umfasst der konzeptuelle Entwurf der *Beschreibungssprache PriMoL* (Kapitel 5.3).
- (2) Zur Erzeugung von Preismodellen mit PriMoL wird in Kapitel 6 ein *Price-Modeler-Prototyp* spezifiziert.
- (3) Ein *prototypisches Lern-Managementsystem* zur Darstellung von Verbrauchsprozessen sowie zur Instanziierung der Preismodelle wird in Kapitel 7 spezifiziert.

Auf der *Instanzenebene* werden in der Phase der Fachkonzeption Durchführungen konkreter Prozesse als Geschäftsvorfälle beschrieben. Diese Beschreibungen sind umgangssprachlich gefasst. Beispiele für Geschäftsvorfälle in einem Lern-Managementsystem finden sich in Kapitel 7.1.5.<sup>96</sup> Ein Beispiel-Geschäftsvorfall ist auch in der Abbildung zu finden: Die Anmeldung eines Lernenden soll zu einer Rechnung am Monatsende führen.

Im fachkonzeptuellen Modell der *Typebene* wird die Sprache PriMoL als Orthosprache definiert (vgl. Abschnitt 4.3). Begriffe werden etabliert und in Beziehung zueinander gesetzt. Neben der Herausbildung der Sprache selbst (vgl. Kapitel 5.4.3) werden unterschiedliche Notationen (Kapitel 5.4.5, 5.4.6) entworfen, die die Begriffe der Sprache repräsentieren. Mit Hilfe dieser Notationen werden Modelle aufgestellt, die in den späteren Phasen sukzessive umgesetzt werden. Die aufgestellten Modelle definieren auf konzeptueller Ebene, wie Geschäftsvorfälle auf Instanzenebene zu interpretieren, d.h. zu bepreisen, sind.

---

<sup>96</sup> Dort aus Sicht der Systementwicklung als „Anwendungsfälle“ bezeichnet.

„Werden Modelle von Modellierungstechniken und Terminologien, die auf der Typebene verwendet werden, erstellt, sind dies mit Bezug auf die Sachverhalte der Instanzenebene *fachkonzeptionelle Metamodelle*. Diese sind der Metaebene zuzuordnen“ (Holten 2003a) Das fachkonzeptuelle Metamodell der Beschreibungssprache ist in Abbildung 29, auf Seite 123 beschrieben.

### ***DV-Konzeption***

Im Rahmen der DV-Konzeption muss auf *Instanzenebene* festgelegt werden, welche Datentypen und Speicherplätze bei der Abwicklung der Geschäftsvorfälle adressiert und reserviert werden. Dabei werden auf Ebene des DV-Konzepts noch keine individuellen Werte betrachtet. Hierzu gehört die Einrichtung von Datenbankschemas zur Ablage der Preismodelle und der Geschäftsvorfälle.

Auf *Typeebene* sind u.a. die technischen Parameter der Preismodelle zu konkretisieren. Für die Elemente eines Preismodells, einfache Preise, komplexe Preise und Aggregationsregeln sowie deren Dimensionen sind Datentypen festzulegen und Bewertungsregeln zu definieren, die die Verknüpfung eines Preismodells mit einem Geschäftsvorfall (Verbrauchsprozess) ermöglichen.

„Auf der *Metaebene* sind die logischen Schemas [...], die die Informationen der Typebene aufnehmen, zu spezifizieren.“ (Holten 2003a) Hierzu gehört der Entwurf eines Speicherformats für Preismodelle z.B. in relationaler Notation oder in der im Anhang A.2.2 XML-Schema von PriMoL aufgeführten XML Schema-Notation.

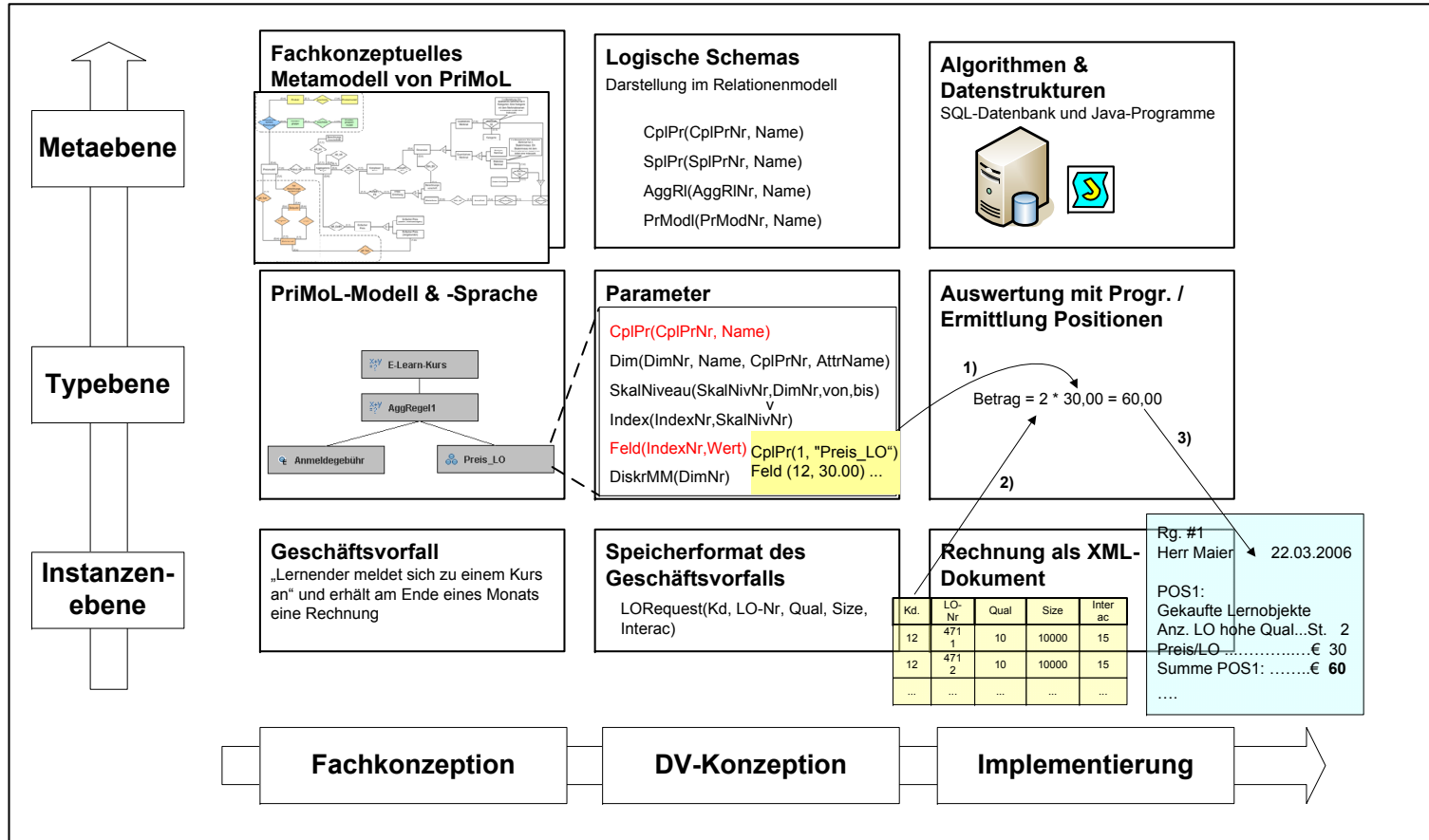
### ***Implementierung***

Auf der *Instanzenebene* wird ein Geschäftsvorfall, z.B. die Interaktion eines Lernenden mit einem Lern-Managementsystem, als Datensatz im Lern-Managementsystem gespeichert und davon ausgehend kontinuierlich oder in regelmäßigen Abständen eine Bepreisung der Verbräuche getriggert. Die Auswertung des Preismodells, d.h. die Berechnung von Rechnungspositionen und die Erstellung der Rechnung findet durch Programme auf *Typeebene* statt. Die Programmgenerierung auf Basis des Preismodells (im Feld Typeebene, DV-Konzeption) kann auf zwei Arten erfolgen: Zum einen können Preismodelle selbst hart kodiert werden, indem diese durch geeignete Algorithmen fest in einem Programm verdrahtet werden. Diese Vorgehensweise bei der Implementierung ist

jedoch unflexibel, da eine Änderung eines Preismodells zu einer Neuprogrammierung des Abrechnungssystems führt. Die in der Arbeit vorgestellten Preismodelle werden durch eine Laufzeitumgebung des Lern-Managementsystems (Kapitel 7.1.6) interpretiert. Eine Neuprogrammierung des Abrechnungssystems ist daher bei einer Änderung des Preismodells nicht notwendig. Zur Implementierung eines derart „flexiblen“ Abrechnungssystems wurden eine Vielzahl von Programmbibliotheken, Entwurfsmuster und Algorithmen analysiert (*Metaebene*) und in der Programmierung des Price Modelers sowie des Abrechnungssystems (*Typeebene*) berücksichtigt.

Das vorgestellte Rahmenmodell bietet einen groben Überblick über die Systementwicklung. Für die Umsetzung der in den „Feldern“ beschriebenen Konzepte kommen in der Arbeit weitere Techniken zum Einsatz. In Kapitel 6 wird die Entwicklung eines Prototyps zur Preismodellierung durch ein Vorgehensmodell einer Architektur-getriebenen Systementwicklung konkretisiert.

Abbildung 24: Rahmenmodell zur Entwicklung nach (Hohen 2002), S. 18



- 1) Auslesen des Komplexen Preises für Lernobjekte
- 2) Bewertung der Verbräuche im Service-Provisioning-System
- 3) Ausfertigung der Rechnung für den Kunden / Lernenden

## 5 PriMoL – Eine Beschreibungssprache für Preismodelle

Eine kundenorientierte Bepreisung digitaler Güter muss aus Sicht des Anbieters die Bedürfnisse der Kundengruppen, z. B. hinsichtlich Quantität und Qualität der Produkte, berücksichtigen. Eigenschaften von Produkten werden, wie in Abbildung 25 dargestellt, in Produktmodellen abgebildet (Diller 2000). Die getrennte Verbreitung von Preis- und Produktmodellen wird durch das Internet gefördert (Inoue u.a. 2001). Sie bietet insbesondere Vorteile bei der Wartung von komplexen Preismodellen, die in der Mobilkommunikation oder im Application-Service-Providing (ASP) vorherrschen. Für die Entwicklung von Strategien im Pricing, die z. B. die Preissensitivität der Kunden durch Modelle für das Käuferverhalten abbilden, sei auf (Nagle und Holden 1995) und (Monroe 2003) verwiesen. Abbildung 25 beschreibt die zur Einführung von Tarifen erforderlichen Modelle, die im Folgenden kurz beschrieben werden.

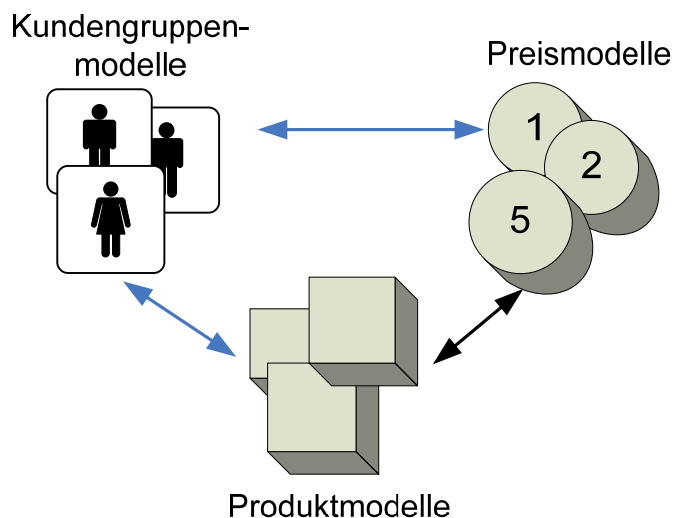


Abbildung 25: Kundenorientierte Preispolitik aus Anbietersicht

### 5.1 Produktmodelle

Eine grundlegende Voraussetzung für eine Bepreisung ist, dass Produkte und Services bzw. Klassen von Produkten und Services als Eigenschaftsbündel in *Produktmodellen* detailliert beschrieben werden (vgl. Kapitel 2.3.6). Produktmodelle für digitale Produkte sind bspw. im *E-Learning* in den Metadaten-Standards für Lernobjekte, z. B. im SCORM-Standard (ADL 2004a),



zu finden. Produktmodelle werden auch bei der Erstellung technischer Produkte verwendet, z. B. im Rahmen des Computer-Aided-Designs (CAD) oder des Computer-Aided-Manufacturing (CAM). Die Verwaltung von Produktmodellen für digitale Produkte wird oft erst durch spezifische Datenmodelle ermöglicht, gerade wenn die Beschreibung bereits wie im Falle von CAD-Daten als eigenes Produkt angesehen werden kann. Um bspw. eine Unterstützung von Produktmodellen für Engineering-Daten durch Datenbanken zu erreichen, müssen Datenbanksysteme im CAD geeignete Datenmodelle bereit stellen, die die Versionierung und Historisierung der Daten, das iterative und explorative Design von Produkten sowie die Verwaltung von Sichten unterstützen (Katz 1990)<sup>97</sup>.

(Dorloff u.a. 2002; Schmitz u.a. 2002) beschreiben *Produktmodelle für digitale Kataloge*, die z. B. zur Beschreibung von Stücklisten verwendet werden. (Schmitz u.a. 2002) geben einen Überblick über XML-Vokabulare zum Austausch digitaler Produkte und Produktdaten.

In *Service modellen* für digitale Dienstleistungen können neben der Auflistung und Beschreibung der zur Verfügung stehenden Dienste, z. B. Lehr-, Lern- und Administrationsprozesse, auch *Service-Levels* (z. B. maximale Antwortzeiten) definiert werden, aus denen im Falle der Nichteinhaltung durch Sanktionsmechanismen Strafkosten, die in Preismodellen definiert wurden, resultieren.<sup>98</sup>

Digitale Produkte sind oft in ihrem Inhalt vordefiniert. Bei digitalen Dienstleistungen findet meist eine Verarbeitung der vom Kunden gelieferten Informationen statt. Im Gegensatz zu traditionellen Dienstleistungen ist ein veränderter Service-Modus zu beobachten, bei dem die „traditionell in den Zuständigkeitsbereich der Anbieter [fallenden Funktionen], wie Beratung, Administration, Spezifikation und Konfiguration von Dienstleistungen“ durch den Kunden übernommen werden. Die eigentliche Dienstleistung jedoch wird nach wie vor durch den Anbieter erbracht.

*„Kennzeichnend für den neuen Service-Modus ist, dass Kunden interaktiv in interne Prozesse und Abläufe der Anbieter einbezogen werden sollen.“*  
(Hanekop u.a. 2001), S. 76

---

<sup>97</sup> Zu CAD-Standards in der industriellen Produktion s. z. B. (Scheer), S. 566 – 568.

<sup>98</sup> Eine Sprache zur Definition von Service-Level-Agreements für elektronische Dienste wird bspw. in (Ludwig u.a. 2003) vorgestellt.

Im Folgenden wird angenommen, dass Produkt- und Servicemodelle (kurz Produktmodelle) die Eigenschaftsdimensionen und –ausprägungen von Produkten und Leistungen beschreiben. Diese lassen sich tabellarisch abbilden und mit Hilfe relationaler Strukturen verwalten. „[Preis-]Verhandlungsprozesse können [dann] über verschiedene Produkteigenschaften ermöglicht oder auf ein einzelnes Attribut wie den Preis beschränkt werden.“ (Weinhardt u.a. 2003)<sup>99</sup>

## 5.2 Kundengruppenmodelle

*Kundengruppenmodelle*<sup>100</sup> ermöglichen aus konzeptueller Sicht die Definition unterschiedlicher Kundengruppenprofile. Für eine Hochschule, die E-Learning-Kurse anbietet, können bspw. die Kundengruppen regulärer Student, Student der Berufsakademie oder institutioneller Nachfrager definiert werden. Mit der Definition von Kundengruppenprofilen werden unterschiedliche Ziele verfolgt. Kundendaten werden u. a. zur *Kundensegmentierung* genutzt und dienen so als Basis für die Preis- und Produktdifferenzierung, die in Kapitel 2.4 beschrieben wurden. Werden Nutzungsdaten zu den Kunden, die in Kundengruppenmodellen definiert sind, gespeichert, so können bspw. auch *Collaborative-Filtering-Mechanismen* angewendet werden (Stegmann u.a. 2003). Diese dienen dazu, den Kunden Produkte anzubieten, die durch „ähnliche“ Kunden bereits erworben wurden. Auch die Personalisierung von Anwendungen und die Zugriffssteuerung (Autorisierung und Authentifizierung) auf Produkte, z. B. als *Digital-Rights-Management* (DRM), werden durch die Einführung von Kunden- und Kundengruppen<sup>101</sup> möglich (vgl. (Sandhu u.a. 1996)).

*„Im Falle der Personalisierung von Medienprodukten wird das [...]Kernproblem flexibler Fertigungsstrukturen ersetzt durch die*

---

<sup>99</sup> Analog (Moog 1998), Kapitel 5.11, S. 94-97: „[...] [P]otentielle Käufer entscheiden über den Abschluß eines Absatzvertrages, indem sie Leistung und Entgelt gegeneinander abwägen. Auch wenn sie dies aus ihrer jeweiligen Sicht unter Berücksichtigung individueller Alternativen tun, legen sie dabei realökonomische Entgelt- und Leistungsmerkmale zugrunde. [...] Ohne die umgangssprachliche Begriffsabgrenzung für die beiden Hauptgruppen Güter [Produkte, d. Verf.] und Dienstleistungen zu hinterfragen, werden daher hier nur die Quellen der *Mehrdimensionalität* der *Merkmale* gegen Entgelt erbrachter Leistungen skizziert.“

<sup>100</sup> In der Arbeit wird der Begriff Kundengruppe verwendet. Unter einer Kundengruppe kann ohne Verlust der Allgemeingültigkeit auch ein einzelner Kunde betrachtet werden, sofern sich keine weitere Person in der Gruppe befindet.

<sup>101</sup> Bei der Erstellung von Informationssystemen werden i. d. R. die Begriffe „Benutzer“ und „Benutzergruppe“ verwendet.

*Notwendigkeit einer automatischen Integration der Kundendaten in den Bündelungs- und Vertriebsprozess.“ (Kaspar und Hagenhoff 2003), S.20*

Im Marketing werden Kundendaten zum *Kundenbeziehungsmanagement* (CRM; Customer-Relationship-Management) verwendet.<sup>102</sup>

Obwohl Kunden(gruppen) die Basis zur Preisdifferenzierung bilden, sollten Regeln zur Preisdifferenzierung durch Einführung von Rabatten und Aufschlägen (Konditionenpolitik) nicht im Kundengruppenmodell fixiert werden, sondern sind Bestandteile von Preismodellen.

Im Rahmen dieser Arbeit wird davon ausgegangen, dass Anbieter digitaler Produkte und Leistungen ihre Kunden in Kundengruppenmodellen beschreiben und / oder diese zumindest eindeutig identifizieren können. Darüber hinaus wird angenommen, dass Produkte bzw. Produktgruppen ebenfalls eindeutig durch den Anbieter identifiziert werden können und – sofern für die Bepreisung erforderlich – die Eigenschaften der Produkte und Leistungen in einem Eigenschaftsraum erfasst worden sind.<sup>103</sup>

## 5.3 Preismodelle

### 5.3.1 Überblick

Zur Bepreisung von digitalen Gütern können analog zur Bepreisung von Telekommunikationsdienstleistungen *Tarife* definiert werden, die mit *Preismodellen* beschrieben werden sollen. Diese knüpfen nach (Diller 2000) Preise an „bestimmte Anwendungsbedingungen [...], die dann [...] häufig differenziert werden.“ „Der Preis stellt [dabei] die Summe aller mittelbar oder unmittelbar mit dem Kauf eines Produktes verbundenen Ausgaben eines Käufers dar. Dabei lassen sich [in Preismodellen] negative und positive Preiskomponenten unterscheiden, die sich über den gesamten Gebrauchszyklus eines Produktes hin zum effektiven Gesamtpreis saldieren“ (Diller 2000), S. 25. Gegenstand der Bepreisung kann das Nutzungsverhalten der angebotenen Dienstleistungen sein. Zudem kann eine leistungsunabhängige Bepreisung beispielsweise in Form von

---

<sup>102</sup> Zum Kundengruppenmanagement, s. z. B. (Kumar und Reinartz 2006).

<sup>103</sup> Produktalternativen können in einem n-dimensionalen Eigenschaftsraum dargestellt werden. Dabei werden in den Dimensionen des Eigenschaftsraumes kontinuierliche und diskontinuierliche Produkteigenschaften abgetragen.

Teilnahmegebühren, monatlichen Grundgebühren oder Gebühren für den Erwerb von Produkten definiert werden.<sup>104</sup>

*Preismodelle* bilden Auswertungsvorschriften für die Bepreisung von Produkten und Dienstleistungen ab, indem sie Preisähler für Bezugsobjekte, d.h. Produkte und Leistungen bzw. Teile dieser (Preisnenner), definieren.

Eine (Gegen-)Leistung kann auch darin bestehen, dass bspw. eine Inanspruchnahme eines Dienstes oder der Download eines Produktes überhaupt möglich ist. Im Falle von Versicherungen handelt es sich bei der Gegenleistung um ein Leistungsversprechen im vorher definierten Leistungsfall (Erleben, Eintritt eines Schadens).

Im Folgenden werden Preismodelle aus unterschiedlichen Branchen dargestellt, die neben dem E-Learning als Basis der Entwicklung der Beschreibungssprache PriMoL dienen. Kapitel 5.3.2 widmet sich der Bepreisung von Versicherungsprodukten. Im Kapitel 5.3.3 werden aktuelle Tarife im Mobilfunk untersucht.

### 5.3.2 Pricing von Versicherungsprodukten

Im Rahmen des Teilprojekts „Flexible Abrechnungssysteme“ des BMBF-Projektes PREMIUM (Preis- und Erlösmodelle im Internet, Umsetzung und Marktchancen) wurden auch Kfz-Versicherungen als digitale (bzw. digitalisierbare) Produkte untersucht.<sup>105,106</sup> Versicherungen sind als Rechtstitel gemäß Kapitel 2.2.1 einzuordnen. In der Phase der Kundengewinnung können sie als reine Informationsprodukte (vgl. Kapitel 2.2.2) und somit digitalisierbare Produkte betrachtet werden (vgl. (Bechmann und Huhn 1999), S. 15). Dabei sind gerade Kfz-Versicherungen nach (Kirby und Dally 2000), S. 35 besonders für die Distribution über das Internet geeignet, da sie als Produkt sehr einfach gestaltet

---

<sup>104</sup> Vgl. z. B. auch (Reichl u.a. 2003): „Usually delivering a service may be divided conceptually to delivering [...] service units, each of which may have an individual (unit) price. The *tariff* describes the mapping of these prices [...] to the resulting charge. [...] Hence a tariff T may be described as a [...] matrix of mappings from (multi-dimensional) input-parameters to (multi-dimensional) output parameters, the so-called Tariff Matrix [...] Tariffs in general are time-dependent, i.e. may vary over time. [...] Tariffs in general can be assumed to be utilization-sensitive. [...] Tariffs also may depend on further input parameters, either pre-set parameters, output-parameters [from other time-scales], or results from measurements.”

<sup>105</sup> Die Erkenntnisse basieren auf Experteninterviews mit einer großen deutschen Versicherung und einer daraufhin an der Universität Karlsruhe durchgeführten Diplomarbeit.

sind und nur über eine eingeschränkte Anzahl an buchbaren Zusatzoptionen verfügen.<sup>107</sup> *Versicherungen* können definiert werden als eine Vereinbarung über die gegenseitige wirtschaftliche Hilfeleistung zwischen den Mitgliedern einer Gemeinschaft bei Eintritt bestimmter Gefahren. Im Versicherungsfalle hat der Einzelne einen Rechtsanspruch auf die vereinbarte Leistung laut Versicherungsvertrag (vgl. (Bodendorf und Robra-Biisantz 2003), S. 273; (Schierenbeck 1990), S. 654). Die Verträge der Versicherungsnehmer werden in *Portfolios* (Kollektiv) zusammengefasst, die im Idealfall Gruppen stochastisch unabhängiger Risiken umfassen. Ein Ausgleich der Einzelrisiken ergibt sich daraus, dass sich die Schadenunterschreitungen bzw. Schadenüberschreitungen bezüglich des Schadenserwartungswertes im Kollektiv nivellieren.<sup>108</sup>

### 5.3.2.1 Kraftfahrtversicherung<sup>109</sup>

Die *Kraftfahrtversicherung* deckt die mit dem Gebrauch eines Kraftfahrzeugs zusammenhängenden Gefahren ab und dient zugleich dem Schutz des Verkehrsopfers. Zur Kraftfahrtversicherung gehören Kfz-Haftpflichtversicherung, Fahrzeugversicherung (Teilkasko und Vollkasko) und Insassenunfallversicherung (im Folgenden nicht behandelt).

### 5.3.2.2 Tarifierung und Tarifmerkmale der Kfz-Haftpflichtversicherung

Die *Kraftfahrzeug-Haftpflichtversicherung* dient der Befriedigung begründeter und Abwehr unbegründeter Ansprüche bis zur gesetzlichen Mindestdeckungssumme, die auf Grund gesetzlicher Haftpflichtbestimmungen gegen den Versicherungsnehmer oder mitversicherte Personen erhoben werden.

---

<sup>106</sup> Zur Digitalität von Versicherungen vgl. (Peterson u.a. 1997), S. 334.

<sup>107</sup> Der *Versicherungsschutz* ist nicht gegenständlich und daher hinsichtlich seiner Produktion nahezu unendlich. Die versprochene Leistung wird über eine lange Zeit nicht oder nicht immer realisiert und deckt daher einen nur schwer erfassbaren Zukunftsbedarf (kein Positivnutzen sondern nur Ausgleich eines negativen Ereignisses).

<sup>108</sup> Der Ausgleich im Kollektiv ist gewährleistet, falls ein homogenes Kollektiv vorliegt, d.h. alle einzelnen Risiken eines Kollektivs hinsichtlich des von ihnen zu erwartenden Schadens gleich hoch sind und die Risiken der Versicherungsnehmer stochastisch voneinander unabhängig sind.

<sup>109</sup> Die Kfz-Versicherung umfasst zirka 40% des Beitragsvolumens der Schaden- und Unfall-Versicherungsunternehmen. Nach (Koch 1998), S. 301 ist die Kraftfahrtversicherung (im Folgenden Kfz-Versicherung) gemessen am Prämienaufkommen die zweitgrößte Versicherungssparte nach der Lebensversicherung.

Versicherungsunternehmen können die Tarife in der Kfz-HP frei kalkulieren.<sup>110</sup> Viele Versicherungsunternehmen orientieren sich dabei an den Tarifempfehlungen des *Gesamtverband der Deutschen Versicherungswirtschaft* (GDV), in dessen Gemeinschaftsstatistik Empfehlungen zur Höhe der Nettorisikoprämie unter Berücksichtigung der fünf primären Tarifmerkmale *Kraftfahrzeugtyp*, *Regionalklasse*, *Schadenfreiheitsklasse*, *Jahres-Kilometer-Leistung* und *Garagennutzung* gegeben werden.<sup>111</sup> Die Tarifmerkmale werden „als Basis für [die] Tarifikalkulation verwendet und erzeugen den größten Teil der Prämiendifferenzierung, wozu das Merkmal Kfz-Typ am stärksten beiträgt. [...] Versicherungsunternehmen verwenden teilweise (z. B. bei den Merkmalen Region und Jahres-km-Leistung) eigene Merkmalsklassen. [...] Einige Merkmale werden aber offensichtlich ohne genaue Kalkulation verwendet (z. B. Mitgliedschaft in einem Umweltschutz-Verein, BahnCard-Besitz). Dazu gehört auch der ‚Abwerberabatt‘, der darin besteht, dass dem Versicherungsnehmer eine Prämie angeboten wird, die um einen bestimmten Betrag unter der Prämie seines Vorversicherers liegt.“ (Korschenewski 2005), S. 39-40.

### 5.3.2.3 Tarifierung und Tarifmerkmale der Kasko-Versicherung

Die *Kaskoversicherung* ist durch folgende Merkmale gekennzeichnet:

- Sie deckt das Interesse des Versicherungsnehmers am Werterhalt des Kfz.
- Die *Teilkaskoversicherung* / *Fahrzeugteilversicherung* deckt Schäden durch folgende Gefahren: Brand und Explosion, Entwendung, Elementarschäden durch Sturm, Hagel, Blitzschlag und Überschwemmung, Wildschaden, Glasbruch sowie Schmorschäden an der Verkabelung durch Kurzschluss.
- Die *Vollkaskoversicherung* deckt zusätzlich Schäden durch Unfall, mut- und böswillige Beschädigung (Vandalismus) und Reifenschäden, sofern durch das Schadenereignis gleichzeitig auch andere ersatzpflichtige Schäden am Fahrzeug entstanden sind.

Eine Tarifierung beruht meist auf den Tariffaktoren Typklasse, Regionalklasse,

---

<sup>110</sup> Ethnische Herkunft und Staatsangehörigkeit dürfen jedoch nicht in der Tarifierung verwendet werden.

<sup>111</sup> Die Versicherungsunternehmen sind gesetzlich verpflichtet ihre Tarifierungsdaten an den GDV jährlich zu übermitteln. Für eine detaillierte Übersicht über die genannten Merkmale s. unter [www.gdv.de](http://www.gdv.de).

Schadenfreiheitsklasse, Fahrleistung, Abstellplatz und Höhe der Selbstbeteiligung.<sup>112</sup>

#### 5.3.2.4 Prämiendifferenzierung<sup>113</sup>

Eine *Prämiendifferenzierung* ist insbesondere in der Kfz-HP wichtig, da i. d. R. kein homogener Gesamtbestand vorliegt. Aus Gründen der *Prämienstabilität*<sup>114</sup> und der Risikoselektion ist es nicht ratsam, für alle einzelnen Risiken des Gesamtbestandes eine einheitliche Prämie festzulegen (vgl. (Mack 2002), S. 352-357). Das Ziel ist es, eine risikogerechte Prämie entlang der Merkmale zu berechnen (vgl. (Meyer 1999), S. 10-11).

Gerade die *Risikoselektion* durch die Versicherungsnehmer erfordert eine Differenzierung. Sobald ein Unternehmen Prämien differenziert, erleiden alle Unternehmen, die dieser Maßnahme nicht folgen, Wettbewerbsnachteile, da sich „schlechtere Risiken“ Versicherer suchen, die günstigere, d.h. weniger differenzierte Tarife anbieten.<sup>115</sup>

#### 5.3.2.5 Tarifmerkmalsbestimmung

Mit Hilfe von *Risikomerkmale*<sup>116</sup> kann eine Prämiendifferenzierung vorgenommen werden. Dazu werden die „Risiken“ in Gruppen mit unterschiedlichem Schadenerwartungswert aufgeteilt. Durch Auswahl geeigneter Risikomerkmale kann ein Kfz-HV-Tarif strukturiert werden. Um eine Tarifierung zu ermöglichen, müssen Klassen für Merkmalsausprägungen gebildet werden, um für die Kombination der Merkmalsausprägungen eine *Prämie* zu definieren. Merkmale, die bei der Aufstellung eines Tarifs Verwendung finden, werden *Tarifmerkmale* genannt. Die Auswahl der Tarifmerkmale kann z. B. mit Hilfe ökonomisch-statistischer Methoden getroffen werden. Mit Hilfe der *Regressionsanalyse* lassen sich nach Vorgabe der funktionalen Form, z. B.

---

<sup>112</sup> Je nach Versicherungsunternehmen kommen weitere Rabatffaktoren (Angestellter des öffentlichen Dienstes etc.) zum Tragen.

<sup>113</sup> Prämie = Preis einer Versicherung.

<sup>114</sup> Ein prämiensstabiler Tarif muss nicht aus Gründen einer geänderten Bestandszusammensetzung neu berechnet werden.

<sup>115</sup> „Der Wettbewerb führt also dazu, dass dort, wo ein Unterschied bzgl. des Schadenerwartungswertes zwischen verschiedenen Gruppen zu erkennen ist, eine Prämiendifferenzierung erfolgt.“ (Korschenewski 2005), S. 48.

<sup>116</sup> Nach (Mack 2002), S. 122-128 steht ein Risikomerkmale für eine Eigenschaft des zu versichernden Risikos, für das gilt: (1) Korreliertheit mit dem Schaden, z. B. Jahres-

$$E[S_r] = f(r) = a_0 + a_1 r_1 + a_2 r_2 + \dots + a_n r_n$$

oder

$$E[S_r] = f(r) = b_0 + r_1^{b_1} + r_2^{b_2} + \dots + r_n^{b_n},$$

die Koeffizienten  $a_0, a_1, \dots, a_n$  bzw.  $b_0, b_1, \dots, b_n$  für den erwarteten Schadenswert  $E[S_r]$  bestimmen.<sup>117</sup> Abbildung 26 gibt den Vorgang der Ermittlung von Tarifmerkmalen als CP-Netz wieder.

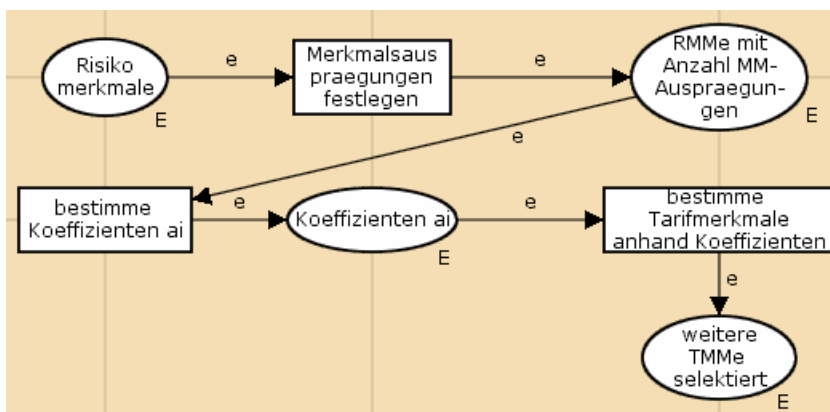


Abbildung 26: CP-Netz zur Ermittlung von Tarifmerkmalen

### 5.3.2.6 Kalkulation von Tarifen mit Tariffaktoren

Nach der Bestimmung der Tarifmerkmale muss für jede mögliche Kombination von Tarifmerkmalsausprägungen eine Prämie festgesetzt werden. Dabei entstehen leicht Probleme durch die Vielzahl von möglichen Kombinationen und damit festzulegenden Prämien.<sup>118</sup> Zur Vereinfachung der Kalkulation werden häufig *Tariffaktoren* verwendet.<sup>119</sup> Das nachfolgende Beispiel beschreibt das Verfahren:<sup>120</sup>

Grundprämie ( $p$ ) = 100

Kilometerleistung, (2) einfache Feststellbarkeit, z.B. Typ des Kfz sowie (3) Nicht-Manipulierbarkeit.

<sup>117</sup> Dabei bezeichnet  $a_0$  bzw.  $b_0$  die Konstante der Regressionsfunktion, und  $a_i$  bzw.  $b_i$  sind die Regressionskoeffizienten zur Abbildung des Einflusses des  $i$ -ten Risikomerkmals. Entsprechend sind die Risikomerkmale  $r_1, r_2, \dots, r_n$ . Vgl. (Hartung und Elpelt 1999), S.77-142.

<sup>118</sup> Für die GDV-Tarifmerkmale gibt es alleine 55.680 unterschiedliche Kombinationen.

<sup>119</sup> Dabei wird ausgehend von einer Grundprämie je nach Tariffaktoren für jedes Tarifmerkmal zur Grundprämie ein Zuschlag bzw. ein Abschlag angewendet. Für die Ermittlung von Tariffaktoren existieren unterschiedliche statistische Verfahren. Neben dem Simon-Bailey-Verfahren kommt bspw. das Marginalsummenverfahren (vgl. (Meyer 1999), S. 19-20; (Mack 2002), S. 168-172) zum Einsatz.

<sup>120</sup> Vgl. (Korschenewski 2005), S. 55-58.



Tarifmerkmale: Jahresfahrleistung (fünf Klassen), Geschlecht des Kfz-Halters (zwei Klassen).

Tarifmerkmal Jahresfahrleistung		
Merkmalsausprägung	Jahresfahrleistung in km	Tariffaktor
Klasse 1	bis 5.000	$x_1=0,80$
Klasse 2	bis 10.000	$x_2=0,85$
Klasse 3	bis 15.000	$x_3=1,00$
Klasse 4	bis 20.000	$x_4=1,10$
Klasse 5	ab 20.001	$x_5=1,25$

*Tabelle 6: Tarifmerkmal Jahresfahrleistung*

Tarifmerkmal Geschlecht	
Merkmalsausprägung	Tariffaktor
Weiblich	$y_1=0,90$
Männlich	$y_2=1,05$

*Tabelle 7: Tarifmerkmal Geschlecht*

Tariffunktion:  $p_{ij} = p \cdot x_i \cdot y_j$ , mit  $p_{ij}$  als Prämie für die Faktorkombination  $ij$ .

		Tarifmerkmal Geschlecht		
		Tariffaktoren		
			$y_1=0,90$	$y_2=1,05$
Tarifmerkmal Jahresfahr- Leistung	Klasse 1	$x_1=0,80$	72,00	84,00
	Klasse 2	$x_2=0,85$	76,50	89,25
	Klasse 3	$x_3=1,00$	90,00	105,00
	Klasse 4	$x_4=1,10$	99,00	115,50
	Klasse 5	$x_5=1,25$	112,50	131,25

*Tabelle 8: Prämienmittlung bei zwei Tarifmerkmalen*

### 5.3.3 Pricing digitaler Produkte im Mobilfunk

Unter *Mobilfunk* werden drahtlose Weitverkehrsnetze (WAN) zusammengefasst. Zu den *Mobilfunkdiensten* zählen Dienste zur Übertragung von Sprache und

Daten. Voraussetzung für Mobilfunk ist die Verfügbarkeit eines *Mobilfunknetzes*, das die technische Infrastruktur zur Übertragung von Signalen bereitstellt (vgl. (Turowski und Pousttchi 2004), S. 8). Das Mobilfunknetz setzt sich aus dem *Mobilvermittlungsnetz* (Core-Network) zur Übertragung der Signale zwischen den ortsfesten Einrichtungen und dem *Zugangsnetz* (Radio-Access-Network) zur Übertragung der Signale zwischen Mobilfunkantennen und dem mobilen Endgerät zusammen. Neben der Art des Dienstes ist für die Berechnung von Gebühren die Lokation des Anrufers und des Angerufenen wichtig. Dazu wird durch Mobilfunknetze auch das so genannte *Roaming* unterstützt.<sup>121</sup>

### ***In Deutschland verwendete Mobilfunkstandards***

In Deutschland werden digitale *Mobilfunknetze der zweiten Generation* nach dem GSM 900- und dem DCS 1800-Standard betrieben.<sup>122,123</sup> Die so genannte 2,5te Generation erweitert diese Mobilfunknetze um Funktionen zur Datenübertragung. Hierzu gehört auch die Übertragung per *GPRS* (General Packet Radio Service) (vgl. z. B. (Aydinli 2002) S. 14-15).

Das *Global System for Mobile Communication* (GSM) ist in das *Base-Station-Subsystem* (BSS), das *Network-and-Switching-Subsystem* (NSS) und das *Operation-and-Support-Subsystem* (OSS) aufgeteilt. Das BSS umfasst die Basisstationen, die den Kunden den Zugang zum Netz ermöglichen. Im NSS werden Nutzerdaten innerhalb des Netzes weitergeleitet und diese anderen Netzen zur Verfügung gestellt. Im OSS werden die Dienste des Netzes verwaltet (vgl. (Rudolf 2003) , S. 28-34).

Nach (Lehner 2003), S. 38-39 können drei Dienstarten in GSM-Netzen unterschieden werden: (1) *Trägerdienste*, die die Parameter des einzelnen eingerichteten Sendekanals bestimmen.<sup>124</sup> (2) *Teledienste* umfassen Kernfunktionen wie Telefondienst, Notrufdienst und SMS (3) *Zusatzdienste* zur Identifikation des Anrufers etc.

Das *Universal Mobile Telecommunications System* (UMTS) erfordert im Gegensatz zu GPRS die Einrichtung eines neuen Zugangsnetzes.<sup>125</sup> Neben einer

---

<sup>121</sup> Roaming meint die Nutzung eines fremden Mobilfunknetzes, sowohl im Inland (National-Roaming), als auch im Ausland (International Roaming).

<sup>122</sup> Die erste Mobilfunknetzgeneration verwendete analoge Übertragungstechnik.

<sup>123</sup> Nach (Lehner 2003), S. 35-37 nutzten im Februar 2003 weltweit zirka 780 Mio. Mobilfunk-Kunden den GSM-Standard. Dies entsprach 69% der Mobilfunknutzer.

<sup>124</sup> Hierzu gehören u. a. die Art und Kapazität der Übertragung.

<sup>125</sup> Das UMTS Terrestrial Radio Access Network (UTRAN).

hohen Bandbreite, die die Unterstützung von Multimedia-Anwendungen ermöglicht, zeichnet sich UMTS durch eine paketorientierte Vermittlung aus.<sup>126</sup>

Eine differenzierte, volumenabhängige und zeitabhängige Bepreisung der Dienstenutzung ist auf Basis der Dienstprofile, die in Tabelle 9 dargestellt sind, möglich.

Dienstprofil	Datenrate	Übertragungsrate	Beispiel
High Interactive Multimedia	128 kBit/s	leitungsvermittelt	Videokonferenzen
High Multimedia	2,0 MBit/s	Paketvermittelt	Audio-/Videoklips
Medium Multimedia	384 kBit/s	leitungsvermittelt	Online-Baking
Switched Data	14,4 kBit/s	leitungsvermittelt	Fax
Simple Messaging	14,4 kBit/s	Paketvermittelt	SMS, Email
Sprache	16,0 kBit/s	leitungsvermittelt	Telefonieren

**Tabelle 9: Dienstprofile bei UMTS<sup>127</sup>**

### 5.3.3.1 Tarifmodellierung<sup>128</sup>

Mobilfunkdienste können *nach der Art der Kommunikation* unterschieden werden. Hierzu gehören Events (SMS, MMS etc.), Sprachübertragung und Datenübertragung. Im Folgenden werden kurz die Möglichkeiten eines Pricings für diese Dienste beschrieben:

- *Events* können pro Stück abgerechnet werden. Der Preis hängt in der Regel vom Empfangsnetz ab.
- Die Abrechnung der *Datenübertragung* (WAP etc.) kann sowohl verbindungsorientiert erfolgen (z. B. im GSM-Netz), als auch paketorientiert (ab 2,5ter Generation). Preise können in Abhängigkeit von der Übertragungsqualität variieren.
- *Sprachdienste* sind im Gegensatz zu den anderen Diensten besonders stark preisdifferenziert. Beispiele für die Tarifierung von Sprachdiensten werden im Folgenden vorgestellt.<sup>129</sup>

<sup>126</sup> UMTS umfasst die gleichen Dienstarten wie GSM. „Deren Leistungsspektrum ist aber [...] etwas höher und differenzierter.“ (Vugdalic 2004), S. 182.

<sup>127</sup> Quelle: modifiziert nach (Vugdalic 2004), S. 20.

<sup>128</sup> Die Darstellung folgt im Wesentlichen (Vugdalic 2004), S. 54-58.

<sup>129</sup> Es wurden lediglich „Vertragstarife“ mit gleichzeitigem Erwerb eines Endgeräts untersucht, die eine langfristige Bindung des Kunden an den Mobilfunkbetreiber, z. B. über einen Zeitraum von 24 Monaten, voraussetzen.

### 5.3.3.2 Tarife für Sprachdienste

Mobilfunkanbieter differenzieren ihre Angebote nach Kundengruppen (Privat- und Geschäftskunden). Tarife unterscheiden sich vor allem hinsichtlich der Höhe einer monatlich zu entrichtenden Grundgebühr (Bestandspreis) und gesprächsabhängigen Nutzungspreisen. Darüber hinaus bieten Provider wie E-Plus ihren Geschäftskunden die Möglichkeit, Mehr-Personen-Angebote zu nutzen, bei denen ein Bestandspreis pro zusätzliche Kundennummer abgerechnet wird. Für Privattarife lassen sich nach (Vugdalic 2004), S. 56 „zwei grundsätzliche Tarif-Typen“ identifizieren. Neben den zuvor genannten zweiteiligen Tarifen, existieren Tarifvarianten, bei denen der Mobilfunknutzer Minutenpakete erwerben muss. Nach Verbrauch des Minutenpakets fallen zusätzliche nutzungsabhängige Gebühren an, die „dann wesentlich höher als die Minutenpreise im Paket sind.“ Neben den von Vugdalic beschriebenen Tarifen vertreibt der Mobilfunkprovider Base ([www.base.de](http://www.base.de)) einen Tarif, der eine Flatrate für Sprachdienste umfasst. Dabei umfasst die Flatrate jedoch nur Telefonate innerhalb des eigenen Netzes (E-Plus) und in das deutsche Festnetz. Hinzu kommen Tarife, bei denen die Gesprächseinheit nach Lokation des Mobilfunknutzers variiert (z. B. Homezone im O2-Genion-Tarif).

Um einen „passenden“ Tarif zu finden, muss ein potenzieller Kunde seine Nutzungsgewohnheiten (Dauer und Lokation der Telefonate, heruntergeladene Bytes, prozentuale Verteilung der eigenen Anrufe über unterschiedliche Netzbetreiber und Tageszeiten etc.) genau kennen. Selbst wenn der Kunde seine derzeitigen Gewohnheiten genau kennt, so können diese über den Vertragszeitraum variieren.

## 5.4 PriMoL – eine Beschreibungssprache für Preismodelle

### 5.4.1 Sprachkonzepte

Die Beschreibungssprache besteht aus drei Grundkonstrukten, den so genannten *einfachen Preisen*, *komplexen Preisen* und *Aggregationsregeln*. Zur Erläuterung der Konstrukte werden in diesem Kapitel Beispiele aus dem E-Learning herangezogen. Die Beschreibungssprache ist jedoch domänenunabhängig und kann daher analog auch für die Beschreibung von Preismodellen für Telekommunikationsdienste oder zur Bepreisung von Versicherungs-

dienstleistungen verwendet werden. Die Konstruktion<sup>130</sup> der Beschreibungssprache als Metamodell basiert auf Untersuchungen unterschiedlicher Preismodelle für digitale Produkte. Um den Untersuchungsbereich einzuengen wurden drei Produktkategorien ausgewählt. Neben Preismodellen im E-Learning, wurden insbesondere Preismodelle im Versicherungsbereich (Kapitel 5.3.2) und in der Telekommunikation (Kapitel 5.3.3) untersucht. Tarife in den vorgenannten Anwendungsbereichen zeichnen sich dadurch aus, dass der Preis an eine Vielzahl von Parametern (Dimensionen) gebunden ist.

#### **5.4.1.1 Einfache Preise**

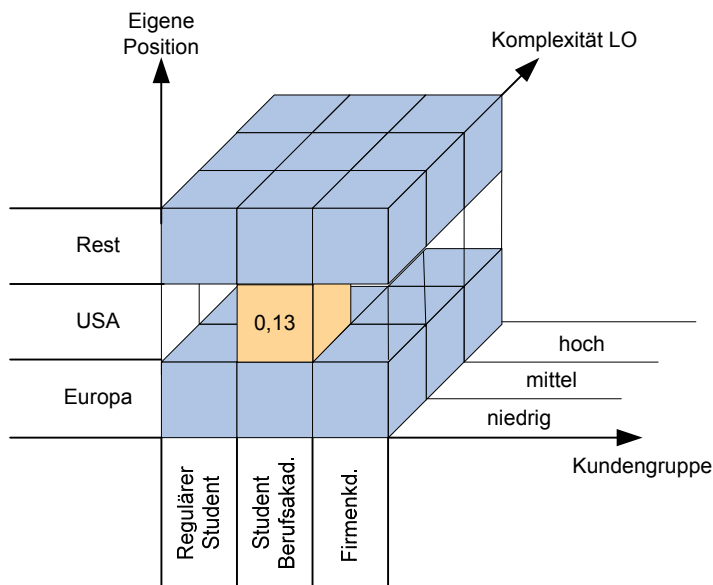
*Einfache Preise* fallen in der Regel nur einmalig für den Bezug eines Produktes oder die Nutzung eines Dienstes an oder werden in regelmäßigen Abständen erhoben. Beispiele für einfache Preise im E-Learning sind Rückmeldegebühren pro Semester, Immatrikulationsgebühren und Teilnahmegebühren für Kurse. Für die Nutzung von Telekommunikationsdiensten (z. B. für DSL oder Telefonie) fallen einfache Preise in Form von Grundgebühren an. Einfache Preise sind Event-gebunden, d.h. die Verrechnung eines durch einen einfachen Preis angegebenen Preiszählers wird durch ein Ereignis ausgelöst, das den Eintritt eines Zeitpunktes (bspw. jeweils am ersten Tage eines Monats) oder die Aktion des Kunden markiert. Das Event „Wintersemester 2005 hat begonnen“ oder „neuer Abrechnungszeitraum“ (Zeitfenster) sind Beispiele für zeitpunktbezogene Events. Im Gegensatz dazu wären „Kunde hat sich für den Kurs X angemeldet“ oder „Kunde hat Software Y herunter geladen“ Beispiele für nicht zeitlich gebundene Ereignisse, die durch Aktionen des Kunden ausgelöst werden. Einfache Preise eignen sich im E-Learning auch dazu, individuelle Service-Leistungen, wie z. B. die Korrektur einer Probeklausur durch Mitarbeiter einer Professur, abzurechnen. Einfache Preise sollten immer dann zur Modellierung verwendet werden, wenn die Berechnung eines Preises nur durch ein Event ausgelöst wird und nicht von weiteren Parametern abhängt.

---

<sup>130</sup> Bei einer Konstruktion werden gemäß (Holten 2001), S. 2 „sprachliche Mittel“ normiert. Sie basiert auf einer Abstraktion der Wahrnehmung, so dass nur die als relevant empfundenen Konstrukte im Metamodell der Beschreibungssprache für Preismodelle verankert werden. „Eine wesentliche Konsequenz des konstruktionsorientierten Modellierungsverständnisses besteht darin, dass niemals gerechtfertigt werden kann, dass ein Modell den dargestellten Sachverhalt richtig in dem Sinne abbildet, dass Komponenten des Originals den Modellkomponenten fehlerfrei zugeordnet sind.“ (Holten 2001), S. 2

### 5.4.1.2 Komplexe Preise

Das Modellierungskonstrukt *komplexer Preis* ermöglicht die Bestimmung von Preisen aus einer oder mehreren beobachtbaren Merkmalsdimensionen. Als Merkmalsdimensionen kommen im Falle der Abrechnung von Lernobjekten durch einen E-Learning-Provider z. B. die Kundengruppe des Lernenden (regulärer Student, Student der Berufsakademie, Firmenkunde), der Ort des Lernenden (Europa, USA, Rest) und die Qualität des nachgefragten Lernobjekts (hoch, mittel, niedrig) infrage (vgl. Abbildung 27).



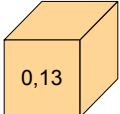
Konzept	Darstellung im Diagramm
Dimensionen / Merkmale	Eigene Position Komplexität LO Kundengruppe
Skalenniveau (Dimension Komplexität LO)	hoch mittel niedrig
Arrayfeld mit Feldwert	

Abbildung 27: Wertearray für drei Merkmalsdimensionen

Ein komplexer Preis wird ausgehend von den Ausprägungen der Merkmalsdimensionen berechnet oder kann durch Ablesen eines Wertes in einem multidimensionalen Array ermittelt werden. Im Falle der Abfrage eines Lernobjekts von geringer Komplexität durch einen Studierenden der Berufsakademie, der in den USA ansässig ist, kann der Preis des Lernobjekts als Feldwert des in Abbildung 27 dargestellten Dimensionswürfels abgelesen werden. Der Student würde für die Abfrage des Lernobjekts € 0,13 in Rechnung gestellt bekommen. Der Array-Index zum ermittelten Wert entspricht dabei der Kombination der Skalenniveaus der Merkmalsdimensionen bzw. Kategorien. Der Index für den oben beschriebenen Wert 0,13 lautet: [Eigene Position = "USA"; Kundengruppe = "Student Berufsakademie"; Komplexität = "niedrig"].

Als Alternative zum Ablesen von komplexen Preisen in dem durch die Merkmalsdimensionen „Eigene Position“, „Kundengruppe“ und „Komplexität“ aufgespannten Wertearray, können diese auch durch Berechnungsvorschriften ermittelt werden. Hierzu müssen den einzelnen Skalenniveaus der Merkmalsdimensionen, wie in Tabelle 10 verdeutlicht, Werte zugewiesen werden.

<b>Eigene Position</b>	<b>Europa</b>	<b>USA</b>	<b>Rest</b>
	1,00	0,90	0,80

<b>Komplexität Lernobjekt</b>	<b>niedrig</b>	<b>mittel</b>	<b>Hoch</b>
	0,096	0,120	0,138

<b>Kundengruppe</b>	<b>regulärer Studierender</b>	<b>Studierender der BA</b>	<b>Firmenkunde</b>
	1,00	1,50	2,00

**Tabelle 10: arithmetische Bestimmung eines Preises**

Ein komplexer Preis kann dann durch Angabe einer mathematischen und/oder logischen Vorschrift ermittelt werden. Für die Abfrage eines Lernobjektes niedriger Qualität durch einen Studierenden der Berufsakademie in den USA würden bei multiplikativer Verknüpfung der Merkmalsdimensionen  $0,9 * 0,096 * 1,5 = 0,1296 \sim € 0,13$  an Gebühren anfallen. Die Zahlen in Tabelle 10 können so gedeutet werden, dass für Kunden aus den USA ein Abschlag von 10% gewährt wird.

In den aufgeführten Beispielen wurden bisher lediglich qualitative Merkmalsdimensionen betrachtet. Damit metrische Merkmale (verhältnis- und intervallskalierte), z. B. das Downloadvolumen in Kilobytes, für die Indizierung von Feldwerten eines Arrays verwendet werden können, müssen diese in eine diskrete Form gebracht werden. Bei qualitativen, d.h. komparativen<sup>131</sup> (z. B. Komplexität des Lernobjekts) und klassifikatorischen<sup>132</sup> Merkmalen (z. B. Kundengruppe), müssen die einzelnen Werte zu den Skalenniveaus manuell erfasst werden.<sup>133</sup>

Komplexe Preise eignen sich zur *Preisdifferenzierung*. Im obigen Beispiel wurde z. B. eine räumliche Preisdifferenzierung durch das Merkmal „Eigene Position“ und eine gruppenspezifische Preisdifferenzierung über das Merkmal „Kundengruppe“ eingeführt. Zur Definition von Berechnungsvorschriften sind in der Beschreibungssprache und im Prototyp einfache arithmetische und logische Operationen implementiert worden.<sup>134</sup>

#### 5.4.1.3 Aggregationsregeln

Durch Aggregationsregeln werden die in einem Preismodell aufgeführten einfachen und komplexen Preise sowie Aggregationen darunter liegender Ebenen (zu Ebenen s. folgendes Kapitel) in einen Betrag verdichtet. Die Definition einer Aggregationsregel erfolgt analog zur Bestimmung eines komplexen Preises mit Hilfe von Berechnungsvorschriften. Im Gegensatz zu den Berechnungsvorschriften in komplexen Preisen gehen bei der Berechnung komplexe und einfache Preise anstelle von Dimensionen ein.

#### 5.4.2 Struktur eines Preismodells

Abbildung 5 fasst die Struktur eines Preismodells (ohne Dimensionen und Berechnungsvorschriften) zusammen. Ein Preismodell kann aus beliebig vielen Ebenen bestehen. Bis auf die oberste Ebene des Preismodells, Ebene 0, die nur

---

<sup>131</sup> ordinalskalierten

<sup>132</sup> nominalskalierten

<sup>133</sup> Bei diskreten Merkmalen könnte die Belegung auch durch eine arithmetische Regel erfolgen.

<sup>134</sup> In einer Folgeversion sollen in der Bepreisung häufig verwendete arithmetische und logische Ausdrücke in der Beschreibungssprache mittels MathML (W3C 2003) definiert und im Prototypen ausgewertet werden. Math-ML bietet ein umfassendes XML-Vokabular zur Definition mathematischer Ausdrücke und wird durch Softwarehersteller wie Design Science ([www.dessci.com](http://www.dessci.com)); MathType) Wolfram Research (Mathematica) und Mathsoft (MathCad) unterstützt.



eine Aggregationsregel enthalten darf, sind in jeder Ebene des Preismodells beliebig viele Aggregationsregeln (AR), einfache Preise (SP) und komplexe Preise (CP) zulässig. Eine Aggregationsregel verdichtet beliebig viele Aggregationsregeln, einfache Preise und komplexe Preise der nächst tieferen Modellebene. Preismodelle werden von unten nach oben durch Berechnung der komplexen Preise, einfachen Preise und Aggregationsregeln ausgewertet. Bei Ausführung des Preismodells aus Abbildung 5 werden in einem ersten Schritt die komplexen Preise  $CP\ 2,0$  und  $CP\ 2,1$  auf der untersten Ebene des Preismodells zur Bepreisung der Anzahl der durch einen Lernenden nachgefragten E-Books und PDF-Dateien berechnet. Diese werden in Aggregationsregel  $AR\ 1,0$  zu einem Rechnungsposten „Preis für Dokumente“ verdichtet. Verdichtungsregeln können frei definiert werden. So könnte neben einer einfachen Addition auch eine multiplikative Verdichtungsregel in der Aggregationsregel definiert werden. Der „Tarif für Kurs“ ( $AR\ 0,0$ ) ergibt sich z.B., indem zu den Preisen der nachgefragten Dokumente eine Teilnahmegebühr ( $SP\ 1,0$ ) und Preise für die nachgefragten Lektionen addiert werden.

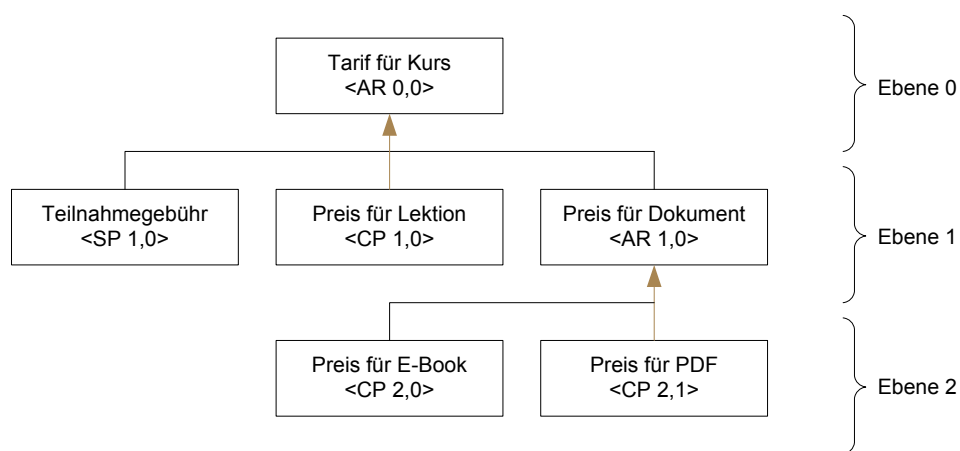


Abbildung 28: Stark vereinfachte Struktur eines statischen Preismodells

### 5.4.3 Definition der Orthosprache

Zur Darstellung der Orthosprache wird in Anlehnung an (Holten 2001) die *Objektypmethode* nach (Wedekind 1981) verwendet. Diese unterscheidet zwischen den Konstruktionsoperatoren Subsumtion, Subordination und Komposition zur Begriffsbildung.

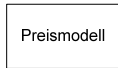
Gemäß (Wedekind 1981), S. 112 ist die *Subsumtion*<sup>135</sup> eine Form der Abstraktion bei der „[g]efragt wird, wie ein Objekt, das durch seinen Nominator vertreten wird, unter einen Begriff gebracht oder subsumiert werden kann.“ Es handelt sich also um die Begriffsschöpfung durch sprachliche Handlung bzw. Treffen von Aussagen über Objekte und somit die Zuordnung von Objekttypen zu Objekten (Holten 1999), S.11. Die Subsumtion wird in Entity-Relationship-Modellen durch die Bildung von Entity-Typen beschrieben.

Eine *Subordination* liegt dann vor, wenn Objekttypen (Prädikatoren) durch Zuweisung oder Absprechen von Eigenschaften in eine Ordnung gebracht werden (Holten 1999), S.11 / (Wedekind 1981), S.117-120. Die beiden Prädikatoren „Stetiges Merkmal“ und „Diskretes Merkmal“ sind z. B. dem Prädikator „Quantitatives Merkmal“ untergeordnet (Spezialisierung). In Entity-Relationship-Modellen kann die Subordinationsbeziehung durch IS-A-Beziehungstypen oder Subsets ausgedrückt werden.

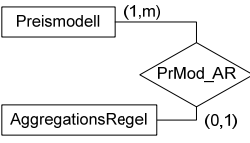

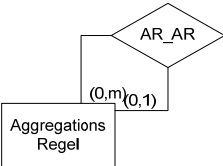
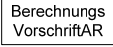
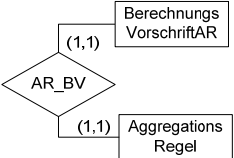
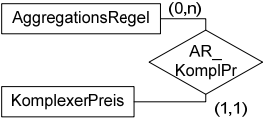
Durch *Komposition* werden „mehrere Begriffe durch sprachliche Handlungen (Aussagen) zueinander in Beziehung gesetzt“ (Holten 1999), S. 11. Beziehungstypen dienen in Entity-Relationship-Modellen der Verbindung von Entity-Typen und können zur Darstellung der Komposition verwendet werden.

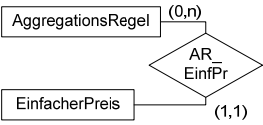

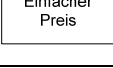
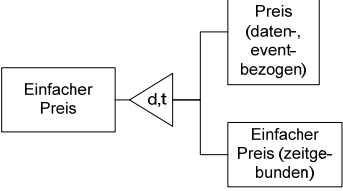
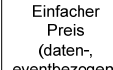
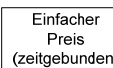
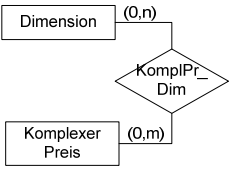

In der folgenden Tabelle wird die Beschreibungssprache definiert, indem Konzepte verbal beschrieben werden. Außerdem wird zu den Konzepten das entsprechende Konstrukt aus dem Metamodell aufgeführt.

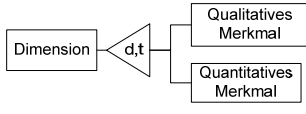
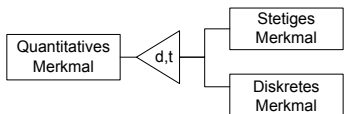
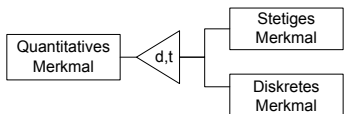
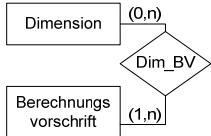
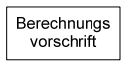
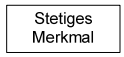

**Tabelle 11: Definition der Orthosprache und Verknüpfung mit dem konzeptuellen Metamodell**

Konzept	Bedeutung	Metamodellkonstrukt
Preismodell	Subsumtion: Hier als der alleinige Ursprungsknoten / die alleinige Wurzel eines Preismodells. Ein Objekt vom Typ <i>Preismodell</i> beinhaltet Metainformationen zu einem Preismodell, z. B. den Autor und die Gültigkeit. Darüber hinaus kann über den Knoten eine Zuordnung zu einem Produkt- und Kunden(gruppen)-modell erfolgen.	

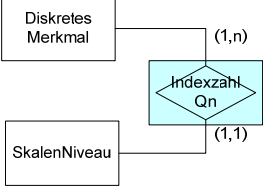

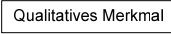
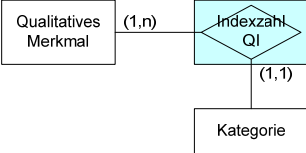
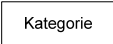
<sup>135</sup> Wedekind verwendet statt „Subsumtion“ den Begriff „Subsumption“.

Konzept	Bedeutung	Metamodellkonstrukt
PrMod_AR (Preismodell, Aggregationsregel)	Komposition: Beziehung zwischen <i>Preismodell</i> (-Wurzel) und <i>Aggregationsregel</i> . Eine Preismodell-Wurzel ist über genau eine Kante mit einer Aggregationsregel verbunden.	 <pre> classDiagram     class Preismodell     class AggregationsRegel     class PrMod_AR     Preismodell "1,m" -- "0,1" AggregationsRegel     PrMod_AR &lt;--&gt; Preismodell     PrMod_AR &lt;--&gt; AggregationsRegel </pre>
Aggregationsregel	Subsumtion: Aggregiert Mengen einfacher und/oder komplexer Preise zu einem Wert. Im Folgenden wird das Ergebnis einer Aggregationsregel auch als <i>Ebenen-Preis</i> bezeichnet.	 <pre> classDiagram     class AggregationsRegel </pre>
AR_AR (Aggregationsregel, Aggregationsregel)	Komposition: Rekursive Beziehung des Konzepts <i>Aggregationsregel</i> zu sich selbst. Aggregationsregeln können beliebig viele untergeordnete Aggregationsregeln umfassen. Eine Aggregationsregel hat dann keine Beziehung zu einer übergeordneten Aggregationsregel, wenn sie mit dem Wurzelknoten Preismodell verbunden ist. Eine Aggregationsregel hat keine untergeordnete Aggregationsregel, wenn sie eine Aggregationsregel der untersten Ebene ist.	 <pre> classDiagram     class AggregationsRegel     class AR_AR     AggregationsRegel "0,m" -- "0,1" AggregationsRegel     AR_AR &lt;--&gt; AggregationsRegel     AR_AR &lt;--&gt; AggregationsRegel </pre>
Berechnungs-Vorschrift-AR	Subsumtion: Eine <i>Berechnungsvorschrift</i> zur Verdichtung der in einer Aggregationsregel eingehenden einfachen Preise, komplexen Preise und Aggregationsregeln unterer Ebenen eines Preismodells.	 <pre> classDiagram     class BerechnungsVorschriftAR </pre>
AR_BVA (Aggregationsregel, Berechnungs-Vorschrift Aggregationsregel)	Komposition: Zur Berechnung eines Ebenen-Preises in einer Aggregationsregel muss genau eine Berechnungsvorschrift definiert werden. Dabei ist eine konkrete Vorschrift grundsätzlich genau einer Aggregationsregel zugeordnet.	 <pre> classDiagram     class BerechnungsVorschriftAR     class AggregationsRegel     class AR_BV     BerechnungsVorschriftAR "1,1" -- "1,1" AggregationsRegel     AR_BV &lt;--&gt; BerechnungsVorschriftAR     AR_BV &lt;--&gt; AggregationsRegel </pre>
AR_KomplPr (Aggregationsregel, Komplexer Preis)	Komposition: Bei der Berechnung eines Ebenen-Preises können in eine Aggregationsregel beliebig viele komplexe Preise eingehen.	 <pre> classDiagram     class AggregationsRegel     class KomplexerPreis     class AR_KomplPr     AggregationsRegel "0,n" -- "1,1" KomplexerPreis     AR_KomplPr &lt;--&gt; AggregationsRegel     AR_KomplPr &lt;--&gt; KomplexerPreis </pre>

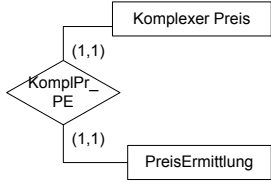

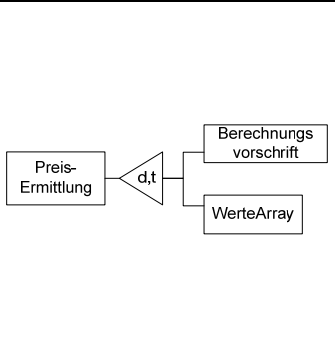

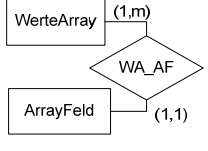
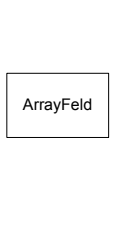
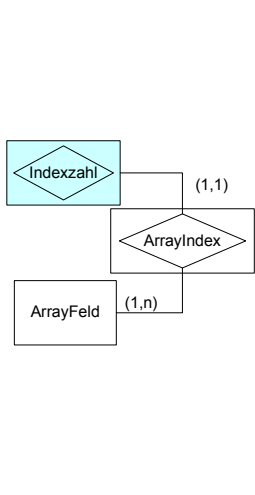
Konzept	Bedeutung	Metamodellkonstrukt
AR-EinfPr (Aggregationsregel, Einfacher Preis)	Komposition: Bei der Berechnung eines Ebenen-Preises können in eine Aggregationsregel beliebig viele einfache Preise eingehen.	
Komplexer Preis	Subsumtion: Ein Preis, der aus Merkmalsdimensionen berechnet oder abgelesen werden kann.	
Einfacher Preis	Subsumtion: Ein Preis, der nicht aus mehreren Dimensionen bestimmt wird.	
Einfacher Preis, Einfacher Preis datenbezogen, Einfacher Preis eventbezogen	Subsumtion und Subordination: Einfache Preise können entweder daten-/eventbezogen oder zeitgebunden sein.	
Einfacher Preis datenbezogen	Subsumtion: Ein <i>daten-/eventbezogener Einfacher Preis</i> wird immer dann verrechnet, wenn der Nutzer ein Event auslöst, z. B. das Benutzer-Event „Buchung eines E-Learning-Kurses“. Einfache Preise werden insbesondere dazu genutzt, verbrauchsunabhängige Gebühren wie Set-Up-Charges abzurechnen.	
Einfacher Preis zeitgebunden	Subsumtion: Ein <i>zeitgebundener Preis</i> wird berechnet, wenn ein Zeitfenster „geöffnet“ wird (z. B. „neues Semester hat begonnen“).	
KomplPr-Dim (Komplexer Preis, Dimension)	Komposition: Ein Komplexer Preis kann mehrere Merkmalsdimensionen umfassen. Eine Dimension kann zur Ermittlung unterschiedlicher komplexer Preise verwendet werden.	
Dimension	Subsumtion: Eine <i>statistische Merkmalsdimension</i> . Beispiele: „Anzahl der abgerufenen Lernobjekte in einem E-Learning-Kurs“, „Qualität eines Produktes“ oder „Anzahl der übertragenen Bytes während einer Internetverbindung.“	


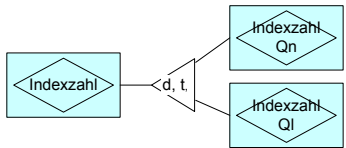
Konzept	Bedeutung	Metamodellkonstrukt
Dimension, Qualitatives Merkmal, Quantitatives Merkmal	Subsumtion und Subordination: Das Konzept Dimension wird vollständig (total, t) und disjunkt (d) durch die Konzepte Qualitatives Merkmal und Quantitatives Merkmal spezialisiert.	
Quantitatives Merkmal	Subsumtion: Ein <i>quantitatives Merkmal</i> kann bspw. ein ratioskaliertes Merkmal sein.	
Quantitatives Merkmal, Stetiges Merkmal, Diskretes Merkmal	Subsumtion und Subordination: Das Konzept Quantitatives Merkmal wird vollständig (total, t) und disjunkt (d) durch die Konzepte Stetiges Merkmal und Diskretes Merkmal spezialisiert.	
Dim_BV (Dimension, Berechnungsvorschrift)	Eine Berechnungsvorschrift zur algebraischen Ermittlung eines komplexen Preises bezieht sich auf mindestens eine Dimension. Dimensionen können in mehreren Berechnungsvorschriften verwendet werden.	
Berechnungsvorschrift	Subsumtion: Eine <i>Berechnungsvorschrift</i> ist ein algebraischer Ausdruck zur Berechnung eines Wertes aus den Merkmalsausprägungen der Dimensionen eines komplexen Preises.	
Stetiges Merkmal	Subsumtion: Ein <i>stetiges Merkmal</i> ist ein kardinal skaliertes Merkmal, das in einem Intervall jeden der unendlich vielen Werte annehmen kann. Beispiel: Zeitdauer bei unendlicher Messgenauigkeit	
Diskretes Merkmal	Subsumtion: Ein <i>diskretes Merkmal</i> ist ein kardinal skaliertes Merkmal <sup>136</sup> , das höchstens abzählbar viele Merkmalsausprägungen hat.	

<sup>136</sup> Eine Skala, die mit Hilfe der reellen Zahlen sowohl die Verschiedenartigkeit als auch die Rangfolge und mess- und zählbare Unterschiede (Abstand, Vielfaches) zum Ausdruck bringen kann, heißt Kardinalskala.

Konzept	Bedeutung	Metamodellkonstrukt
Indexzahl zu Diskretem Merkmal (Indexzahl Qn)	Komposition: Eine <i>Indexzahl zu einem diskreten Merkmal</i> ist eine Zahl, die ein konkretes Skalenniveau identifiziert, z. B. 500 für „500 Telefoneinheiten“. Ein diskretes Merkmal kann beliebig viele Indexzahlen umfassen.	
Skalenniveau	Subsumtion: Ein <i>Skalenniveau</i> beschreibt ein ablesbares Niveau eines diskreten Merkmals. Das Skalenniveau „500“ der Dimension „Megabytes“ bedeutet dann z. B. „Verbrauch = 500 MB“. <sup>137</sup>	
Qualitatives Merkmal	Subsumtion: <i>Qualitative Merkmale</i> sind ordinal- und nominalskalierte Merkmale. <i>Ordinalskalierte Merkmale</i> lassen lediglich Aussagen zur Verschiedenartigkeit und Rangfolge der Merkmalsausprägungen zu. <i>Nominalskalierte Merkmale</i> lassen nur Aussagen zur Verschiedenartigkeit zu.	
Indexzahl zu Qualitativem Merkmal	Komposition: Die <i>Indexzahl für ein qualitatives Merkmal</i> kann am Beispiel des Merkmals Kundengruppe beschrieben werden. Gibt es drei Kundengruppen (A, B, C), so ist die Indexzahl entweder Kundengruppe-A, Kundengruppe-B oder Kundengruppe-C. Eine Indexzahl umfasst den Namen des Merkmals, da sonst eine Zuordnung zu einem Merkmal später eventuell unmöglich wird.	
Kategorie	Subsumtion: <i>Merkmalskategorie</i> eines nominal oder ordinalskalierten qualitativen Merkmals. Beispiel: Merkmal Kundengruppe mit den Kategorien A, B und C(-Kunden).	

<sup>137</sup> Das Skalenniveau stetiger Merkmale kann für die Indizierung von Arrayfeldern nicht verwendet werden. Der Begriff „Skalenniveau“ ist daher nur für Diskrete Merkmale definiert.

Konzept	Bedeutung	Metamodellkonstrukt
KomplPr-PE (Komplexer Preis, Preisermittlung)	Komposition: Zu jedem komplexen Preis gibt es eine <i>Methode der Preisermittlung</i> .	
PreisErmittlung	Subsumtion: <i>PreisErmittlung</i> beschreibt die Ermittlung eines Preises in einem komplexen Preis.	
Preisermittlung, Wertearray, Berechnungsvorschrift	Subsumtion und Subordination: Die Ermittlung eines Preises kann entweder durch eine <i>Berechnungsvorschrift</i> oder durch ein <i>Wertearray</i> erfolgen. Das Konzept „Preisermittlung“ wird vollständig (total, t) und disjunkt (d) durch die Konzepte „Werte-Array“ und „Berechnungsvorschrift“ spezialisiert.	
Wertearray	Subsumtion: Ein n-dimensionales <i>Array mit Feldwerten</i> , das durch qualitative und diskrete Merkmalsdimensionen aufgespannt wird.	
WA_AF (Werte-Array, Array-Feld)	Komposition: Ein Werte-Array umfasst mehrere Feldwerte. Ein Feldwert gehört immer nur zu einem Wertearray.	
Array-Feld	Ein <i>Feldwert zu einem Array</i> (z. B. einer Tabelle), der einen Preis kapselt. Gebraucht im Sinne einer n-dimensionalen statistischen Variable (Vgl. (Bohley 1989), S. 65).	
ArrayIndex	Komposition und Subsumtion: Ein <i>ArrayIndex</i> besteht aus mehreren Indizes und spannt dadurch einen n-dimensionalen Raum auf. Ein konkretes Arrayfeld kann durch mehrere Indizes beschrieben werden. Ein Array-Index indiziert die Felder eines Arrays. Beispiel: $a_{s_1, s_2, \dots, s_n} \rightarrow$ Das Arrayfeld (und der zugehörige Feldwert) $a$ wird durch die Indizes $s_1, s_2, \dots, s_n$ der diskreten oder qualitativen Merkmale	

Konzept	Bedeutung	Metamodellkonstrukt
	$1, 2, \dots, n$ indiziert. $\{s_1, s_2, \dots, s_n\} \rightarrow a$ .	
Indexzahl	Subsumtion und Komposition: Eine <i>Indexzahl</i> ergibt sich generell aus einem Namen und einer Messgröße. Z. B. „Kundengruppe“ mit der „Messgröße“ „A“ Bedeutet „Kundengruppe-A“	
Indexzahl, Indexzahl Qn, Indexzahl Ql	Subsumtion und Subordination: Indexzahlen können grundsätzlich für qualitative und quantitative Merkmale aufgestellt werden. Stetige Merkmale können für Indexzahlen erst nach Einteilung in des stetigen Merkmals in Bereiche verwendet werden (nicht im Metamodell dargestellt.)	

#### 5.4.4 Definition des fachkonzeptuellen Metamodells

Das *konzeptuelle Metamodell* der Beschreibungssprache für Preismodelle *PriMoL* (Price-Modeling-Language) ist im Entity-Relationship-Diagramm<sup>138</sup> in Abbildung 29 auf Seite 123 definiert. Neben der Darstellung der Begriffe, die in der Beschreibungssprache für Preismodelle Verwendung finden, müssen zur Umsetzung weitere Modelle erstellt werden. Diese sind im Entity-Relationship-Diagramm in Abbildung 29 mit gestrichelten Linien umrandet und farblich hervorgehoben. Es handelt sich dabei um Verweise auf die in Abbildung 25 beschriebenen Teilmodelle, d.h. auf Produkt- und Kundengruppenmodelle. Zur Instanziierung von Abrechnungsmechanismen ist in Abrechnungssystemen insbesondere auch die Darstellung von Zeiten wichtig. Ein Preismodell wird in regelmäßigen zeitlichen Abständen zur Berechnung der Preise für einen bestimmten Tarif ausgewertet. Eine Auswertung kann alternativ durch Anstoßen eines Abrechnungsalgorithmus durch einen Anwender ausgelöst werden. Auch im Fall einfacher, zeitgebundener Preise müssen in regelmäßigen zeitlichen Abständen Zurechnungen durchgeführt werden. Das konzeptuelle, sprachbasierte Metamodell, das lediglich die Begriffe der Sprache ordnet und zusammen mit der

<sup>138</sup> Entity-Relationship-Diagramm = grafisches Modell in Entity-Relationship-Notation. (Chen 1977)



Orthosprache deren Bedeutung sowie deren Beziehungen festlegt, kann nicht direkt zur Definition eines Preismodells verwendet werden. Das konzeptuelle Modell muss in eine Notation überführt werden, die die Darstellung konkreter Instanzen von Preismodellen zulässt.<sup>139</sup> Neben einer grafischen Notation, die Anwendern zur visuellen Gestaltung von Preismodellen dient, können Preismodelle durch Relationen im Datenbank-Sinn dargestellt und letztlich durch Überführung in eine Datenbank persistent gemacht werden. Bei der Definition einer Notation für ein Metamodell ist darauf zu achten, dass insbesondere die Ausdrucksmächtigkeit der Sprache durch die Definition der Notation nicht vermindert wird. Die Einhaltung der semantischen Bedingungen aus dem Metamodell kann z. B. durch Definition von Integritätsbedingungen, bspw. über Constraint-Languages oder durch eine Software, die die Modellierungssprache implementiert, gewährleistet werden.

#### **5.4.5 Definition einer grafischen Notation**

Die Definition einer grafischen Repräsentation / Notation bietet für einen Designer, der mit der Beschreibungssprache Preismodelle konstruieren möchte, Vorteile. Grafische Modelle abstrahieren von der internen, formalen Darstellung eines Preismodells und machen die Struktur von Preismodellen „greifbar“. *Grafische Artefakte werden für Aggregationsregeln, einfache Preise und komplexe Preise eingeführt.* Durch grafische Modelle wird die Kommunikation über Preise und das Verständnis der Preise durch die Vorgabe eines festen grafischen Symbolvorrats verbessert. Grafische Modelle können als Ausgangspunkt für Neuentwicklungen dienen. Auch die Bildung von Varianten eines Preismodells kann durch Anpassung eines grafischen Modells visualisiert werden, z. B. mittels Verfeinerung durch Hinzufügen von Elementen.

---

<sup>139</sup> Wünschenswert wäre auch, dass die Repräsentation eines Preismodells direkt oder mit möglichst wenig Aufwand gespeichert werden kann. Bei der Entwicklung von Datenbanken findet man eine ähnliche Vorgehensweise. Ein Metamodell, das als ER-Diagramm vorliegt, kann durch Überführung in ein logisches Datenmodell in Relationenschreibweise (Repräsentation) als Basis für die Entwicklung eines Datenbankschemas dienen. Das Datenbankschema kann dann in einem zweiten Schritt zum Entwurf einer konkreten Datenbank genutzt werden, die die Speicherung eines bzw. mehrerer Preismodell-Instanzen in den durch das Schema definierten Relationen ermöglicht.



In Tabelle 12 auf S. 125 werden neben der grafischen Repräsentation die entsprechenden konzeptuellen Elemente aus dem Metamodell aufgeführt, das auf Seite 123 abgebildet ist und dessen Elemente in Abschnitt 5.4.3 beschrieben wurden. Die grafische Repräsentation der einfachen Preise, komplexen Preise und Aggregationsregeln ist zweigeteilt. So sind neben der Kennzeichnung des Typs durch ein Icon und durch einen Modifier (z. B. SP für den Typ „Einfacher Preis“) auch zugehörige Eigenschaften des jeweiligen Elements in einem Eigenschaftsfenster sichtbar. Grafische Elemente werden durch (gerichtete) Kanten miteinander verbunden, die die Bestandteile eines Preismodells hierarchisch anordnen und die Auswertungsreihenfolge festlegen. Die grafische Notation wird durch den erstellten Prototyp, der in Kapitel 6 beschrieben wird, in Teilen unterstützt.

#### **5.4.6 Definition einer relationalen Notation**

Die relationale Notation der Beschreibungssprache bildet die Basis für die Speicherung von Preismodellen im Prototyp. Sie implementiert einen großen Teil der im konzeptuellen Metamodell vorgegebenen Begriffe und ist damit als logisches Schema im Feld Metaebene / DV-Konzeption in der Abbildung 24 auf Seite 96 anzusiedeln.<sup>140</sup> Neben Relationenschemata für einfache Preise und komplexe Preise werden auch Schemata für Aggregationsregeln und für den Wurzelknoten 'Preismodell' definiert.

---

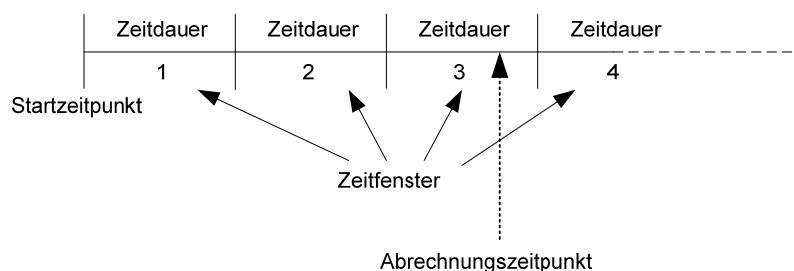
<sup>140</sup> Es werden bisher keine ordinal- und nominalskalierten Merkmalsdimensionen unterstützt.

<p>&lt;Name&gt; : CP</p> <p>ID: _____ Beschreibung: _____</p> <p>Merkmalsdimensionen: _____ (D1, D2, D3, ..., Dn)</p> <p>Preisermittlung: _____ (berechnet / Wertearray)</p> <p>Metered: _____ (Ja / Nein)</p>		<p>Komplexer Preis</p>
<p>&lt;Name&gt; : SP</p> <p>ID: _____ Wert: _____ Bezugsperiode: _____ Zeiteinheit: _____ Anzahl: _____</p>		<p>Einfacher Preis</p>
<p>&lt;Name&gt;: AR</p> <p>ID: _____ Beschreibung: _____</p> <p>Untergeordnete ARs: _____ (AR1, AR2, AR3, ..., ARn)</p> <p>Einfache Preise: _____ (SP1, SP2, SP3, ..., SPn)</p> <p>Komplexe Preise: _____ (CP1, CP2, CP3, ..., CPn)</p>		<p>Aggregationsregel</p>
<p style="text-align: center;">↑</p>		<p>Hierarchisierungsbeziehung</p>

**Tabelle 12: Grafische Artefakte für Einfacher Preis, Komplexer Preis und Aggregationsregel**

### 5.4.6.1 Einfacher Preis

Abbildung 31 beschreibt die Relationenschemata für einfache Preise. Einfache Preise haben einen Namen und einen Wert, der verbucht wird. Bei zeitgebundenen Preisen wird der angegebene Wert jeweils nach Ablauf eines Zeitfensters verbucht. In der in Abbildung 30 dargestellten Situation wären Zeitfenster 1 und 2 zum Abrechnungszeitpunkt abgelaufen. Daher würde der Preiszähler / Wert für den einfachen Preis zweimal verbucht werden. Zeitfenster 3 ist noch nicht abgelaufen und kann daher noch nicht abgerechnet werden. Der Startzeitpunkt in Abbildung 30 entspricht dem Attribut `ErstesAbrgDatum` in Abbildung 31. In der Regel wird ein Abrechnungsbetrag für einen einfachen Preis durch Summenbildung ermittelt (z. B. € 10,00 + € 10,00 = € 20,00 für die abgelaufenen Zeitfenster). Für komplexere Berechnungen können Ausdrücke mit der Minisprache *AccLan* (Accounting-Language) gebildet werden. Die Sprachdefinition ist im Anhang dargestellt. *AccLan*-Ausdrücke werden im Codeblock in der Relation `MediationExpression` gespeichert.



**Abbildung 30: Verbuchung eines einfachen Preises**

Bei event-/datenbasierten Preisen wird ein Wert für jedes Event verbucht. Events sind durch einen SQL-Ausdruck (Attribut `SQLAusdruck` in Abbildung 31) zu spezifizieren. Dabei gibt jedes Tupel der Ergebnismenge des SQL-Ausdrucks ein Event wieder. Ein Beispiel-SQL-Ausdruck zur Überprüfung eines „Download-Events“ könnte wie folgt formuliert werden:

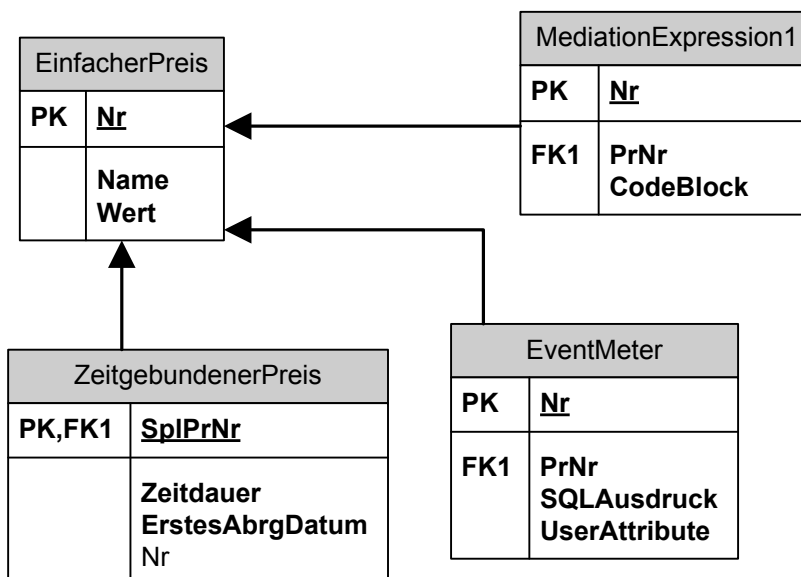
```
SELECT User-ID, Download-ID FROM Downloads;
```

Tabelle 13 gibt das Ergebnis des SQL-Ausdrucks wieder. Für den Kunden mit der `User-ID` 1 wurden drei Events registriert, die mit einem einfachen Preis bewertet werden sollen. Damit die Events einem Kunden zugeordnet werden können, ist es bei der Speicherung wichtig, den Bezeichner des SQL-Attributs, das die `User-ID` enthält zu speichern. In der Relation `EventManager` ist dafür ein

Attribut namens `UserAttribute` definiert. Im Beispiel würde `UserAttribute` den Wert 'User-ID' enthalten, also den Namen des Attributs, das den Kunden identifiziert. Analog zu einem zeitgebundenen Preis kann eine `MediationExpression` in einem Codeblock definiert werden.<sup>141</sup>

User-ID	Download-ID
1	123ggh
1	345zzf
1	Ok1456
2	121212
2	131313
2	1234gghz
...	...

*Tabelle 13: Events für einfachen Preis*



*Abbildung 31: Einfacher Preis als Relationenschema*

#### 5.4.6.2 Komplexer Preis

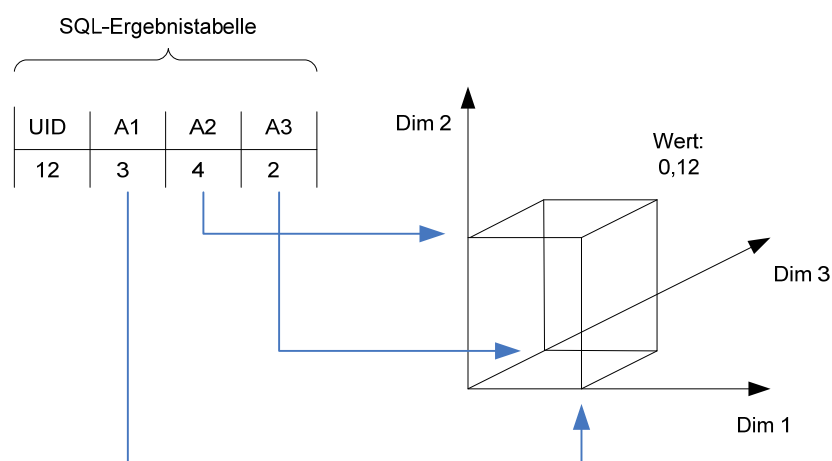
Komplexe Preise ermöglichen die Bepreisung anhand von Dimensionen. Dabei wird im Rahmen des Prototyps, dessen Entwicklung in Kapitel 6 beschrieben wird, vorausgesetzt, dass Merkmalsdimensionen in diskreter Form abgebildet

<sup>141</sup> Wird keine Verarbeitungsregel definiert, so wird die Summe berechnet.

werden. Für die Dimensionen Kundengruppe, Eigene Position und Komplexität zur Berechnung des Preises für ein Lernobjekt (vgl. Abbildung 27 auf S. 111) bedeutet dies, dass den Skalenniveaus numerische Werte zugeordnet werden müssen, z. B. 1 = Europa, 2 = USA, 3 = Rest usw. Ein Beispiel-SQL-Ausdruck zur Überprüfung der Nachfrage nach Lernobjekten könnte wie folgt formuliert werden:

```
SELECT User-ID AS UID, Complexity AS A1, Cust-Group AS
A2, Position AS A3 FROM LO-Consumption;
```

Wie bei eventbasierten, einfachen Preisen kann die Zurechnung von Preisen zu Kunden nur erfolgen, wenn das den Kunden identifizierende Attribut in UserAttribut (Relation Meter) festgelegt wird. Als Ergebnis der Auswertung eines komplexen Preises wird zu jedem Kunden eine Menge von einzelnen Preisen gesammelt. Mit Hilfe von AccLan können wiederum beliebig komplexe Verdichtungsvorschriften für die einzelnen Werte definiert werden. Die in Abbildung 33 aufgelisteten Relationen Dimension, Skalenniveau, Index und Feld dienen der Beschreibung des n-dimensionalen Raumes, in dem die Einzelpreise festgelegt werden. Im Meter wird der SQL-Ausdruck angegeben, der die Tupel für die Bestimmung der Einzelpreise liefert. Ein Tupel enthält in den Attributen die Skalenniveaus, mit denen eine Wertebestimmung erfolgt (vgl. Abbildung 32).



**Abbildung 32: Wertermittlung im komplexen Preis**

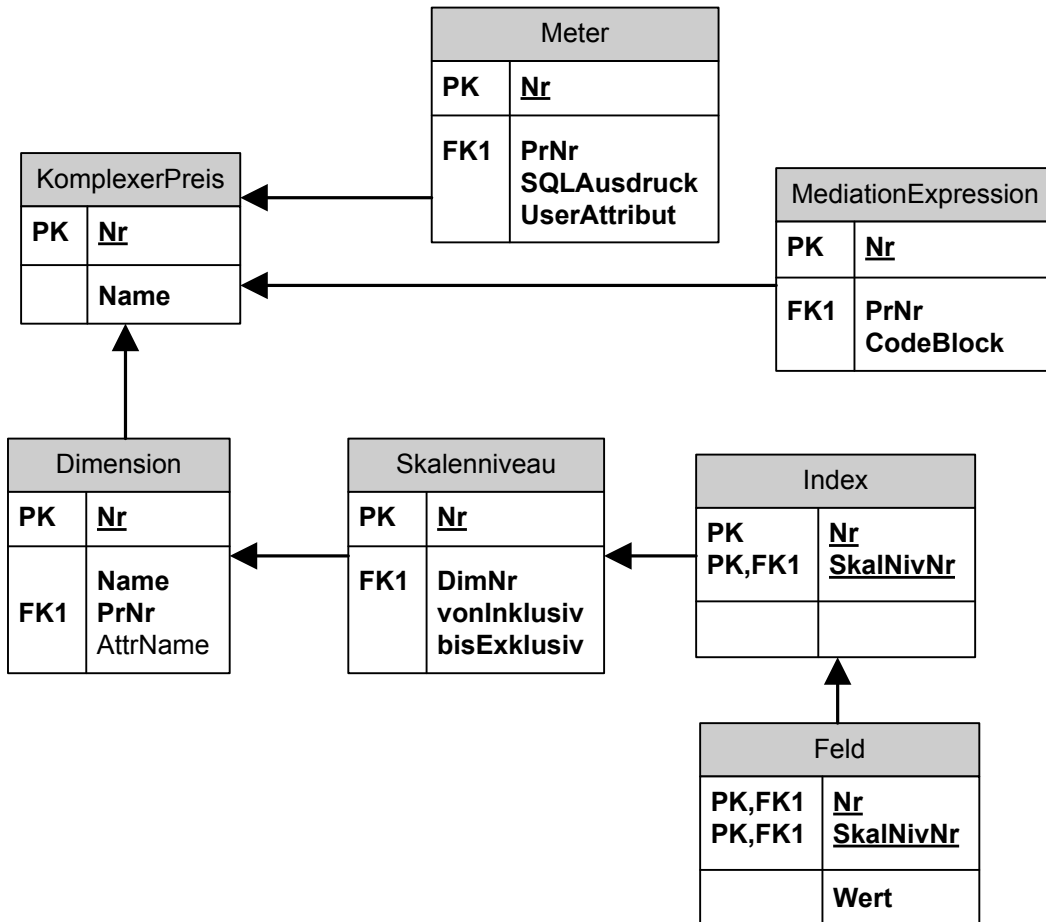


Abbildung 33: Komplexer Preis als Relationenschema

### 5.4.6.3 Aggregationsregel

Aggregationsregeln verdichten die eingehenden komplexen und einfachen Preise. Berechnungsvorschriften zur Verdichtung werden analog zu einem komplexen Preis in einem Codeblock definiert, der einen AccLan-Ausdruck enthält. Eine Aggregationsregel liefert als Ergebnis einen skalaren Wert zurück. Dieser kann als Preiszähler interpretiert werden. Im Gegensatz zu komplexen Preisen können die AccLan-Konstanten MIN, MAX, SUM und COUNT nicht verwendet werden, da diese auf Mengen definiert sind und in eine Aggregationsregel nur Skalarwerte eingehen können. Die Relationenschemata zu Aggregationsregeln sind in Abbildung 34 dargestellt. Die Relationschemata AR\_KomplPr und AR\_EinfPr dienen der Referenzierung eingehender einfacher und komplexer Preise.



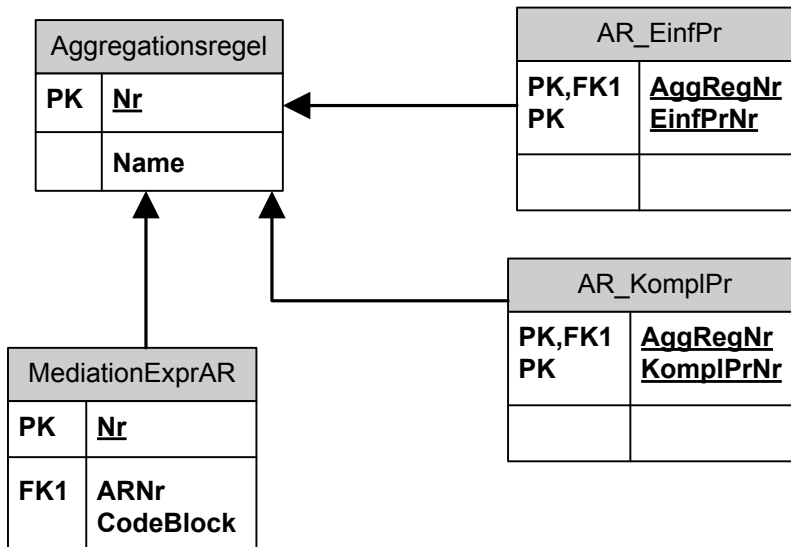


Abbildung 34: Aggregationsregel als Relationenschema

#### 5.4.6.4 Auswertung von Preismodellen

Preismodelle werden, wie in Kapitel 5.4.2 beschrieben, von unten nach oben ausgewertet. Dabei kann ein einziges Preismodell für beliebig viele Kunden verwendet werden. Damit eine Auswertung für einen einzelnen Kunden möglich wird, muss in komplexen und einfachen, eventbasierten Preisen angegeben werden, durch welche Attribute in der Produktivdatenbank, d.h. der Datenbank des Service-Provisioning-Systems, Kunden identifiziert werden. Meter für komplexe und einfache Preise steuern die „Messung“ der auszuwertenden Daten. Ein Meter wird in der aktuellen Version der Beschreibungssprache durch einen SQL-Ausdruck definiert.<sup>142</sup> Der Ausdruck liefert im Falle eines komplexen Preises Datensätze / Tupel mit Attributwerten zurück, die die Skalenniveaus der Dimensionen im komplexen Preis setzen und somit individuelle Werte zurückgeben. Als wesentliche Eigenschaft von Preismodellen gilt, dass jede Komponente nur einen einzelnen skalaren Wert zurückgibt. Um die Berechnung eines Skalarwertes zu erzwingen, sollten in komplexen Preisen Ausdrücke in *AccLan* definiert werden. Wird kein Ausdruck angegeben, so erfolgt eine Summierung der Einzelwerte. In *AccLan*-Ausdrücken können z. B. durch Angabe einer Treppenfunktion Staffeltarife definiert werden. Die Sprachdefinition zur Accounting-Language *AccLan* ist im Anhang „A.2.1 Accounting-Language-Definition in EBNF“, S. 202 abgedruckt.

<sup>142</sup> Eine grafische Definition des Meters ist zurzeit nicht möglich.

## **5.5 Einsatzmöglichkeiten und Bewertung von PriMoL**

Neben der Verwendung von PriMoL für das Pricing im E-Learning<sup>143</sup>, wurden in den Kapitel 5.3.2 und 5.3.3 aktuelle Tarife in der Versicherungsbranche und im Mobilfunk beschrieben. Diese lassen sich mit Hilfe von PriMoL beschreiben. Konzeptuelle Hinweise zu einer Umsetzung der Tarife mittels PriMoL finden sich in den folgenden Kapiteln 5.5.1 und 5.5.2. Daran anschließend wird PriMoL mit Hilfe unterschiedlicher Kriterien nach Zelewski, Kapitel 5.5.3 und Schütte, Kapitel 5.5.4 analysiert.

### **5.5.1 Abbildung von Kfz-Haftpflichttarifen mit PriMoL**

Tabelle 14 setzt die Begrifflichkeit der Tarifierung von Versicherungen in Bezug zu den in der Modellierungssprache PriMoL verwendeten Begriffen. Versicherungstarife sind im Falle von multiplikativen Tarifen einfach strukturiert. Die Darstellung der geschilderten Tarife ist mit PriMoL ohne Einschränkungen möglich. Die Ermittlung von Tariffaktoren und die Ableitung von Tarifmerkmalen aus Risikomerkmale werden durch PriMoL nicht unterstützt. Diese erfordern weitere Software<sup>144</sup> zur Durchführung multivariater Analysen mit Massendaten.

### **5.5.2 Abbildung von Mobilfunktarifen mit PriMoL**

Die Einzeltarife für Events, Sprach- und Datenübertragung lassen sich durch Konzepte der Beschreibungssprache PriMoL abbilden. Dabei stellt die Abbildung der Bepreisung in der Sprachkommunikation eine besondere Herausforderung dar. Regeln für eine mögliche Abbildung eines Tarifs sind in der folgenden Tabelle aufgelistet. Es bietet sich an, die Einzeldienste im Preismodell durch Einführung von Aggregationsregeln zu bezeichnen. So können bspw. Event-Dienste wie SMS und MMS in einer Aggregationsregel „Event-Dienste“ zusammengefasst werden.

---

<sup>143</sup> Ein detailliertes Vorgehen zur Umsetzung von Preismodellen in einem integrierten Lern-Managementssystem ist in Kapitel 7 zu finden.

<sup>144</sup> Die Durchführung multivariater Analysen wird z. B. durch die folgenden Software-Produkte unterstützt: SAS, SPSS, RATS oder Watson Wyatt Pretium.

Begriff in der Tarifierung	Begriff in PriMoL
Tarif	Preismodell
Merkmal	Komplexer Preis: ein komplexer Preis kann verwendet werden, um ein Tarifmerkmal wie „Jahresfahrleistung“ zu modellieren. Dabei wird das Merkmal durch eine Dimension des komplexen Preises abgebildet. Alternativ kann ein einfacher, eventbasierter Preis verwendet werden.
Merkmalsausprägung und Tariffaktor	Skalenniveau und Feldwert zu einem Skalenniveau
Tariffunktion	Aggregationsregel(n), die die mit komplexen Preisen ermittelten Werte verdichtet/verdichten.
Zuordnung zu einem Kunden	Mittels Angabe eines „User-Attribute“ in einem SQL-String/Meter.

**Tabelle 14: Überführung von Tarifen in Preismodelle**

Teil eines Mobilfunktarifs	Darstellung in PriMoL
<i>Events</i>	Eventgebundener, einfacher Preis. Falls der Preis eines Events von mehreren Dimensionen abhängt, so sollte stattdessen ein „komplexer Preis“ verwendet werden
<i>Sprachdienste</i>	
Grundgebühren	Zeitbasierter, einfacher Preis.
Minutenpreis	Komplexer Preis, bei dem in den Dimensionen die Parameter zur Berechnung des Minutenpreises festgelegt werden. Freiminuten können in einem AccLan-Ausdruck definiert werden.
Flat-Fee	One-Time-Charge
<i>Datenübertragung</i>	Analog zu Sprachdiensten
<i>Sonstiges</i>	
Tariffunktion	Aggregationsregel, die die Tarifkomponenten verdichtet.
Zuordnung zu einem Kunden	Mittels Angabe eines „User-Attribute“ in einem SQL-String/Meter.

**Tabelle 15: Abbildung eines Mobilfunktarifs in Preismodellen**

### 5.5.3 Bewertung anhand der Kriterien nach (Zelewski 1996)

Obwohl die in Kapitel 4.2 genannten Kriterien von (Zelewski 1996) lediglich zur Beurteilung von Petri-Netzen aufgestellt wurden, sind diese auch für die Beurteilung der Beschreibungssprache PriMoL geeignet. Im Folgenden soll insbesondere die Modellierungsgüte von PriMoL beurteilt werden.<sup>145</sup> Die spezielle Modellierungsfähigkeit von PriMoL, die in der Abbildung von Tarifsystemen liegt, wird durch die Kapitel 7, 5.3.2 und 5.3.3 implizit beschrieben, indem Anwendungsfälle für den Einsatz der Beschreibungssprache skizziert werden.

Die „*pragmatisch ausgerichteten*“ Determinanten der *Modellierungsgüte* beleuchten die anwendungsnahen Qualitäten eines Modellierungskonzepts. Dabei wird beurteilt, ob sich mit der Hilfe des Konzepts reale Probleme ‚praxisgerecht‘ modellieren lassen.“ (Zelewski 1996), S. 371. Die pragmatische Modellierungsgüte eines Modells lässt sich anhand der in Abbildung 19 genannten Unterkriterien analysieren.

„Aus der Perspektive der Modellierungsphasen wird grob zwischen der Konstruktion [...], der Analyse [...] und der Anpassung (Wartung) von Modellen unterschieden. Dieser Differenzierung entsprechen die Beurteilungskriterien der *Konstruktivität*, *Analysierbarkeit* und *Adaptivität*.“ (Zelewski 1996), S. 371. Daneben bilden die *Anwendungs- /Benutzerfreundlichkeit* sowie die *Implementierbarkeit* wichtige Kriterien zur Beurteilung einer Modellierungssprache. Die zuvor genannten Kriterien werden in der folgenden Tabelle bewertet.

Kriterium	Bewertung
Konstruktivität	PriMoL und der in Kapitel 6.5 beschriebene Price-Modeler-Prototyp unterstützen die <i>modulare Gestaltung von Tarifsystemen</i> , indem <i>grafische Artefakte</i> durch Kombination zu einem Preismodell zusammengesetzt werden. Teile von Preismodellen können darüber hinaus im Price-Modeler als „Pattern“ gespeichert und in anderen Preismodellen wieder

<sup>145</sup> „Für die *allgemeine* Modellierungsfähigkeit oder Universalität eines Modellierungskonzepts hat sich [...] die Ausdrucksmächtigkeit von TURING-Automaten als gemeinsamer Vergleichsstandard durchgesetzt.“ (Zelewski 1996), S. 370. Die Allgemeine Modellierungsfähigkeit wird nicht untersucht, da die Untersuchung den „wirtschaftsinformatischen Kontext“ und Rahmen der Arbeit sprengen würde.

Kriterium	Bewertung
	<p>verwendet werden. Auf diese Weise kann „das Gesamtobjekt [Preismodell] zunächst in einfachere, jeweils in sich abgeschlossene Teilobjekte zerlegt werden.“ (S. 371).</p> <p>Durch <i>Einführung neuer Sprachkonstrukte</i> durch den Anwender ist PriMoL einer Erweiterung zugänglich.</p> <p>Eine <i>Top-Down-Entwicklung</i> im Sinne einer Verfeinerung von Modellelementen, analog zum Ersetzen von Transitionen durch transitionsberandete Teilnetze in Petri-Netzen, wird in PriMoL nicht unterstützt.</p>
Analysierbarkeit	<p>Analysetechniken zur Untersuchung von Preismodellen sind im Price-Modeler-Prototyp kodiert. So wird unmittelbar anhand des in Kapitel 5.4.3 beschriebenen <i>Metamodells</i><sup>146</sup> der Sprache PriMoL überprüft, <i>ob ein Preismodell vollständig spezifiziert</i> und somit ausführbar ist (ansonsten wird eine Warnung ausgegeben). Auch das „Durchspielen“ von Preismodellen mit Vergangenheitsdaten von Kunden ist möglich. Somit kann bspw. auch der Wechsel zwischen zwei Preismodellen ceteris paribus simuliert und Auswirkungen auf die Erlöse eines Service Providers untersucht werden.</p>
Adaptivität	<p>Unter Adaptivität „wird die Möglichkeit verstanden, ein Modell an unterschiedliche Modellierungsbedürfnisse anzupassen.“ (S. 372) Preismodelle können im Nachhinein nur eingeschränkt <i>verfeinert</i> werden, z. B. durch Hinzufügen von Dimensionen in komplexen Preisen. <i>Horizontale Modellerweiterungen</i> durch Einbettung neuer Elemente und <i>Modellverkleinerungen</i> durch Restriktion werden hingegen unterstützt.</p>

<sup>146</sup> Die Überprüfung findet streng genommen mit einem abgeleiteten XML-Schema ab, gegen das die Preismodelle durch Parsing validiert werden. Das XML-Schema ist im Anhang „A.2.2 XML-Schema von PriMoL“ ab Seite 203 zu finden.

Kriterium	Bewertung
Anwendungs-effizienz und Benutzerfreundlichkeit	Da im Gegensatz zu textuellen und programmatischen Verfahren in marktgängigen Softwareprodukten <sup>147</sup> die Modelle in PriMoL weitgehend durch <i>grafische Konstruktion</i> erstellt werden, ist davon auszugehen, dass die Anwendungseffizienz und Benutzerfreundlichkeit der Modellierungssprache höher als bei den vorgenannten Verfahren ist.
Implementierbarkeit	Preismodelle, die in PriMoL beschrieben werden, können unmittelbar in einer <i>Laufzeitumgebung</i> umgesetzt werden. Die Implementierbarkeit wird in Kapitel 7 am Beispiel eines E-Learning-Systems beschrieben. Durch Austausch von Preismodellen lassen sich Abrechnungsmodalitäten in Service-Provisioning-Systemen „on-the-fly“ ändern.

*Tabelle 16: Kriterien „Konstruktivität“, „Analysierbarkeit“, „Adaptivität“ und „Implementierbarkeit“ als anwendungsnahe Qualitäten eines Modells*

#### 5.5.4 Bewertung anhand der GoM nach (Schütte 1998)

In diesem Kapitel wird PriMoL anhand der in Kapitel 4.2 aufgestellten Kriterien bewertet.

Kriterium	Bewertung
Konstruktionsadäquanz	Unter der Konstruktionsadäquanz ist der <i>Nutzen eines Modells für seine praktische Anwendung</i> zu verstehen. Da PriMoL eigens für die <i>Abbildung von Tarifsystemen</i> entworfen wurde und die Umsetzung von Modellen in prototypisch erfolgte, ist die Konstruktionsadäquanz als hoch anzusehen.
Sprachadäquanz	Unter der Sprachadäquanz werden u. a. die semantische Mächtigkeit der Sprache und ihr Formalisierungsgrad betrachtet. Die <i>semantische Mächtigkeit</i> von PriMoL ist auf den konkreten Anwendungsfall des Pricings beschränkt. Mit PriMoL können beliebig komplexe Preisberechnungsmecha-

<sup>147</sup> Eine Auflistung von Softwareprodukten befindet sich im Anhang „A.5: Industrielle Abrechnungssysteme“ ab Seite 229.

Kriterium	Bewertung
	<p>nismen abgebildet werden. Darüber hinaus ist eine Erweiterung um Sprachkonstrukte durch Anpassung der Orthosprache und des Metamodells möglich.</p> <p>PriMoL unterstützt unterschiedliche <i>Formalisierungsgrade</i>. Durch die reine grafische Zusammenstellung von Modellkonstrukten können Preismodelle visualisiert werden (geringer Formalisierungsgrad). Diese sind jedoch erst umsetzbar, wenn arithmetische Operationen für die Preisberechnung formal – mittels AccLan – definiert wurden (hoher Formalisierungsgrad).</p>
Sprachrichtigkeit	Siehe unter „Analysierbarkeit“ in Tabelle 16.
Wirtschaftlichkeit	Zur Überprüfung der Wirtschaftlichkeit der entwickelten Methode, müsste diese in einer vermarktungsfähigen Version vorliegen. Da bisher eine prototypische Version vorliegt, ist eine <i>ökonomische Bewertung</i> (Amortisationszeit usw.) <i>nicht durchführbar</i> .
Klarheit	Die Verständlichkeit der Sprache wird durch die Erklärung ihrer Semantik gefördert. Bei der Entwicklung wurde ein besonderes Augenmerk darauf gelegt, möglichst wenige grafische Artefakte einzuführen und somit eine schnelle Einarbeitung eines Anwenders zu ermöglichen. Die eindeutige Definition der Sprachkonstrukte erfolgt im Rahmen der Definition der Orthosprache.
Systematischer Aufbau	Die von Schütte genannten Kriterien „Inter-Modellkonsistenz zwischen Struktur- und Verhaltensmodellen“, „Unterstützung von Sichten“ und „semantische Deckungsgleichheit zweier Modelle“ konnten nicht überprüft werden.

*Tabelle 17: Kriterien „Konstruktionsadäquanz“, „Sprachadäquanz“, „Sprachrichtigkeit“, „Wirtschaftlichkeit“, „Klarheit“ und „Systematischer Aufbau“ zur Bewertung von Modellierungssprachen*

## 5.6 Schlussbetrachtung und Kritik

PriMoL und die integrierte Sprache AccLan bieten ein mächtiges Instrumentarium zur Formulierung von Preismodellen. Der Definition von Preismodellen in PriMoL sollte grundsätzlich eine Analyse des Marktes (Zahlungsbereitschaften, Preiselastizitäten, regelmäßig verwendete Parameter in Preismodellen etc.) vorausgehen, um die Anwendbarkeit der zu erstellenden Preismodelle zu gewährleisten.

Gerade weil mit PriMoL beliebige Rechenregeln auf relational gespeicherte Daten definiert und im Price-Modeler umgesetzt werden können, können die in der Sprache und im Modeler definierten Integritätsbedingungen dem Anwender eine Überprüfung bspw. der Sinnhaftigkeit formulierter Preismodelle niemals vollständig abnehmen. In den folgenden Beispielen werden Probleme beschrieben, die bei der Modellierung entstehen können:

- *Widersprüche im Preismodell durch Mehrfachverwendung von Verbrauchsdaten:* Anwender formulieren Preismodelle, die mehrfach auf den gleichen Daten Auswertungen durchführen. Die Mehrfachverwendung kann sinnvoll sein, wenn bspw. aufgrund der Daten eine Zuordnung des Käufers eines Produktes zu einer Kundengruppe erfolgt. Probleme können z. B. auftreten, wenn mehrfach derselbe Betrag ermittelt und abgerechnet würde, z. B. als Discount bzw. als Aufschlag in gleicher Höhe.
- *Preis- und Produktmodelle nicht synchron:* Preis- und Produktmodelle müssen i. d. R. miteinander synchronisiert werden. Insb. im E-Learning werden Lernobjekte als Produkte in mehreren Kursen verwendet. Falls ein Kunde ein Lernobjekt bezahlt hat, muss sichergestellt werden, dass er für die erneute Bereitstellung des Lernobjekts in einem anderen Kurs nicht belastet (charged) wird. Mit PriMoL können Preismodelle für vordefinierte Produkte und Dienste erstellt werden. Eine Konsistenzsicherung über mehrere Preismodelle eines Nutzers wurde nicht implementiert.<sup>148</sup>
- *PriMoL enthält keine Zeitinformationen:* Da PriMoL eine konzeptuelle Beschreibungssprache ist, werden keine Zeitinformationen gespeichert, die darüber Auskunft geben, wann Verbrauchsdaten abgerechnet wurden



(Zeitstempel zur Vermeidung einer doppelten Abrechnung). Es wird vorausgesetzt, dass eine Laufzeitumgebung diese Funktionalität bereitstellt.

- *Preismodelle zu umfangreich*: PriMoL limitiert nicht die Anzahl der Elemente in einem Preismodell. Für das Verständnis eines Preismodells ist eine Beschränkung der Anzahl der Elemente jedoch sinnvoll.

---

<sup>148</sup> Die Konsistenzsicherung kann jedoch dadurch erreicht werden, dass ein drittes Preismodell definiert wird, in dem solche Regeln, z. B. für die Vermeidung einer mehrfachen Bepreisung desselben Produkts, formuliert werden.

## 6 Entwicklung eines Prototyps zur Preismodellierung

Obwohl ein wichtiger Teil des Marketings die Gestaltung von Preisen für Produkte und Dienste im Rahmen der Preis- und Konditionenpolitik – z. B. im Marketing-Mix – bildet, finden in der Marketingliteratur Abrechnungssysteme keine direkte Erwähnung. Stattdessen wird vorausgesetzt, dass Informationssysteme<sup>149</sup> z. B. für das interne Rechnungswesen existieren, die den Marketingverantwortlichen einen Überblick über Verkaufszahlen und Kostenentwicklungen geben, also ähnliche Funktionen wie Abrechnungssysteme bereitstellen (Schewe und Smith 1983), S.86 ff. Diese Systeme können als Teile eines umfassenden Marketing-Informationssystems betrachtet werden, das „den Verantwortlichen im Marketing die benötigten Informationen aufbereiten und zum richtigen Zeitpunkt zuleiten [wird].“ (Kotler u.a. 1999), S. 236. Eine spezifische Betrachtung von Abrechnungssystemen, die das Design von Preismodellen unterstützen, fehlt. Auch die Forschung in Wirtschaftsinformatik bzw. Angewandter Informatik hat sich bisher nur am Rande mit Abrechnungssystemen und der Beschreibung von Preismodellen befasst. Verschiedene Arbeiten haben sich bei ihren Untersuchungen auf ganz spezielle Anwendungsbereiche beschränkt, etwa auf Multimediadienste oder den Mobilkommunikationsbereich (Cushnie 2000; Bushan u.a. 2002), bzw. auf spezielle Fragen wie die Wahrung von Anonymität bei der Abrechnung (Franz u.a. 1998). Eine grobe Klassifikation von Abrechnungssystemen gibt (Abrazhevich 2001). Allerdings sind die beschriebenen Abrechnungssysteme nicht modellbasiert, d.h. zugrunde liegende Preismodelle sind nur verhältnismäßig schwer zu warten. Beim Entwurf von Abrechnungssystemen können Techniken aus dem Software Engineering, z. B. die Informationsbedarfsermittlung und -analyse, eingesetzt werden. (Kotler u.a. 1999), S. 236 rät bei der Einführung zu einem schrittweisen Vorgehen, bei dem nach der Festlegung des

---

<sup>149</sup> Ein Informationssystem verkörpert nach (Böhm u.a. 1996), S. 18 „das reale Modell eines Unternehmens oder von Teilbereichen eines Unternehmens. Je nach Anforderung und Auslegung dient es zur Beobachtung, Gestaltung, Überprüfung, Steuerung, Simulation, Prognose usw. der Geschäftsvorfälle.“ Abrechnungssysteme als Teile von Informationssystemen oder als eigenständige Informationssysteme bieten dem betrieblichen Nutzer Funktionen zur Sammlung, Mediation und Bewertung von Ressourcennutzungsdaten an.

Informationsbedarfs in einem weiteren Schritt festgestellt wird, „wie die benötigten Informationen aus unternehmensinternen Aufzeichnungen, aus [...] Rückmeldungen von den Märkten und aus der [...] Marktforschung gewonnen werden können.“ Eine derart detaillierte Vorgehensweise würde den Rahmen des Kapitels sprengen, müssten doch sämtliche Informationsflüsse eines Unternehmens analysiert und bewertet werden. Stattdessen wird in diesem Kapitel in Abschnitt 6.4 die Architektur von Abrechnungssystemen nach (Hille u.a. 2000; Hille u.a. 2002) skizziert und als idealtypisch angenommen. Eine Auflistung und Kurzbeschreibung industrieller Abrechnungssysteme ist im Anhang „A.4 Abrechnungssysteme – Begriffsdefinitionen“ ab Seite 222 zu finden.

Nach der idealtypischen Beschreibung von Abrechnungssystemen, wird in Kapitel 6.1 die Entwicklung eines *Price-Modeler-Prototyps* skizziert, der die Modellierung von Preismodellen ermöglicht.

## 6.1 Architektur-getriebene Systementwicklung

### 6.1.1 Architekturbegriff

Abrechnungssysteme als Informationssysteme lassen sich durch *Informationssystem-Architekturen* (kurz IS-Architekturen) abbilden. Eine IS-Architektur „enthält die Beschreibung von Elementen, aus denen [Informations-] Systeme gebaut werden, Interaktionen zwischen diesen Elementen, Muster, die deren Zusammensetzung steuern und Einschränkungen in Bezug auf diese Muster.“ (Zuser u.a. 2004), S. 276. Im Software Engineering wird unter dem Begriff Architektur ein „design at the highest level“ verstanden (Braude 2001), S. 250, analog (Mili u.a. 2002), S. 382.

Nach (Foegen 2003), S. 57 erfüllt eine Architektur zwei wesentliche *Ziele*:

*„[Sie] legt das Modell des späteren Systems in einem gewissen Umfang fest und definiert damit die Gegenstandsebene [als präskriptives Modell], u. a. durch die Strukturierung des Systems in bestimmte Subsysteme und durch die Entwicklung von Basiskomponenten wie z. B. einer standardisierten Datenhaltungs-Zugriffsschnittstelle (Persistenzframework).“*

und

*In der Architektur werden Regeln definiert, „die bei der Entwicklung des Systems einzuhalten sind.“*

Architekturen als Modelle dienen außerdem der Dokumentation, Kommunikation und der Verständigung über zu entwerfende bzw. existierende Systeme.

„Die Modellierung der Architektur ist der erste Schritt zum Systementwurf, der dann noch weiter detailliert werden muss.“ (Hasselbring 2006), S. 50.

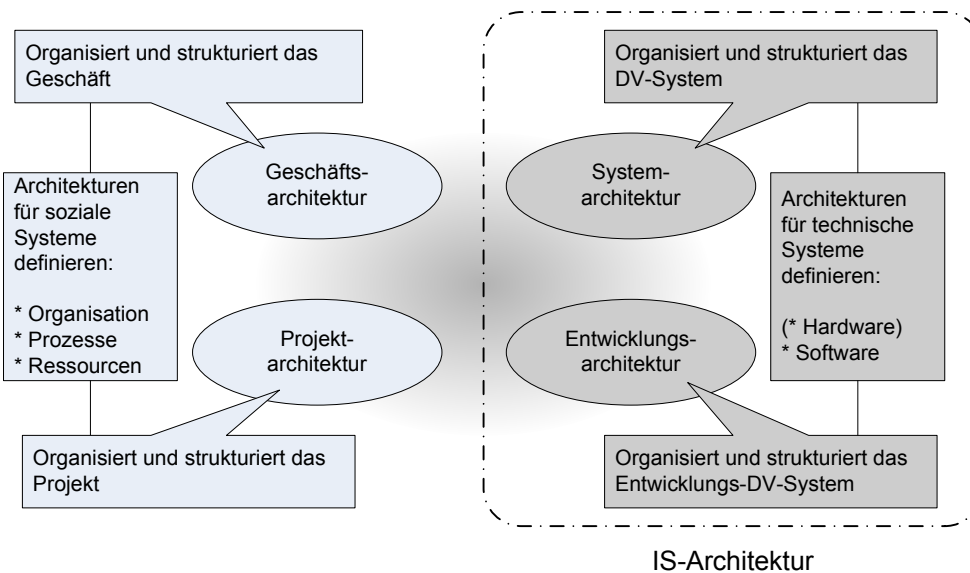


Abbildung 35: Geschäfts-, Projekt-, System- und Entwicklungsarchitektur<sup>150</sup>

Zur Darstellung eines architektonischen Systemmodells sollten gemäß (Sommerville 2001) Modelle zur Beschreibung des Systemverhaltens, so genannte *Verhaltensmodelle*, und zur Abbildung von Strukturen des Systems sowie der zur verarbeitenden Daten, so genannte *Strukturmodelle*, zur Anwendung gelangen. Darüber hinaus sollten Beziehungen eines Informationssystems zu seiner Umwelt in einem *Kontextmodell* abgebildet werden.

### 6.1.2 Architektur-getriebene Entwicklung

Durch eine Architektur-getriebene Software-Systementwicklung kann ein „effizienter Entwicklungsprozesses“ eingeführt (und dokumentiert) werden, indem „Teilaufgaben voneinander entkoppelt und so Arbeitsteilung und eine flexible Projektorganisation ermöglicht [werden].“ (Reussner und Hasselbring 2006), S. 1; (Nord und Tomayko 2006), S. 50-51. Die Architektur dient dabei als Kommunikationsmedium zwischen den Projektbeteiligten und als Wissensspeicher. Sie wird während der Architektur-getriebenen Entwicklung rekursiv

<sup>150</sup> Quelle: modifiziert nach (Foegen 2003), S. 232.

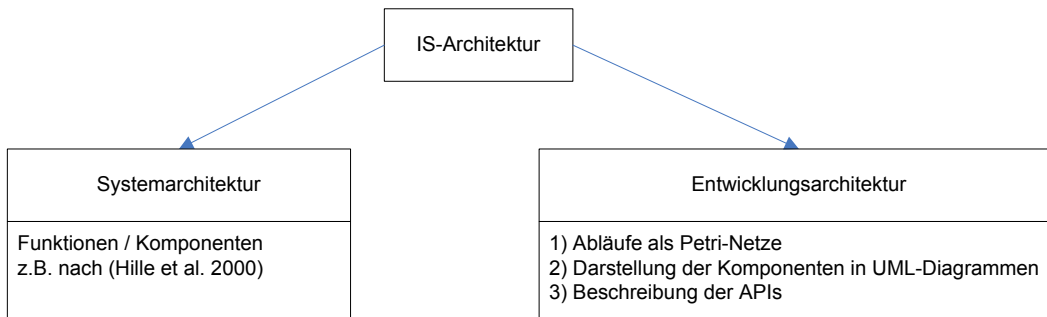
verfeinert. Durch das Architektur-getriebene Vorgehen im Rahmen der Arbeit wird sichergestellt, dass der Aufbau und der Entwurf des entwickelten Software-Prototyps nachvollziehbar sind. Das Vorgehen zur Prototyp-Entwicklung wird in Kapitel 6.5 ab Seite 150 näher beschrieben. Mittels Architekturen werden im Folgenden neben dem allgemeinen Aufbau von Abrechnungssystemen und dem internen Aufbau des Prototyps auch Kommunikations-Schnittstellen zu anderen Software-Systemen (z. B. Billing-Systemen) definiert. Ein Architektur-getriebenes Vorgehen ist im Gegensatz zu Vorgehensmodellen, die in der Software-Industrie Verwendung finden, schlanker. (Nord und Tomayko 2006) zeigen, dass insb. in der Agilen Software-Entwicklung Architektur-getriebene Ansätze zur Qualitätssicherung verwendet werden können. Eine Architektur-getriebene Entwicklung bildet in der Arbeit einen Kompromiss zwischen der Verwendung industrieller Vorgehensmodelle und einer Agilen Vorgehensweise (z. B. eXtreme Programming). Es lässt ausreichend Freiraum zur Definition einer kreativen aber dennoch kontrollierten Entwicklungsmethode für den Software-Prototyp.

## 6.2 IS-Architektur

Der Begriff der *IS-Architektur* wird in der Arbeit wie in Abbildung 35 durch die gestrichelte Linie dargestellt mit der Architektur des DV-Systems (Hard- und Software) gleichgesetzt. Geschäftsarchitekturen und Projektarchitekturen, die in der linken Hälfte der Abbildung zu finden sind und die bei der gewerblichen Entwicklung von Software zu beachten sind, da sie die Geschäftsziele, organisationalen Prozesse und Strukturen sowie die Ressourcen festlegen, werden hier nicht thematisiert.<sup>151</sup> Stattdessen liegt der Fokus auf der *IS-Architektur*, die aus einer softwaretechnischen IS-Architektur, im Folgenden kurz *Systemarchitektur*, und einer softwaretechnischen Entwicklungsarchitektur, kurz *Entwicklungsarchitektur* besteht.

---

<sup>151</sup> Der Begriff der Architektur wird oft auch auf das gesamte Unternehmen bezogen. Zum Architektur-Gesamtbild gehören dann neben der Geschäftsarchitektur (Gesamtzusammenhang der Leistungsverflechtung in einem Wertschöpfungsnetzwerk) die Prozessarchitektur (Leistungsentwicklung, -erstellung und -vertrieb), die Applikationsarchitektur (informatische Verflechtung von Applikationen) und die IT-Architektur (funktionale Verflechtung zwischen Komponenten) (Reussner und Hasselbring 2006), S. 214-215. Die IS-Architektur bildet dabei im Wesentlichen die Applikations- und IT-Architektur ab.



**Abbildung 36: Verwendung des Begriffs Architektur in der Arbeit**

In der *Systemarchitektur* wird auf abstrakter Ebene die funktionale Sicht auf das Informationssystem definiert (vgl. Abschnitt 6.3.2). „Hierzu gehören unter anderem die Struktur und Aufteilung der Softwarekomponenten [und die darin enthaltenen *Funktionen*], die Schnittstellen der Komponenten, die Beziehungen zwischen den Komponenten und die Zusammenarbeit der Komponenten.“ (Foegen 2003), S. 60.

In der *Entwicklungsarchitektur* wird u. a. skizziert, welche Programmiersprache, welche Programmpakete / APIs<sup>152</sup> und welche Entwurfsframeworks bei der Entwicklung eines Prototypen zum Einsatz gelangen (vgl. Kapitel 6.5).

Um eine umfassende Beschreibung eines Informationssystems in Form einer IS-Architektur zu gewährleisten, sollten mit Hilfe von Modellierungstechniken verschiedene Modelle erstellt, zu einer Gesamtsicht integriert und fortentwickelt werden (Denaro und Pezzè 2003), S. 3. In der Literatur vorgestellte Ansätze zur Beschreibung von IS-Architekturen setzen unterschiedliche Akzente bei der Wahl der in Abschnitt 6.3 vorgestellten Modelle.

### 6.3 Modellarten zur Beschreibung von IS-Architekturen

Facetten von Informationssystemen können durch strukturelle, funktionale oder organisationale / hierarchische Modelle erfasst werden (Becker und Schütte 2004), S. 31 ff., 83-116.<sup>153</sup>

<sup>152</sup> Das Application Programming Interface (API) ist eine Sammlung von Softwarekomponenten bzw. -klassen, die bestimmte Funktionen, beispielsweise die Verwaltung von Datenbankabfragen, in einer Software übernehmen.

<sup>153</sup> Oft werden neben diesen weitere Modelltypen zur Beschreibung von Informationssystemen aufgeführt. (Vetter 1998) sieht einen Entwicklungstrend von der funktionsorientierten und datenorientierten hin zu einer objektorientierten Modellierung von Informationssystemen. (Bullinger und Fähnrich 1997), S.16 erweitern die Einteilung von (Becker und Schütte 2004) um eine ereignisorientierte Modellierung, die sie als Vorgängerin der objektorientierten Modellierung ansehen.

### 6.3.1 Strukturmodelle

Nach (Wedekind u.a. 1998; Becker und Schütte 2004) werden in *Strukturmodellen* Elemente und Beziehungen zwischen den Elementen eines Informationssystems abgebildet. Strukturmodelle können im Falle von *Datenmodellen* als Entity-Relationship-Diagramme vorliegen. Neben den Datenmodellen werden auch Prozessschemata, die wiederkehrende Prozesse in einem betrieblichen Informationssystem definieren, in Strukturmodellen, den so genannten *Prozessmodellen*, beschrieben. Prozessmodelle, z. B. die in Kapitel 3 beschriebenen Petri-Netz-Varianten, bilden das Verhalten eines Systems als Folge von Aktivitäten ab. Petri-Netze werden im Rahmen der Arbeit bei der Entwicklungsarchitektur zur Darstellung der Prozesse des Prototyps verwendet.

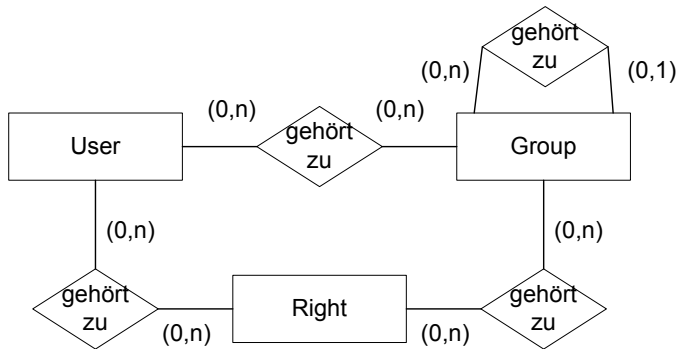
### 6.3.2 Funktionale Modelle

*Funktionale Modelle* teilen das Verhalten eines Systems in unterschiedliche Funktionen auf. Eine *Funktion* kann „als Leistung bestimmter Elemente und Subsysteme definiert werden“ (Becker und Schütte 2004), S. 103. Funktionen können i. d. R. durch Hierarchisierung in weitere Teilfunktionen zerlegt werden. Als Funktionale Modelle werden im Kapitel 6.4 *IDEF0*-Diagramme (vgl. (NIST 1993)) verwendet, um die Komponenten eines Abrechnungssystems abzubilden.

### 6.3.3 Organisationsmodelle

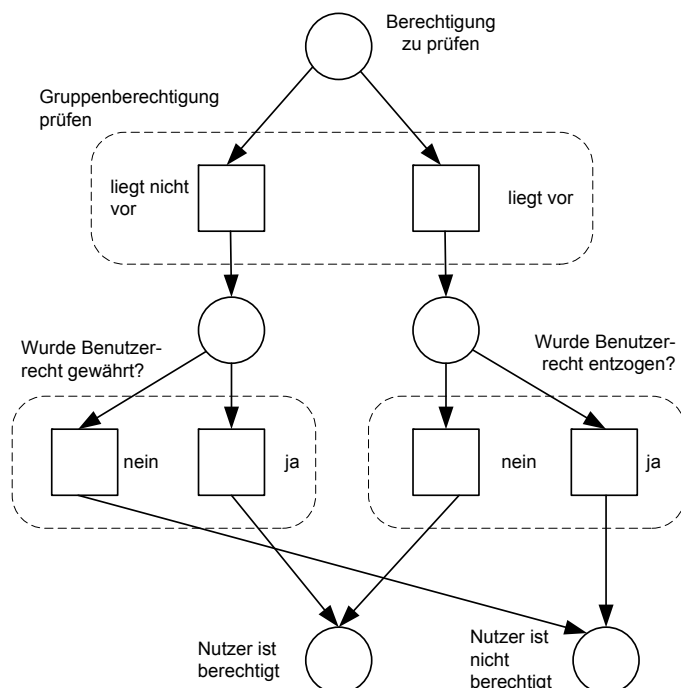
Das Ziel der *Organisationsmodellierung* besteht darin, die Struktur (Aufbauorganisation) und das Verhalten (Ablauforganisation) einer Wirtschaftseinheit abzubilden. Die Ablauforganisation kann mit Hilfe von Prozessmodellen (s. unter 6.3.1) abgebildet werden. Im Rahmen der Aufbauorganisation wird eine Beziehungsstruktur festgelegt, in der den Aufgabenträgern „Funktionen in einem formalen Beziehungsgefüge zugewiesen werden“, die diese zu erfüllen haben (Böhm u.a. 1996), S. 51. Organisationsmodelle finden auch beim Entwurf von Software im Rahmen der Definition von Berechtigungskonzepten Anwendung. *Berechtigungskonzepte* räumen den Anwendern Berechtigungen für bestimmte Funktionen einer Software ein, z. B. den Ausdruck von Dokumenten, oder entziehen diese. Durch die Zuordnung von Anwendern zu Gruppen ist die Einrichtung von Standardberechtigungen möglich. Als Beispiel für ein Organisationsmodell wird

das Berechtigungskonzept des in Kapitel 7 dargestellten E-Learning-Systems im Entity-Relationship-Modell in Abbildung 37 beschrieben.



**Abbildung 37:** Berechtigungskonzept des dem E-Learning-System aus Kapitel 7.1 zugrunde liegenden Content-Management-Systems

In Abbildung 38 wird der interne Mechanismus des E-Learning-Systems als Petri-Netz beschrieben, der bei der Prüfung der Berechtigung eines Anwenders zur Durchführung einer Funktion im E-Learning-System/CMS abläuft. Das Petri-Netz umfasst eine „Handlungsanweisung“, wie in einer Datenbank gemäß Abbildung 37 definierten Berechtigungsregeln zu deuten und anzuwenden sind.



**Abbildung 38:** Petri-Netz zur Ermittlung der Anwenderberechtigung für eine Funktion des E-Learning-Systems aus Kapitel 7.1



## 6.4 Systemarchitektur – Komponenten von Abrechnungssystemen

In diesem Kapitel wird eine Systemarchitektur für Abrechnungssysteme skizziert. Die Darstellung orientiert sich an den Ausführungen von (Hille u.a. 2000; Hille u.a. 2002), die in der GigaABP-Arbeitsgruppe des Telematica Instituts, Enschede, Niederlande eine Architektur für Abrechnungssysteme aufgestellt haben.

### 6.4.1 Kontextmodell

Ein Abrechnungssystem bietet die in Abbildung 39 beschriebenen Funktionen zur wirtschaftlichen Verwertung (Commercial-Exploitation-Support) der Dienstinanspruchnahme durch die Kunden eines Service Providers. Verbräuche, z. B. auf den Clients des Kunden oder in einem Service-Provisioning-System (SPS), müssen durch das Abrechnungssystem gemessen werden und Informationen zu den (bewerteten) Verbräuchen, z. B. in Form von Rechnungen, den Kunden präsentiert werden.

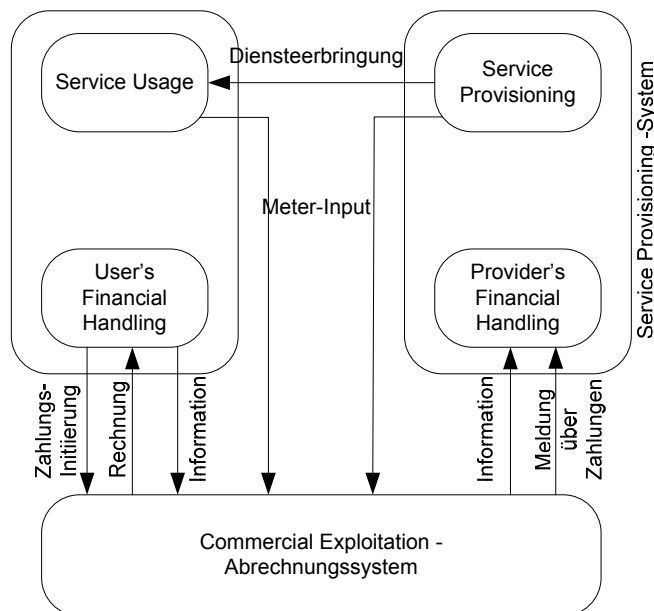


Abbildung 39: Anbindung eines SPS an ein Abrechnungssystem<sup>154</sup>

<sup>154</sup> Quelle: modifiziert nach (Hille u.a. 2002).

## 6.4.2 Komponenten

Die Kernkomponenten und die in den Komponenten enthaltenen Funktionen im Sinne einer Systemarchitektur sind in Abbildung 40, S. 149 dargestellt und reichen von der Datensammlung (technische Dimension) bis zur Bepreisung und Abrechnung der Daten im Billing und Payment (ökonomische Dimension). Jede Funktion führt mit Hilfe so genannter „Mechanismen“ (Datenbanken, Server, Messfühler, Collectors etc.) Transformationen von Eingabe- in Ausgabedaten durch, die durch Controls (Policies, Preismodelle etc.) gesteuert werden.<sup>155</sup>

Die grundlegenden Komponenten eines Abrechnungssystems umfassen nach (Hille u.a. 2002), S. 9-10 (vgl. Abbildung 40):

- 1) die *Usage-Data-Acquisition*, die die Erkennung von Meter Events steuert und Daten über den Ressourcenverbrauch sammelt,
- 2) das *Financial-Settlement*, das für die Abrechnung und das Billing gegenüber Kunden verantwortlich ist, Zahlungen überwacht und Zahlungen mit Rechnungen abgleicht,
- 3) das *Information-Provisioning*, das Kunden und Anbieter mit Informationen versorgt, sowie
- 4) das *Data- & Process-Management*, in dem Daten gespeichert, aggregiert und verarbeitet werden, Geldbeträge den so genannten Charges zugewiesen werden, Verbrauchs- und Konto-Daten archiviert werden und der Zugriff für externe Partner geregelt wird.

### 6.4.2.1 Usage-Data-Acquisition-Komponente

Die *Usage-Data-Acquisition* sammelt Verbrauchsdaten über die vom Kunden angeforderten Dienste, Ressourcen und Inhalte, die die Voraussetzung für die betriebswirtschaftliche Verwertung bilden. *Meter*, d.h. Messpunkte für das Messen von Verbräuchen, können z. B. auf Clients von Kunden, in Netzwerk-Routern oder auch in den Applikationen eines Anbieters implementiert werden. Die Messung mittels *Metern* erfolgt kontinuierlich in Echtzeit z.B. durch Protokollierung in Logfiles, Datenbanken oder Routern. Im *Collecting* werden Daten aus unterschiedlichen Messpunkten gelesen, ggf. konsolidiert und verdichtet. *Collectors* überstellen die verdichteten Daten an eine Accounting-Funktion. *Collectors* werden oft auch dazu verwendet, (1) Daten zu filtern, d.h.

nur die Daten weiterzuleiten, die bestimmten Anforderungen genügen, (2) nur aggregierte Daten weiterzuleiten, z. B. um den Netzwerkverkehr zu verringern, (3) Daten für einen bestimmten Zeitraum zu puffern und (4) Daten in Paketen (Batches) weiterzuleiten.

#### 6.4.2.2 Data- & Process-Management-Komponente

Das *Accounting* selbst kann wieder in zwei Teilfunktionen aufgeteilt werden (nicht in der Abbildung dargestellt). Die erste Funktion ist das *Usage-Data-Logging*, das die eigentlichen Nutzungsdaten aus der Accounting-Data-Acquisition verwaltet. In der zweiten Funktion, dem *Account-Management*, werden Daten zu den einzelnen, internen Kunden-Konten<sup>156</sup> des Providers gespeichert. Hierzu gehören Außenstände eines Kunden oder Kontensalden eines Prepaid-Kontos.

Die *Charging-Funktion* berechnet die Charge für einen oder mehrere Kunden ausgehend von den im Usage-Data-Logging aufbereiteten Daten (vgl. Abschnitt 6.4.2.1) und einer *Pricing-Policy*. Die Charging-Funktion führt die Berechnung einer Charge durch (*Charge-Calculation*) und teilt die Charges den Kunden zu (*Charge Distribution*).

Das *Pricing* bestimmt die anzuwendenden Tarife / Preismodelle. Neben dem Nachschlagen (Lookup) von Parameterwerten in einem Preismodell – dem *statischen Pricing* – kann das Pricing auch selbst durch die Accounting-Data beeinflusst werden. Im zweiten Fall sprechen (Hille u.a. 2002) von einem *dynamischen Pricing*.<sup>157</sup>

#### 6.4.2.3 Financial-Settlement-Komponente

Im Financial Settlement werden das Billing und das Payment abgewickelt.

Die *Billing-Funktion* kann weiter in eine *Bill-Preparation-Funktion* zur Erzeugung von Rechnungen und eine *Bill-Presentation-Funktion*, die der Vorlage der Rechnung gegenüber dem Kunden dient, unterteilt werden. Das *Payment*

---

<sup>155</sup> Für eine detaillierte Beschreibung von Funktionen im IDEF0-Standard s. (NIST 1993).

<sup>156</sup> Es handelt sich nicht um Konten des Kunden bei einem Kreditinstitut. Diese sind erst in der Payment-Phase relevant.

<sup>157</sup> Unter dynamischem Pricing wird in der Arbeit im Gegensatz zur Definition von (Hille u.a. 2002) eine dynamische Preisbildung bspw. durch Auktionierung verstanden. Die in der Arbeit entworfene Beschreibungssprache unterstützt über die Mediation-Expressions (s. Anhang „A.2.1 Accounting-Language-Definition in EBNF“) die Definition von dynamischen Preismodellen nach der Definition von (Hille u.a. 2002).

umfasst die Unterfunktionen *Payment-Transaction-Processing* und *Settlement*. Im *Payment-Transaction-Processing* wird die Autorisierung für Zahlungen angefordert. Das *Settlement* entspricht der Übertragung von monetären Werten als Teil des Erfüllungsgeschäfts.

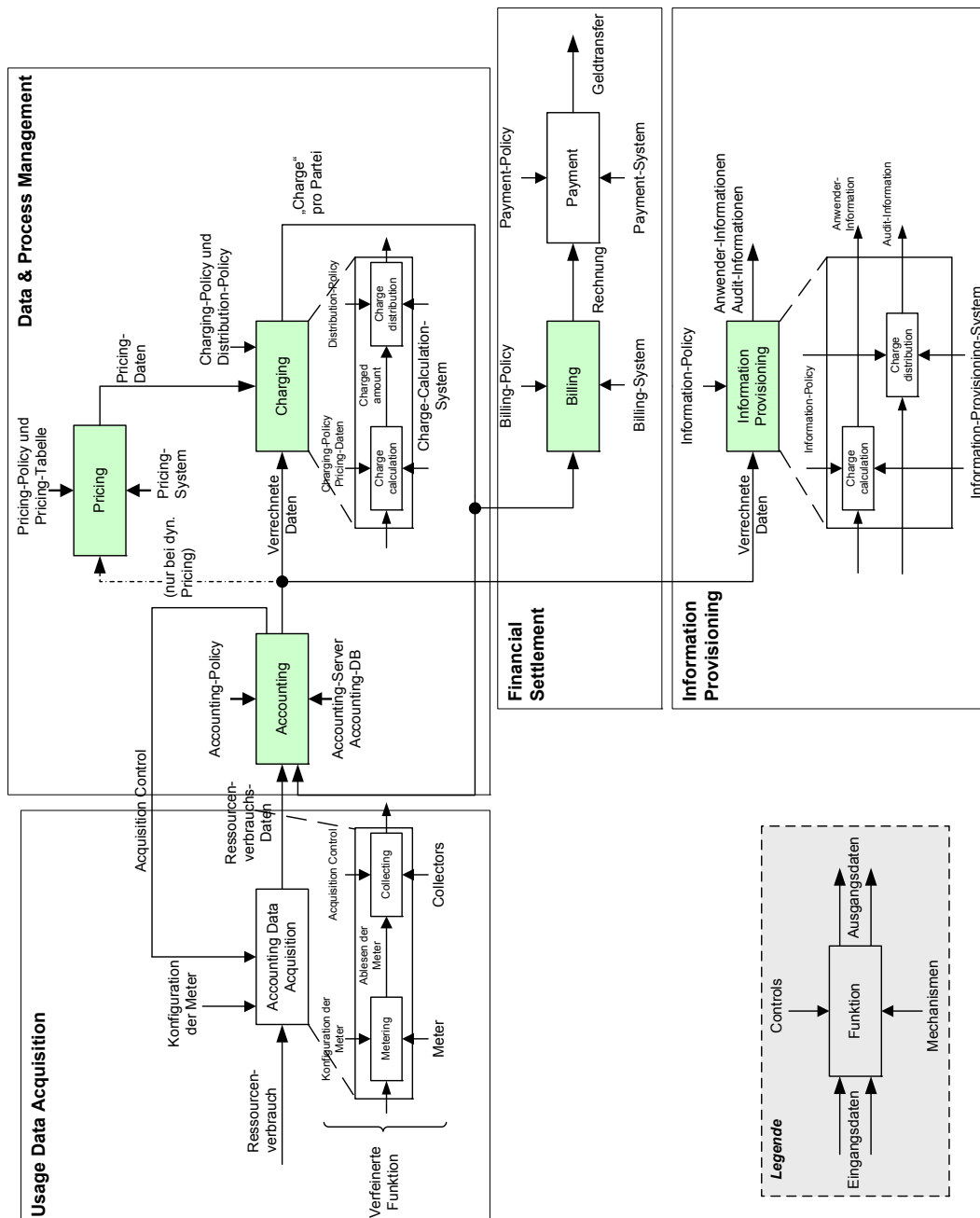


Abbildung 40: Commercial-Exploitation-Komponenten eines Abrechnungssystems<sup>158</sup>

<sup>158</sup> Quelle: modifiziert nach (Hille u.a. 2000; Hille u.a. 2002). Der Begriff Meter wird hier, im Gegensatz zur englischen Schreibweise, auch für den Plural verwendet.

#### **6.4.2.4 Information-Provisioning-Komponente**

Die *Information-Provisioning-Komponente* versorgt die Kunden und den Anbieter der digitalen Produkte mit Information zur Dienstenutzung, zu fälligen Beträgen usw. Das Information-Provisioning kann daher als Basis für Kapazitätsplanungen und für eine kundenindividuelle Ansprache durch den Anbieter dienen. Es ist als Zusatzfunktion zu betrachten, da es für die Abrechnung von Verbräuchen nicht benötigt wird.

### **6.5 Vorgehensmodell zur Architektur-getriebenen Entwicklung eines Price-Modeler-Prototyps**

Die Entwicklung des Software-Prototyps erfolgt in einem iterativen Prozess. Nachdem in den Kapiteln 5.4.2 - 5.4.6 die Ziele der Entwicklung im Sinne einer Anforderungsspezifikation z. T. durch die Definition der Beschreibungssprache beschrieben wurden, soll die Funktionalität des zu erstellenden Prototyps mit Hilfe einer architekturgetriebenen Vorgehensweise<sup>159</sup> so weit verfeinert werden, dass diese durch einen lauffähigen Java-Prototyp umgesetzt werden kann. Dazu wird im folgenden Kapitel vorab der Begriff der *Architektur von Software-Systemen* (IS-Architektur) beschrieben. Das Vorgehen bei der Entwicklung, dem der Aufbau der folgenden Kapitel folgt, ist in Abbildung 41 dargestellt. In einem ersten Schritt wurde dazu in Kapitel 6.4 die Systemarchitektur von Abrechnungssystemen nach (Hille u.a. 2000; Hille u.a. 2002) beschrieben. Die Aufstellung einer Systemarchitektur dient in der Arbeit dazu, den Gegenstandsbereich der Prototypentwicklung abzugrenzen. Sie liefert zudem einen Überblick über allgemeine Komponenten von Abrechnungssystemen, wobei nur ein Teil der beschriebenen Komponenten implementiert werden soll. Die für den Prototyp als relevant erachteten Komponenten müssen für die Umsetzung in einem lauffähigen Programm verfeinert werden. Dazu wird in der statischen Entwicklungsarchitektur in Kapitel 6.6.2 eine Drei-Ebenen-Architektur mit Zwischenschichten und in Kapitel 6.6.3 eine Prozessarchitektur (dynamische Entwicklungsarchitektur) beschrieben. Die Prozessarchitektur definiert die Anwendungslogik des Prototyps mittels Petri-Netzen. Die Verwendung von Petri-Netzen für die Modellierung der Anwendungslogik bietet insbesondere Vorteile

bei der Analyse der Prozessschemata und bei der schrittweisen Verfeinerung der dargestellten Prozesse. Eine ausführliche Darstellung der Eignung von Petri-Netzen im Software Engineering ist im Anhang „A.3 Petri-Netze und Software Engineering“ ab S. 206 zu finden.

Abbildung 41 stellt das *Vorgehensmodell* bei der Entwicklung des Prototyps dar. Die ersten Phasen des Entwicklungsprozesses (Zieldefinition und Systemarchitektur) folgen linear aufeinander. Im Gegensatz dazu erforderte die Implementierung des Prototyps ein inkrementelles Vorgehen, vgl. z. B. (Mills 1980, 1988) und (Sommerville 2001), S. 51-53. Der Prototyp wurde regelmäßig gegenüber der Entwicklungsarchitektur validiert, und Erkenntnisse insbesondere bei der Gestaltung der Wizards während der Programmierung des Prototyps flossen in die Darstellung der Prozesse in der Entwicklungsarchitektur zurück.

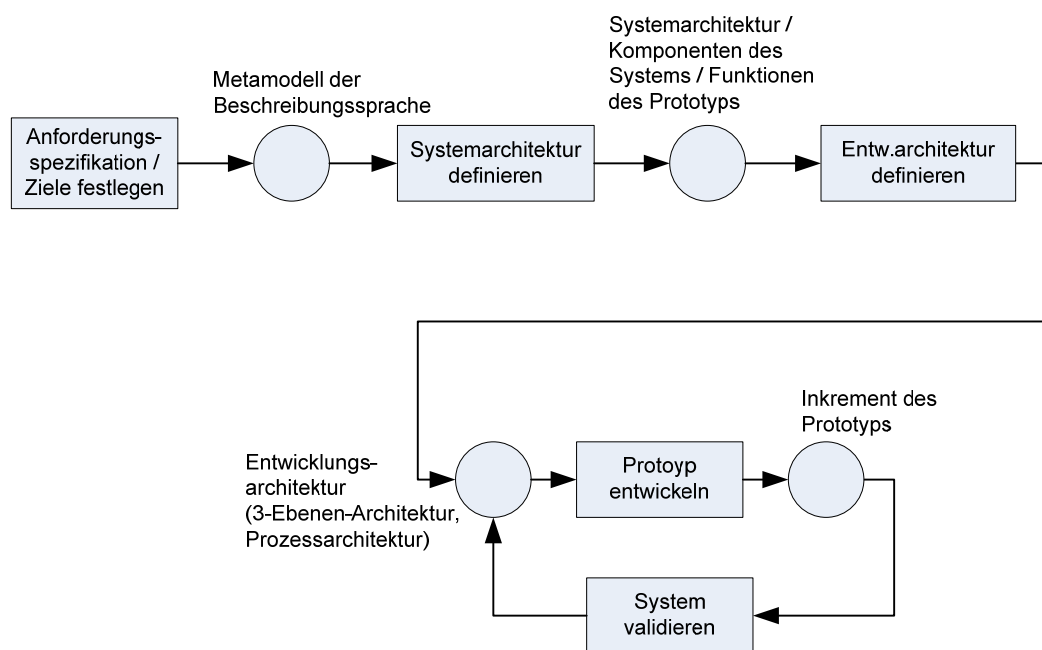


Abbildung 41: Petri-Netz des Vorgehensmodells zur Entwicklung des Price-Modeler-Prototyps

## 6.6 Entwicklungsarchitektur

### 6.6.1 Auswahl der zu implementierenden Funktionen

Im Prototyp werden nur die Funktionen eines Abrechnungssystems implementiert, die für die Implementierung notwendig sind und mit deren Hilfe Konzepte der

<sup>159</sup> Für eine Beschreibung des Architecture-Driven-Development (Architektur-getriebene Softwareentwicklung) vgl. (Nord und Tomayko 2006).

Beschreibungssprache vollständig umgesetzt werden können. Diese sind in Abbildung 40 hervorgehoben.

Die Auswahl der Komponenten ergibt sich aus den folgenden Überlegungen:

- 1) Bei der Entwicklung eines Meters sind spezielle Kenntnisse der zugrunde liegenden Systeme unerlässlich. Eine Metering-Software muss genau auf das jeweilige Service-Provisioning-System abgestimmt sein und darf insbesondere dessen Betrieb nicht stören. Zwischen der Entwicklung von Metern und Preismodellen besteht eine Wechselbeziehung: Zum einen kann eine Bepreisung nur für tatsächlich gemessene Daten erfolgen, sofern nicht eine reine Flat-fee- oder zeitabhängige Bepreisung durchgeführt wird. Und zum anderen sollten nur Meter für Komponenten implementiert werden, die für die Durchführung betrieblicher Aufgaben, z. B. das Pricing mittels Preismodellen, wichtig sind. Das Metering wird aufgrund seiner Komplexität und seiner Spezifität nicht betrachtet. Da Metering und Collecting Daten aufbereiten, die für das Accounting benötigt werden, und diese entsprechend den Erfordernissen des Accountings an die Accounting-Funktion übergeben, wird weiterhin angenommen, dass die gemessenen Daten entweder bereits aufbereitet in einer SQL-Datenbank vorliegen oder direkt auf die SQL-Datenbasis des Service-Provisioning-Systems, die die Produktivdaten enthält, zugegriffen werden kann. In der in Kapitel 7 dargestellten Fallstudie im E-Learning werden daher zur Durchführung der hervorgehobenen Funktionen SQL-Abfragen direkt an die Datenbasis des prototypischen E-Learning-Systems gestellt.
- 2) Auch für die Durchführung des Payments sind spezifische technische Kenntnisse, z. B. in der Verschlüsselung und in Signaturen, sowie rechtliche Kenntnisse, z. B. bei der Erstellung von Mahnungen und der Überwachung von Einlösefristen, erforderlich. Da die abgerechneten Daten sowohl in einem relationalen, als auch in einem XML-Format aufbereitet werden können, ist davon auszugehen, dass die Übergabe der Abrechnungsdaten an ein Payment-System mit geringem Aufwand möglich ist. Der Prototyp selbst umfasst keine Payment-Komponente.
- 3) Der Schwerpunkt der Prototypentwicklung liegt in der Bereitstellung der Pricing-Funktionen. Für das Billing, Accounting und Charging wurde eine separate Laufzeitumgebung programmiert, die mit geringem Aufwand in

ein Service-Provisioning-System integriert werden kann.<sup>160</sup> (s. Kapitel 7.1.1)

## 6.6.2 Statische Entwicklungsarchitektur

In Tabelle 18 ist die statische Architektur des Prototyps zur Erstellung von Preismodellen dargestellt. Die Gestaltung des Price-Modelers folgt einer Drei-Ebenen-Architektur mit Zwischenschichten. Als Entwicklungssprache für den Prototyp wird Java verwendet. Eine alternative Implementierung könnte ohne Einschränkung der Funktionalität mittels Dotnet z. B. in C# erfolgen.

<b>Ebene</b>	<b>Übersicht</b>	<b>Bibliotheken / Produkte / Standards</b>
Präsentationsschicht	Grafische Oberfläche (GUI) des Modelers und für weitere Werkzeuge wie Workflow-Clients, Simulator, Runtime	<i>Renderer</i> : JGraph, SVG-Viewer GUI: Swing-Widgets
Model-Mapping-Layer (Zwischenschicht)	Konvertierung von Speicherobjekten in grafische Modelle, die durch die Anwendungsschicht interpretiert werden können und vice versa,	<i>JGraph</i> ↔ JDOM
Anwendungsschicht	Wahrung der Integritätsbedingungen in einem Preismodell (Parsing), Verwaltung einzelner Preismodelle, DB-Verbindungen etc.	Geschäftslogik als Java-Code und Handling der Modelle in JDOM
Persistenzschicht (Zwischenschicht)	DB-Mounter, De-/Serializer (Objects-to-SQL)	JDOM, Eigenentwicklung
Daten(-bank)-schicht	DB-Engines, MetadatenSpeicherung, Repositorien für Preismodelle	MySQL, Oracle, MSSQL

**Tabelle 18: Drei-Ebenen-Architektur des Price-Modelers mit Schichten**

### 6.6.2.1 Präsentationsschicht

Anwender kommunizieren mit dem Price-Modeler über die *Präsentationsschicht*. Diese bietet z. B. in Form von Menüs, Buttons und Eingabefeldern (so genannten Widgets) einen Zugriff auf die Funktionalität des Modelers. Die Menüstruktur

<sup>160</sup> Die Laufzeitumgebung wurde in PHP implementiert. Eine Programmierung der Laufzeitumgebung als Web-Service wird im PREMIUM-Projekt weiterverfolgt. Siehe dazu auch



wird in *Swing* (Loy 2002) entwickelt<sup>161</sup>. Neben Standardmenüs zur Dateiverwaltung soll die Basisfunktionalität zur Erstellung von Preismodellen (Einfügen von komplexen und einfachen Preisen sowie Aggregationsregeln) über Menüs erreichbar sein. Anwender werden bei der Erzeugung von Modellelementen durch *Wizards* unterstützt.

Die Zeichenfläche für Modelle in der grafischen Oberfläche des Prototyps besteht aus einer *JGraph*-Komponente<sup>162</sup>. *JGraph* (Alder 2002) ([www.jgraph.com](http://www.jgraph.com)) ist ein Swing-kompatibles API, das die Verwaltung von Graphenstrukturen<sup>163</sup> in Java erlaubt und Java-Klassen für Kanten, Knoten und deren Rendering bereitstellt.

### 6.6.2.2 Model-Mapping-Layer

In der Model-Mapping-Layer werden u. a. die Funktionsaufrufe / Aktionen des Anwenders zur Erstellung von grafischen Elementen im Preismodell an die Anwendungsschicht weitergeleitet. Dabei werden Events in der grafischen Oberfläche des Prototyps, z. B. das Klicken eines Buttons in Objektmethoden in der Anwendungsschicht gemappt. Methoden der Anwendungsschicht wiederum übergeben neue Informationen, z. B. das Update eines Preiselements, an die Model-Mapping-Layer. Ein Modell, das in den Speicher geladen wird, wird durch die Anwendungsschicht aufbereitet und an die Model-Mapping-Layer weitergeleitet. Diese übernimmt die Kommunikation mit der Präsentationsschicht, indem sie die entsprechenden Informationen, die für das Rendering oder die Aktualisierung von Menübefehlen wichtig sind, weiterleitet.

### 6.6.2.3 Anwendungsschicht

Die Anwendungsebene übernimmt zwei wesentliche Aufgaben. Zum einen werden Preismodelle als XML-Baum in einem *JDOM*-Tree ([www.jdom.org](http://www.jdom.org)) vorgehalten<sup>164</sup> und zum anderen kapselt sie die *Verarbeitungslogik* der Anwendung. *JDOM* ist ein API, das auf Collections als Datentyp aufbaut und das Lesen, Schreiben und Manipulieren von XML-Bäumen vereinfacht. Für das

---

die Hinweise im Ausblick.

<sup>161</sup> Die Dokumentation zu Swing ist unter folgender Internet-Adresse zu finden: <http://java.sun.com/j2se/1.5.0/docs/api/javaw/swing/package-frame.html>

<sup>162</sup> Eine *JGraph*-Komponente ist von der Klasse *JPanel*, einer Zeichenfläche, abgeleitet.

<sup>163</sup> Einführungen in die Informationsvisualisierung mit Graphen, Algorithmen und Übersichten über APIs sind z. B. in (Dogrusoz u.a. 2002) und (Herman u.a. 2000) zu finden.

<sup>164</sup> Alternativ zur Speicherhaltung in einem XML-Baum mittels *JDOM* könnten Modelle auch in einem *JTree* vorgehalten werden.

Parsing von XML können in JDOM sowohl *SAX* (Simple API for XML, [www.saxproject.org](http://www.saxproject.org)), als auch *DOM* (Document Object Model, [www.dom4j.org](http://www.dom4j.org)) verwendet werden (vgl. (McLaughlin 2002)).

Die Verarbeitungslogik der Anwendung ist in mehrere Pakete<sup>165</sup> unterteilt (s. Kapitel 6.6.4). Neben Paketen für die Verwaltung von XML-Strukturen wird auch ein Paket für die Verwaltung von Datenbankverbindungen (Mountpoints) auf der Anwendungsebene bereitgestellt.

#### 6.6.2.4 Persistenzschicht

Die Persistenzebene übernimmt als Zwischenschicht die *Kommunikation mit Datencontainern* auf der Datenbankebene mittels *JDBC* (<http://java.sun.com/products/jdbc/>). Durch die Persistenzebene wird das Öffnen von Datenbankverbindungen und das Schreiben bzw. Lesen von Modellen über die Datenbankverbindungen realisiert. Das Schreiben in XML wird durch JDOM-Funktionen geregelt.<sup>166</sup>

#### 6.6.2.5 Datenbankschicht

Der Price-Modeler kann auf alle relationalen Datenbanken zugreifen, die eine JDBC-Schnittstelle anbieten. Zu Testzwecken wurde eine *MySQL-Datenbank* aufgesetzt.<sup>167</sup> In Datenbanken werden die erzeugten Modelle gemäß des in Kapitel 5.4.6 spezifizierten relationalen Modells abgelegt. Darüber hinaus wird vereinfachend bei der Entwicklung davon ausgegangen werden, dass die aufbereiteten Verbrauchsdaten, die zur Bepreisung anstehen, in einer relationalen Datenbank des Service-Provisioning-Systems vorgehalten werden.

### 6.6.3 Dynamische Entwicklungsarchitektur

Die folgenden mit CPN Tools modellierten, Gefärbten Petri-Netze dienen als Grundlage zur Implementierung der Prozesslogik des Prototyps. Es werden Prozesse zur Definition der Bestandteile eines Preismodells, d.h. Prozesse für die Definition von einfachen Preisen, komplexen Preisen und Aggregationsregeln,

---

<sup>165</sup> Mit Paketen ist es in Java möglich, Klassen zu logischen Einheiten zu gruppieren. Pakete können ineinander geschachtelt werden.

<sup>166</sup> Für die Speicherung der Modelle in relationale Datenbanken wurde in einem ersten Prototyp ein eigenes Paket entwickelt. In einer Folgeversion soll auf bestehende Persistenzframeworks z. B. Hibernate ([www.hibernate.org](http://www.hibernate.org)) oder Castor ([www.castor.org](http://www.castor.org)) zurückgegriffen werden.

<sup>167</sup> Auch für DB2-, Oracle- und MS SQL-Datenbanken stehen JDBC-Schnittstellen zur Verfügung.

beschrieben. Sämtliche Petri-Netze sind als Bedingungs-Ereignis-Netze zu interpretieren.<sup>168,169</sup>

### 6.6.3.1 Anwendungsfälle

In Abbildung 42 sind die Anwendungsfälle des Prototyps dargestellt. Die zentrale Aufgabe des Prototyps liegt in der Erstellung von Preismodellen. Hierzu muss ein Benutzer *Modellkomponenten anlegen und löschen* können. Die erstellten Preismodelle müssen verwaltet, d.h. gespeichert und geladen werden (*Modelle verwalten*). Dazu gehört auch der Export von Preismodellen in XML. Der Prototyp soll neben dem Erstellen von Preismodellen die Auswertung von Modellen anhand der Daten eines Service-Provisioning-Systems (SPS), die durch die Dienstinanspruchnahme und das Downloadverhalten eines Anwenders entstehen, ermöglichen. Dazu ist es notwendig, dass der Prototyp Datenbankverbindungen verwalten kann (Anwendungsfall *DB-Verbindungen verwalten* in Abbildung 42). Darüber hinaus soll eine Speicherung von Preismodellen entsprechend dem in Kapitel 5.4.6 vorgegebenen relationalen Datenbankschema möglich sein. Die blau hinterlegten Textblöcke in Abbildung 42 symbolisieren eine Verfeinerung der gleichnamigen Transitionen. Die Transition *DB-Verbindungen verwalten* wird bspw. in einem gefärbten Petri-Netz auf einer gleichlautenden Seite<sup>170</sup> verfeinert. Gleiches gilt für die übrigen Anwendungsfälle.

---

<sup>168</sup> Die in den Abbildungen zu findende Kapazität  $E$  der Stellen ist ein Colour-Set auf Nulltupel. Die Anfangsmarkierung der Stelle „Warte auf Benutzeraktion“ ist  $e$  bzw.  $1^e$ , d.h. ein einzelnes Nulltupel. Vgl. Kapitel 3.3.4 Die Kantenbeschriftungen *Marker* bzw.  $t$ ,  $t1$ ,  $t2$  und  $t3$  entspricht einer Variable, die nur ein Nulltupel aufnehmen kann. Grün umrandete Transitionen sind aktiviert. Die Belegung einer Stelle wird mit einem grünen Kreis kenntlich gemacht, der die Anzahl der enthaltenen Marken angibt. Daneben wird das enthaltene Element in einem grünen Kästchen aufgeführt.

<sup>169</sup> Die grafische Aufbereitung und das Parsing von Modellen werden nicht beschrieben, da sie den Umfang der Arbeit sprengen würde.

<sup>170</sup> Seiten (Pages) werden in CPN Tools dazu verwendet, Netze logisch aufzuteilen. Netze auf unterschiedlichen Seiten können miteinander verknüpft sein. Die Verknüpfung wird durch Input- und Output-Stellen ermöglicht.

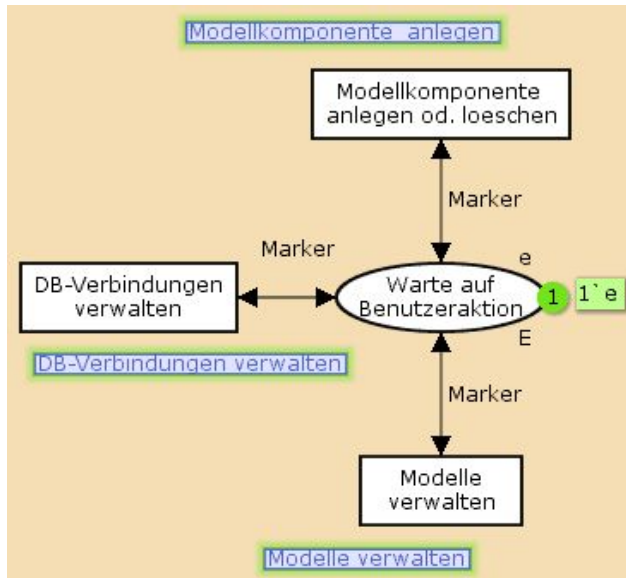


Abbildung 42: Anwendungsfälle des Prototyps als CP-Netz

Die Funktionalität der Anwendungsfälle wird dem Anwender durch die Benutzeroberfläche des Price-Modelers (Graphical User Interface, GUI) zugänglich gemacht (vgl. Abbildung 43). Neben Funktionen zum Lesen und Schreiben von Modellen ist z. B. unter dem Menüpunkt „Data Handling“ die Verwaltung von Datenbankverbindungen möglich. Die Umsetzung der als CP-Netze beschriebenen Geschäftsvorfälle im Prototyp wird in Kapitel 6.6.3.5 anhand der Anlage eines komplexen Preises – stellvertretend für die übrigen Prozesse – beschrieben.

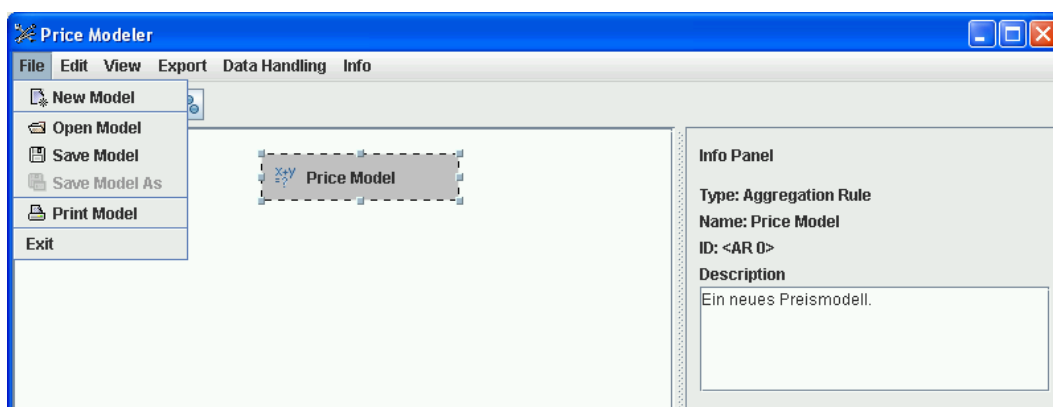
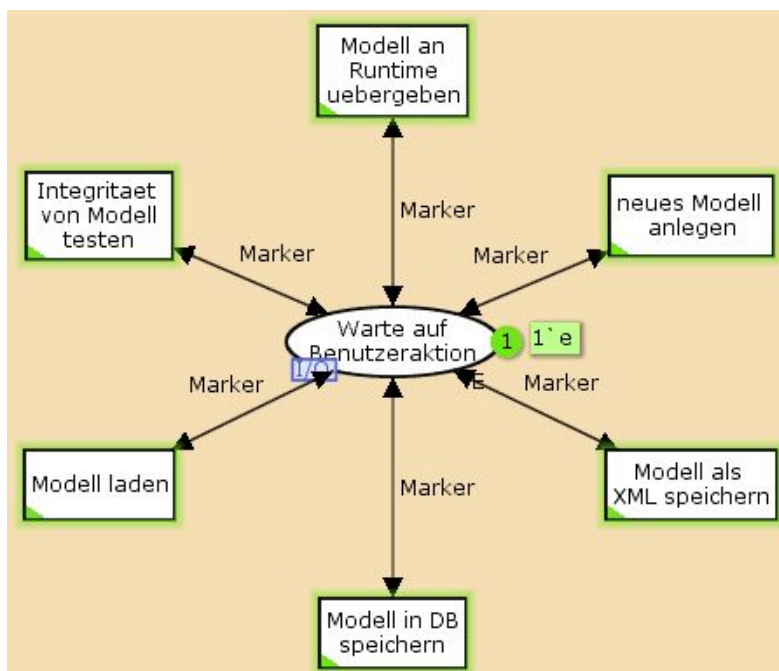


Abbildung 43: GUI des Price-Modelers

### 6.6.3.2 Modelle verwalten

Der Anwendungsfall „Modelle verwalten“ wird in Abbildung 44 spezifiziert. Neben dem Anlegen eines neuen Modells hat der Anwender die Möglichkeit,

Modelle in XML zu speichern und gespeicherte Modelle zu laden. Daneben kann er die Integrität von Modellen testen, die mit anderen Werkzeugen erstellt wurden. Der Integritätstest findet beim Laden des Modells statt. Die Übergabe von Modellen an eine Laufzeitumgebung (Runtime) ist durch Wahl eines Speicherortes möglich, auf den die Laufzeitumgebung zugreift. Die Verknüpfung eines Modells mit einem konkreten Produkt wird in Kapitel 7.1.4.4 am Beispiel eines E-Learning-Systems demonstriert.



**Abbildung 44: Anwendungsfall Modelle verwalten als CP-Netz**

Das Speichern der Modelle in Relationen einer SQL-Datenbank ist durch die Definition einer relationalen Notation der Sprache vorgesehen. Abbildung 43 zeigt das Menü zur Verwaltung von Modellen in der Prototyp-GUI.

### 6.6.3.3 Datenbank-Verbindungen verwalten

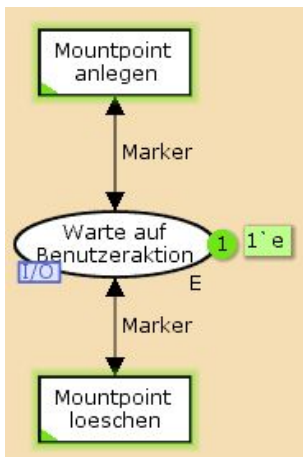


Abbildung 45: Datenbank-Verbindungen verwalten als CP-Netz

Zur Verwaltung von Datenbanken können Mountpoints angelegt und gelöscht werden. Im CP-Netz in Abbildung 46 wird das Anlegen eines Mountpoints dargestellt. Für die Adressierung der Datenbank ist die Definition des Host-Namens, der Port-Nummer und des Datenbanktreibers notwendig. Darüber hinaus muss ein gültiger User angegeben werden, der den Zugriff auf die benötigten Tabellen gestattet.

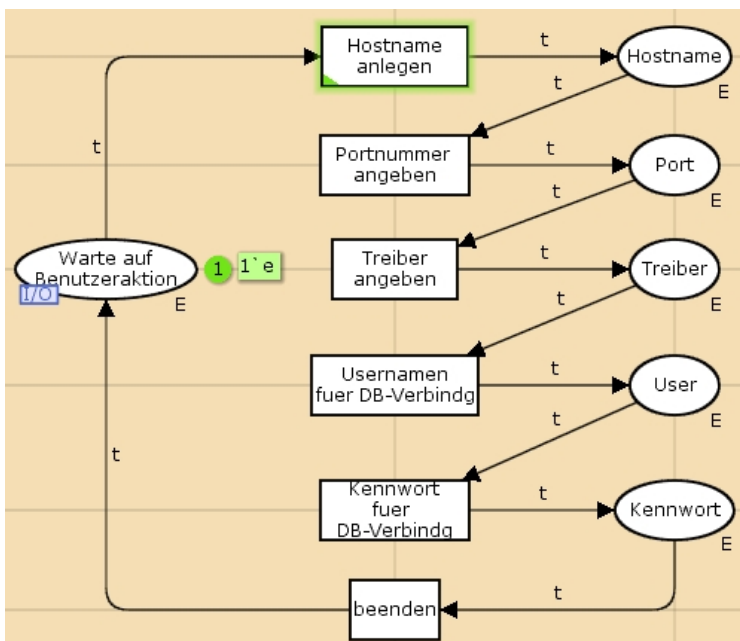


Abbildung 46: Mountpoint anlegen als CP-Netz

Die Einrichtung einer Verbindung über eine Eingabemaske des Prototyps wird in den folgenden Screenshots beschrieben. Die Masken wurden ausgehend von der

CP-Netz-Spezifikation programmatisch umgesetzt. Um die Unterscheidung von Verbindungen zu ermöglichen und Verbindungen zu verwalten, wurde zusätzlich ein Attribut „Mountpoint Name“ implementiert.

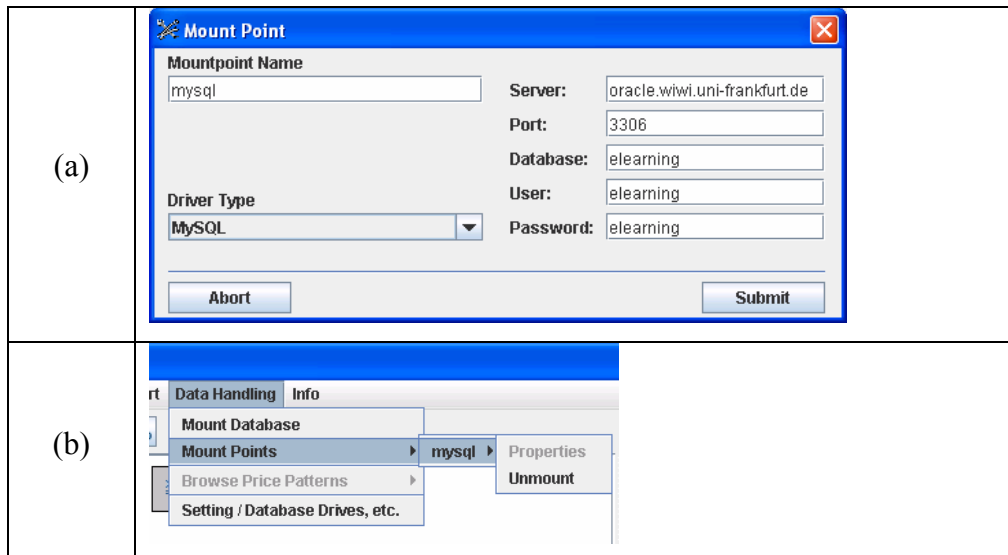


Abbildung 47: Maske zum Anlegen eines Mountpoints im Price-Modeler

Nach dem Anlegen des Mountpoints (vgl. Abbildung 47 (a)) wird dieser in einem Menü aufgelistet und ist für die Verwendung in einfachen Preisen und komplexen Preisen sichtbar (vgl. Abbildung 47 (b)). Das Löschen der Datenbankverbindung erfolgt durch Ausführen des Befehls „Unmount“.

### 6.6.3.4 Modellkomponente anlegen oder löschen

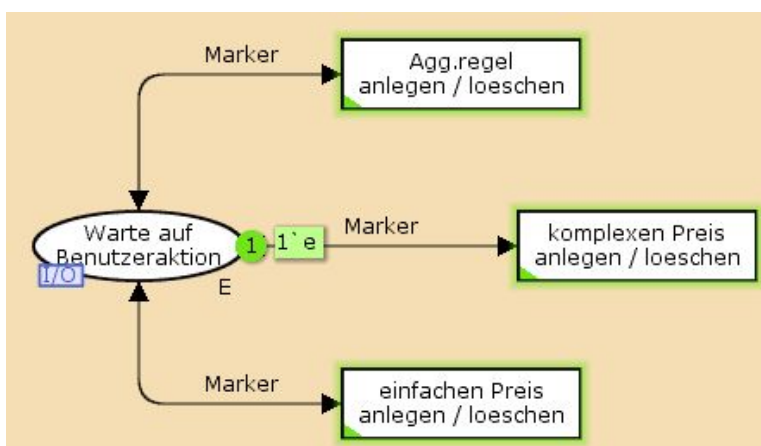


Abbildung 48: Modellkomponente anlegen oder löschen als CP-Netz

Abbildung 48 beschreibt den Anwendungsfall *Modellkomponente anlegen oder löschen*. Die Beschreibungssprache PriMoL umfasst die Modellkomponenten Aggregationsregel, komplexer Preis und einfacher Preis. Entsprechend wird

*Modellkomponente anlegen* durch stellenberandete Netze zum Anlegen eines komplexen Preises, einfachen Preises und einer Aggregationsregel verfeinert.

Das Löschen eines Elements wird an dieser Stelle nicht als Petri-Netz beschrieben. Beim Löschen werden das betreffende Element und im Falle von Aggregationsregeln die eventuell in der Baumstruktur des Preismodells untergeordneten Elemente aus einem Preismodell entfernt. Vor dem Anlegen einer Modellkomponente muss die übergeordnete Aggregationsregel angegeben werden.

### 6.6.3.5 Komplexen Preis anlegen

Ein komplexer Preis beinhaltet Regeln zur Bepreisung von *Datentupeln*. Durch die Aktionen eines Kunden in einem Service-Provisioning-System werden Datentupel in eine Datenbanktabelle bzw. mehrere Tabellen geschrieben. Diese umfassen neben einem Attribut, das den Kunden identifiziert, mehrere Attribute zur Dienstanspruchnahme durch den Kunden (z. B. Länge eines Telefonates oder Anzahl der übertragenen Bytes in einer Internet-Verbindung), die sich für eine Bepreisung eignen. Pro Kunde können beliebig viele Tupel entstehen. Für die Bepreisung ist es wichtig, Tupel nach Kunden zu ordnen, anschließend den Tupeln individuelle Preisähler zuzuweisen und danach die Menge der Preisähler zu einem Kunden mit einem algebraischen Ausdruck zu verdichten. Durch die Einführung beliebig komplexer algebraischer Ausdrücke können bspw. Staffelpreise eingeführt werden.

Das Petri-Netz aus Abbildung 49 ist wie folgt zu deuten: Das Anlegen eines komplexen Preises besteht darin, den *Attributen* eines SQL-Ausdrucks *Dimensionen* zuzuordnen. Die Kombinationen der möglichen Attributausprägungen liefern *Indizes* für ein n-dimensionales Array. Zu einem individuellen Index kann ein Wert für die Bepreisung der durch den SQL-Ausdruck erzeugten Tupel definiert werden. Nachdem die Beschreibung und der SQL-String des komplexen Preises erfasst wurden, muss ein Attribut (bzw. eine Attributkombination) zur Identifikation der Kunden angegeben werden. Parallel dazu können den übrigen Attributen Dimensionen zugeordnet werden. Als nächsten Schritt muss der Anwender des Price-Modelers die Dimensionen



skalieren und den Skalenniveauekombinationen Preiszähler zuordnen. In einem letzten Schritt kann eine AccLan-Anweisung auf die Tupel definiert werden.<sup>171</sup>

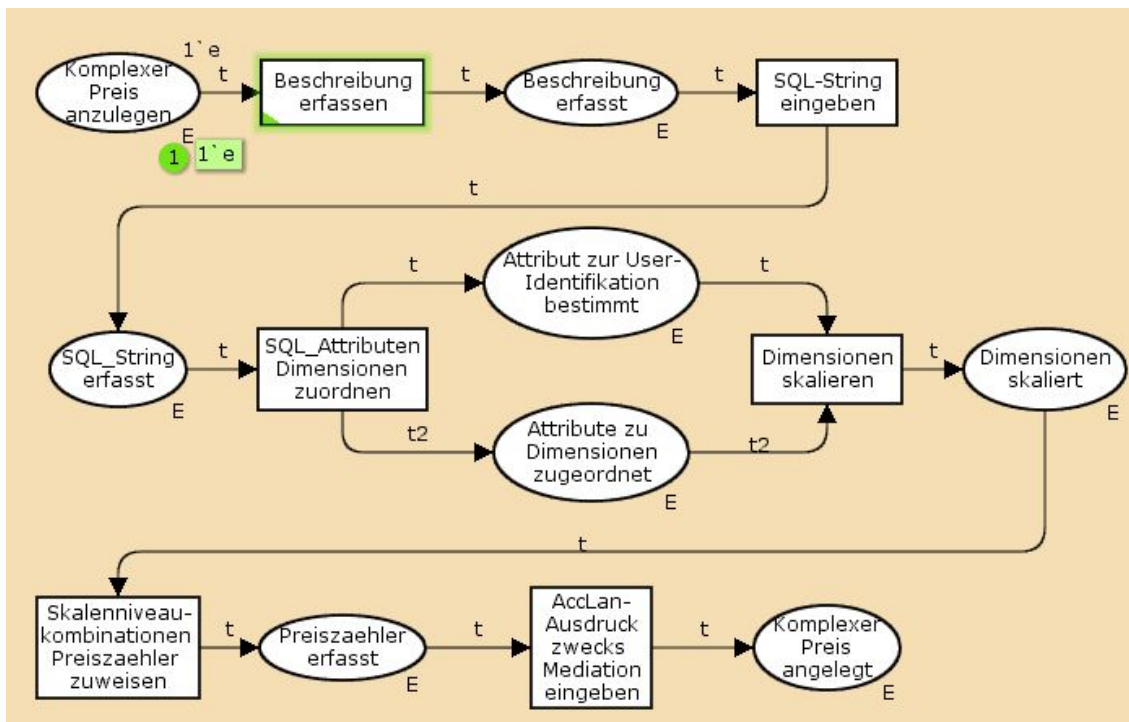



Abbildung 49: Komplexen Preis anlegen als CP-Netz

### 6.6.3.6 Beispiel: Umsetzung eines CP-Netzes in einen Wizard

Das Anlegen eines komplexen Preises kann der Anwender entweder

- durch Rechtsklick im Modell,
- durch Auswahl der Aktion in einem Dropdown-Menü oder
- durch das Anklicken des entsprechenden Icons in der Taskleiste(  ) initiieren.

Nach der Angabe, unter welcher Aggregationsregel der komplexe Preis angelegt werden soll, startet der Wizard, der den Anwender durch die folgenden 6 Schritte führt:

*Schritt 1:* Beschreibung erfassen → Beim Erfassen der Beschreibung legt der Prototyp automatisch einen fortlaufenden Identifier (im Beispiel <CP 0>) für den

<sup>171</sup> Wird kein AccLan-Ausdruck angegeben, so werden sämtliche ermittelten Preiszähler für einen Kunden summiert.

komplexen Preis fest. In Formular in Abbildung 50 wird die Anlage eines komplexen Preises für ein Lernobjekt initiiert.

The screenshot shows a dialog box titled "Complex Price" with a close button in the top right corner. Below the title bar, it says "General Information for Complex Price". There are two input fields: "ID" with the value "<CP 0>" and "Name" with the value "Lernobjekt". Below these is a "Description" field containing the text: "Der Preis fuer die Abfrage eines Lernobjektes, z.B. einer Übung oder einer Lektion in einem E-Learning-Kurs." The dialog box has a blue border and a standard Windows-style title bar.

**Abbildung 50: Komplexen Preis anlegen - Beschreibung angeben**

*Schritt 2:* SQL-String definieren → Nach der Auswahl des Mountpoints (Datenbank) muss ein SQL-String erfasst werden. Im Beispiel aus Abbildung 51 wird im SQL-String eine Abfrage definiert, die Komplexität, Größe und den Provider der einem Lernenden bereitgestellten Lernobjekte erhebt. Daneben wird durch die Abfrage die `user_id` des Nachfragers eines Lernobjektes festgestellt. Durch Drücken des Buttons „Validate“ wird überprüft, ob die SQL-Abfrage ein zulässiges Ergebnis liefert.<sup>172</sup>

The screenshot shows the same "Complex Price" dialog box, but now it prompts the user to enter a valid SQL statement. The text reads: "Please insert a valid SQL-Statement to determine the Attributes which will be mapped to Dimensions in later steps. Also don't forget to select an Attribute which indicates a Customer's Identification in the productive data." Below this is a "Database" dropdown menu set to "mysql". The "SQL-Statement for Event" field contains the following SQL query:
 

```
SELECT edi.complexity, edi.size,
       eud.user_id,
       edi.provider,
       ed.name
FROM
  elearning_user_downloads as eud,
  elearning_downloads as ed,
```

 At the bottom of the dialog box, there are two buttons: "Validate" and "Preview". The dialog box has a blue border and a standard Windows-style title bar.

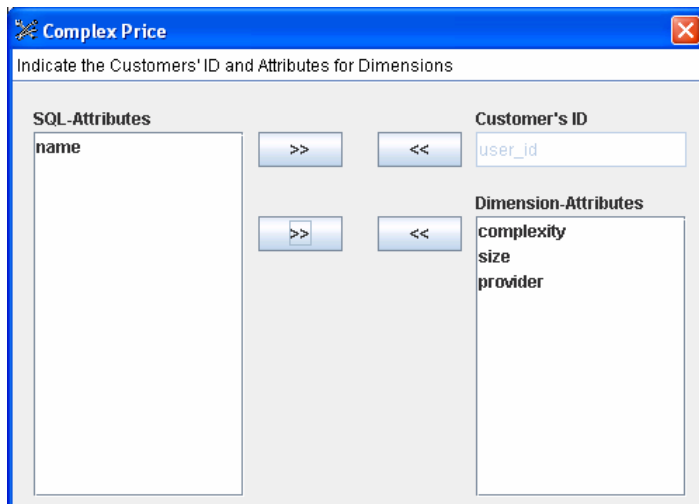
**Abbildung 51: Komplexen Preis anlegen - SQL-String definieren**

<sup>172</sup> Das Ergebnis kann in einer Tabelle ausgegeben werden.

*Beispiel für einen SQL-String*

```
SELECT      edi.complexity, edi.size, eud.user_id, edi.provider,
           ed.name
FROM        elearning_user_downloads as eud, elearning_downloads
           as ed, elearning_downloads_information as edi
WHERE       eud.download_id = ed.id AND edi.download_id = ed.id
```

*Schritt 3:* Attribut zur User-Identifikation und Attribute für Dimensionen festlegen  
 → Die im SQL-String enthaltenen Attribute müssen Dimensionen zugeordnet werden. Damit eine individuelle Abrechnung möglich ist, muss auch ein den User des SPS identifizierendes Attribut festgelegt werden (*user\_id*).



**Abbildung 52: Komplexen Preis anlegen - Kunden identifizieren und Dimensionen anlegen**

*Schritt 4:* Dimensionen skalieren → Nachdem durch das Mapping der SQL-Attribute die Dimensionen bekannt sind, müssen diese skaliert werden. Durch die Skalierung wird ein n-dimensionales Array aufgespannt, in das die Preiszähler eingetragen werden können. In Abbildung 53 wird die Skalierung der Dimension „Complexity“ gezeigt. Die Skalenniveaus müssen überlappungsfrei für jede einzelne Dimension definiert werden. Da durch das nachträgliche Hinzufügen von Dimensionen eine eindeutige Zuordnung der Preiszähler aus dem folgenden Schritt nicht mehr möglich ist und daher alle Preiszähler im Modell gelöscht würden, muss der Anwender des Price-Modelers durch Aushaken von „Lock Dimensions and Scale Niveaus“ bestätigen, dass er die Dimensionen nicht mehr verändern möchte. Erst dann kann der nachfolgende Schritt 5 im Wizard aufgerufen werden.

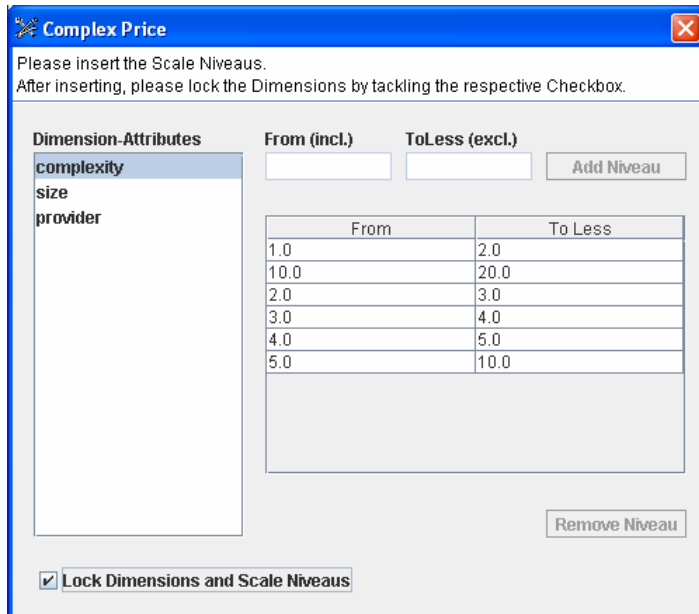


Abbildung 53: Komplexen Preis anlegen - Dimensionen skalieren

Schritt 5: Preiszähler erfassen → Im vorletzten Schritt werden die Preiszähler erfasst. Dazu werden zu den Skalenniveauekombinationen der Dimensionen Werte eingegeben.

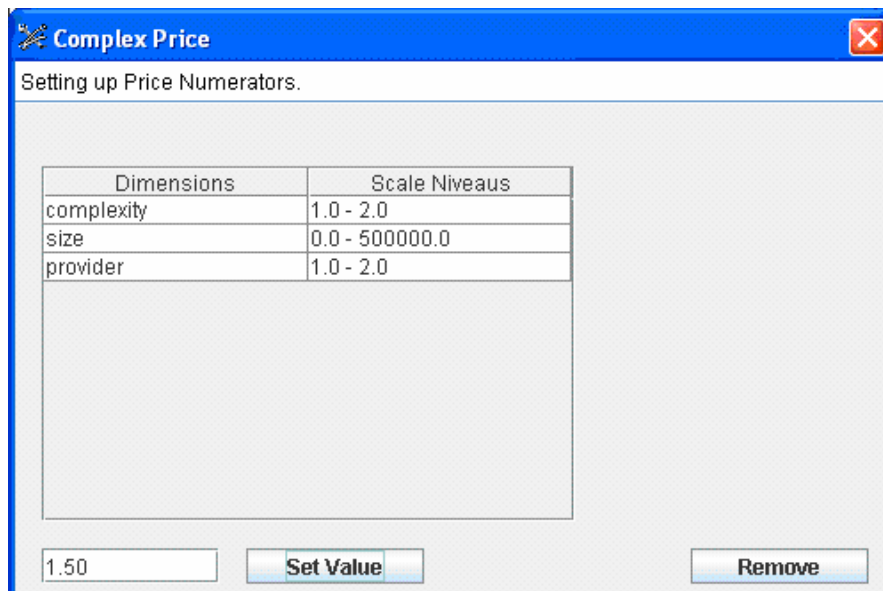


Abbildung 54: Komplexen Preis anlegen - Preiszähler erfassen

Schritt 6: AccLan-Ausdruck → Als letztes wird der AccLan-Ausdruck angegeben, der zur Verdichtung der für einen User ermittelten Preiszähler angewendet wird. Falls kein Ausdruck angegeben wird, werden die Preiszähler aufsummiert. Im Beispiel aus Abbildung 55 wird ab einem Download-Volumen von über 50

Lernobjekten ein Rabatt von 5 %, bei über 60 Lernobjekten 10 % und bei mehr als 70 Lernobjekten von 15 % gewährt.

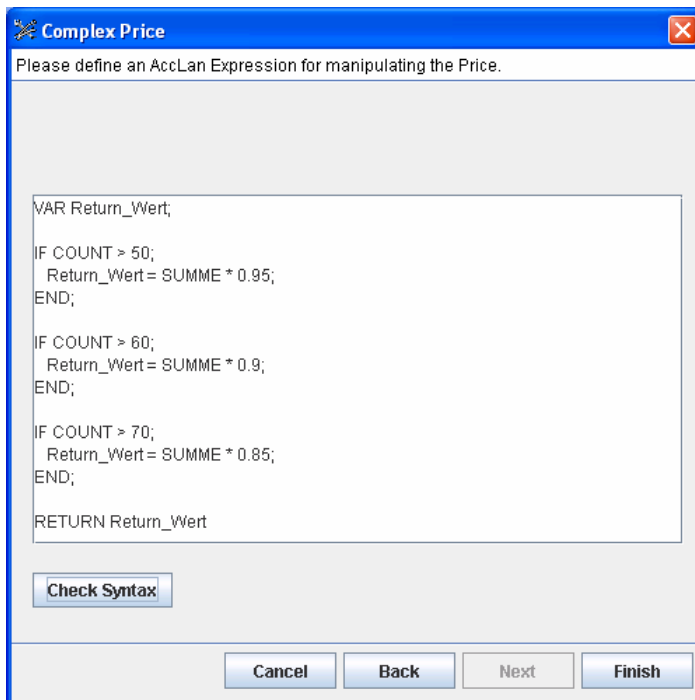


Abbildung 55: Komplexen Preis anlegen - AccLan-Ausdruck definieren

### 6.6.3.7 Einfachen Preis anlegen

Einfache Preis können entsprechend Abbildung 56 für die Bepreisung von Events und für eine zeitbasierte Bepreisung angelegt werden. Für die beiden Varianten einfacher Preise werden in den folgenden Abschnitten Petri-Netze definiert.

### 6.6.3.8 Eventbasierten Preis anlegen

Eventbasierte Preise werden gemäß Abbildung 57 ähnlich den komplexen Preisen definiert. Da nur die Identifizierung des Kunden über eine User-ID wichtig ist, fehlt jedoch die Zuordnung von Attributen zu Dimensionen etc. Stattdessen wird ein einheitlicher Preiszähler festgelegt, der für jedes *Event*<sup>173</sup> verrechnet werden soll. Auch bei eventbasierten Preisen ist eine Angabe eines AccLan-Ausdrucks möglich.

<sup>173</sup> Ein Event schlägt sich in der Datenbank als Tupel nieder.

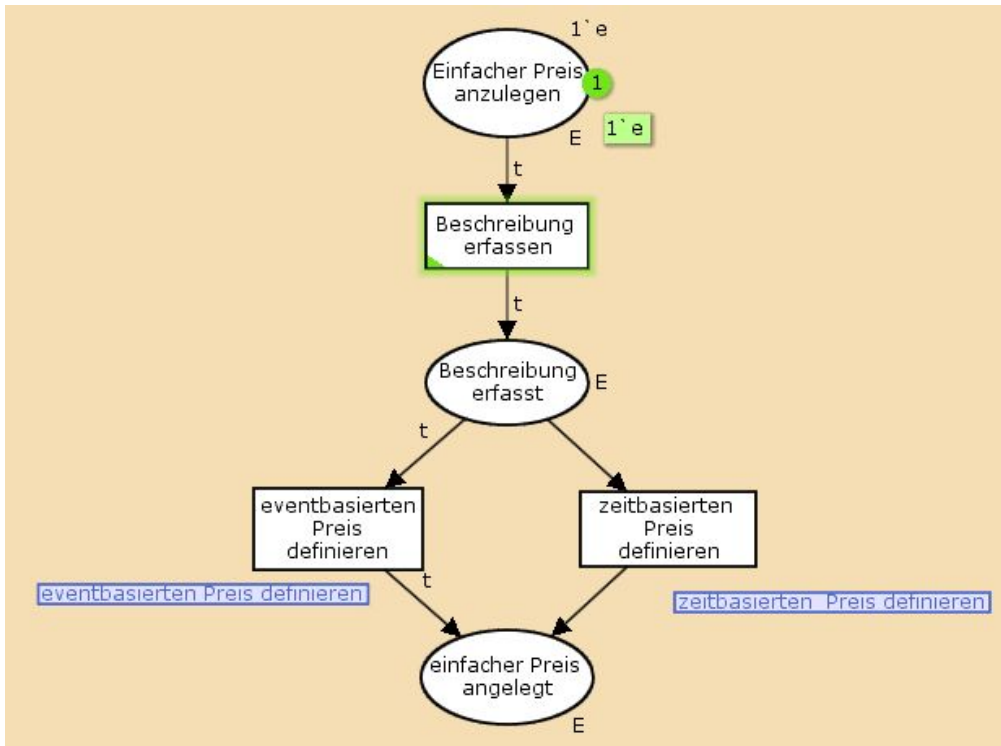


Abbildung 56: Einfachen Preis anlegen als CP-Netz

Das Anlegen eines eventbasierten Preises zerfällt in die folgenden Schritte:

*Schritt 1:* Event in SQL definieren → Die bei der Auswertung eines SQL-Strings zurückgegebenen Tupel definieren die „Events“, die einem User zuzuordnen sind.

*Schritt 2:* Definition des User-Attributs → Um ein Tupel einem User zuzuordnen, muss ein identifizierendes Attribut festgelegt werden.

*Schritt 3:* Preiszähler festlegen → Der Preiszähler definiert, welcher Preis für einen Event abzurechnen ist.

*Schritt 4:* AccLan-Ausdruck → Durch einen AccLan-Ausdruck kann schließlich die Ergebnisberechnung wie im Falle eines komplexen Preises variiert werden.

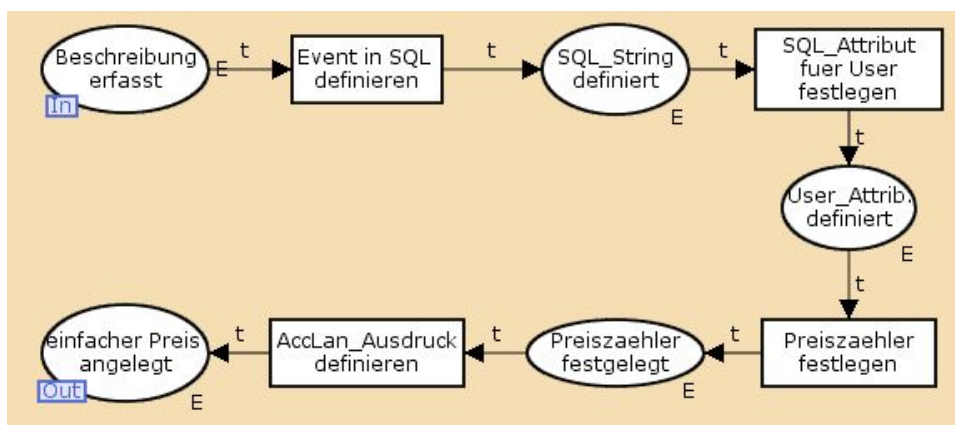


Abbildung 57: Eventbasierten Preis anlegen als CP-Netz

### 6.6.3.9 Zeitbasierten Preis anlegen

In einem zeitbasierten Preis kann entweder ein Zykluspreis für die zyklisch wiederkehrende Berechnung von Preisen oder eine One-Time-Charge für die einmalige Preisberechnung definiert werden. Bei der Definition eines Zykluspreises müssen ein Startzeitpunkt und die Zyklusdauer erfasst werden (vgl. Abbildung 58).

Für das Anlegen eines *Zykluspreises* sind die folgenden Schritte erforderlich:

*Schritt 1:* Preiszähler setzen → Wie zuvor.

*Schritt 2:* Anfangszeitpunkt definieren → Festlegen des ersten Abrechnungsdatums für den Preis.

*Schritt 3:* Zyklusdauer definieren → Festlegen der Zeitspanne, nach deren Ablauf der Preis erneut abgerechnet wird. Fakultativ: Anlegen eines Endzeitpunktes.

*Schritt 4:* AccLan-Ausdruck setzen → Analog wie bei eventbasierten Preisen, jedoch findet eine rein zeitabhängige Berechnung eines Preiszählers statt.

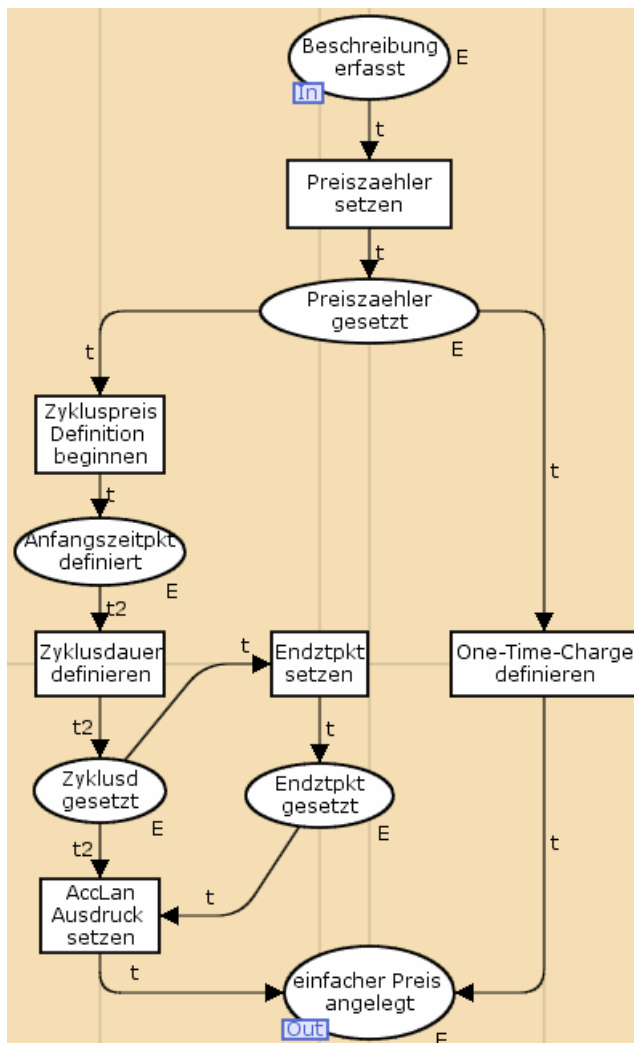


Abbildung 58: Zeitbasierten Preis anlegen als CP-Netz

Für das Anlegen einer *One-Time-Charge* ist lediglich die Definition des Preiszählers erforderlich.

### 6.6.3.10 Aggregationsregel anlegen

Zum Anlegen einer Aggregationsregel sind zwei Schritte erforderlich (vgl. Abbildung 59). In einem ersten Schritt wird dazu die Beschreibung der Aggregationsregel (Name und Text) definiert, so dass die Aggregationsregel bereits in einem grafischen Modell abgebildet werden kann. Da Aggregationsregeln die Preiszähler der untergeordneten Elemente verdichten, kann nach dem Anlegen der untergeordneten Elemente, d.h. weiterer Aggregationsregeln, komplexer und einfacher Preise – ausgedrückt durch die Stelle „*untergeordnete Elemente liegen vor*“ – ein AccLan-Ausdruck auf diese in der Aggregationsregel definiert werden.

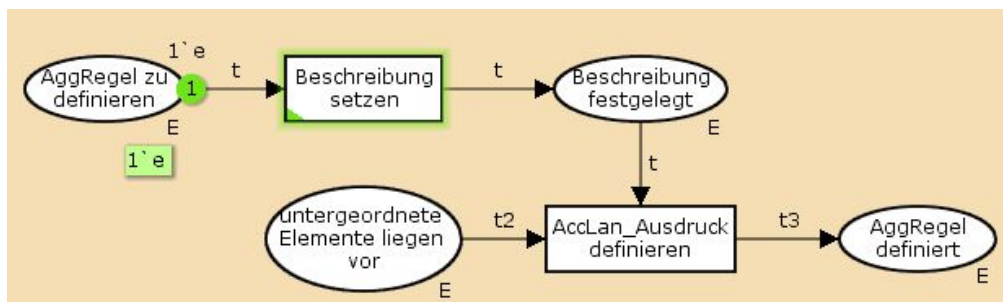


Abbildung 59: Aggregationsregel anlegen als CP-Netz

Die zuvor beschriebenen CP-Netze zeichnen sich dadurch aus, dass lediglich Nulltupel als Markierung zugelassen sind. Sie können jedoch ohne Einschränkung als Vorlage z. B. für die Definition von Wizards<sup>174</sup> verwendet werden. Eine Präzisierung durch Hinzufügen von Colour-Sets ist möglich. In Abbildung 60 ist ein beispielhaftes CP-Netz dargestellt, das das Einlesen einer Beschreibung und eines Anfangszeitpunktes über Textdateien erlaubt.

<sup>174</sup> Ein Wizard kapselt eine Folge von Formularen in einem Programm, die einen Benutzer dabei unterstützen, Parameter z. B. für einen einfachen Preis festzulegen.



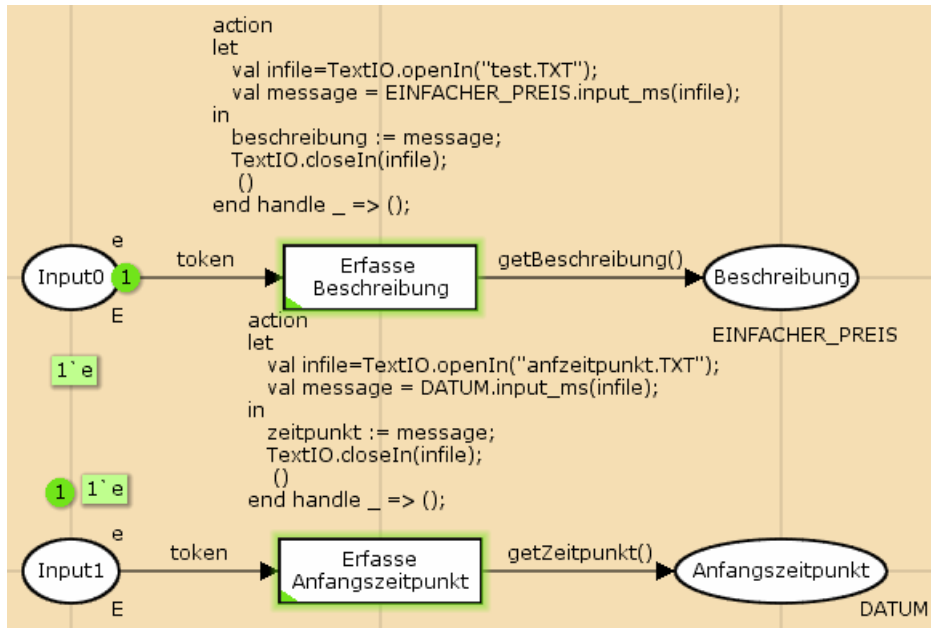
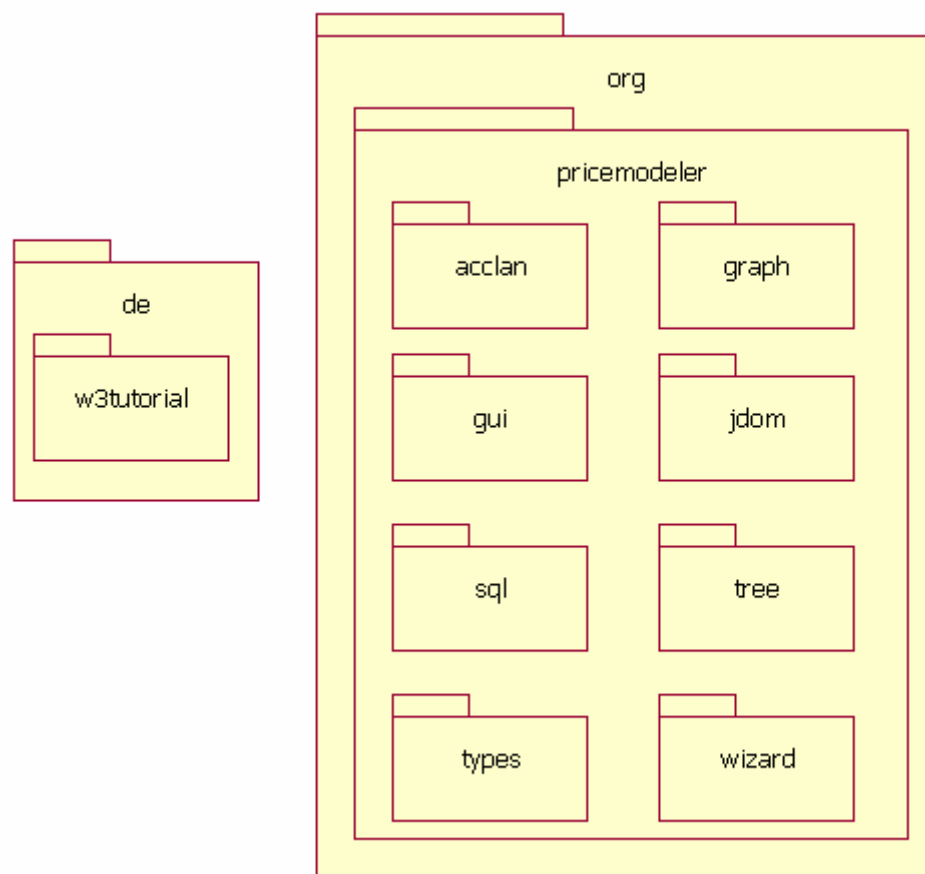


Abbildung 60: Präzisierung durch Definition von Colour-Sets für Stellen

#### 6.6.4 Paketstruktur des Price-Modeler-Prototyps

Die wichtigsten Pakete der Paketstruktur des Price-Modelers sind in der folgenden Abbildung dargestellt. Die Pakete enthalten Funktionen zu den in Kapitel 6.6.2 beschriebenen Ebenen der Architektur. Im Paket „graph“ werden Java-Klassen zum Rendering der Modelle in Graphen vorgehalten. Graphen als Teil eines GUIs werden durch Klassen im „gui“-Paket verwaltet. Die interne Struktur der Preismodelle wird durch „jdom“ verarbeitet, das wiederum auf das Paket „tree“ zurückgreift. Das Paket „sql“ enthält Funktionalität zur Ansteuerung von Datenbanken und zur Definition der SQL-Meter, das „wizard“-Paket entsprechend für die Erzeugung der Wizards zum Anlegen von Modellkomponenten. Neben der Anwendungslogik, die im Paket „org.pricemodeller“ verpackt ist, werden im Paket `w3tutorial` Hilfsklassen abgelegt, z. B. zum Sortieren und zum Speichern der Modelle.



*Abbildung 61: Java-Paketstruktur des Price-Modeler-Prototyps*

## 7 Ein Lern-Managementsystem mit integriertem Pricing

In den vorherigen Kapiteln wurde eine Beschreibungssprache eingeführt, die zum Pricing digitaler Produkte (i. w. S.) verwendet werden kann. Zudem wurde ein Software-Prototyp erstellt, der es einem Anwender ermöglicht, Preismodelle zu erzeugen. Das Ziel dieses Kapitels ist es, anhand eines Beispiels die Verwendung der Beschreibungssprache in der Domäne „E-Learning“ zu demonstrieren.<sup>175</sup>

### 7.1 Ein prototypisches Lern-Managementsystem

In diesem Abschnitt wird ein prototypisches Lern-Managementsystem (LMS) beschrieben. Das wesentliche Ziel der Entwicklung des LMS liegt in der Überprüfung der Pricing-Konzepte hinsichtlich ihrer Anwendbarkeit sowie der Implementierbarkeit der zuvor beschriebenen Modellierungssprache im Sinne eines „Proof-of-Concept“. Die Funktionalität des LMS ist daher auf das Wesentliche beschränkt und nicht mit professionellen Systemen wie z. B. WebCT/Blackboard ([www.webct.com](http://www.webct.com))<sup>176</sup> und CLIX ([www.clix.de](http://www.clix.de)) zu vergleichen.<sup>177</sup> Das Kapitel endet mit der Darstellung einer Architektur zur Integration von E-Learning-Anwendungen in einem Web-Portal.

#### 7.1.1 Entwurf eines prototypischen Lern-Managementsystems

Neben dem Zugriff auf Lernressourcen und dem Anlegen von Nutzern soll das prototypische LMS die Abrechnung der Dienstinanspruchnahme durch die Lernenden auf Basis der Daten des LMS unterstützen. Mit Hilfe des Price-Modeler-Prototyps werden Preismodelle erstellt und in XML-Dokumenten abgelegt (vgl. Abbildung 62). Das LMS umfasst eine PHP-Laufzeitumgebung für Preismodelle, die die Bepreisung des E-Learnings anhand der von einem Modellierer bereitgestellten Modelle erlaubt. Darüber hinaus ermöglicht sie den

---

<sup>175</sup> E-Learning wurde detailliert in Kapitel 0 beschrieben.

<sup>176</sup> WebCt und Blackboard wurden in XI/2005 zusammengeführt.

<sup>177</sup> Auch professionelle Lern-Managementsysteme weisen Nachteile auf: (1) Durch ihre monolithische Struktur ist eine Integration in die Anwendungslandschaft von Bildungsprovidern aufwändig (Grüne u.a. 2003). (2) Zudem handelt es sich bei diesen Systemen im Wesentlichen um Content-Management-Systeme, denen es an einer Prozessunterstützung für Lernende und Lehrende mangelt. In Kapitel 7.2 wird eine Architektur für ein Portal-basiertes Lern-Managementsystem skizziert und somit ein Vorschlag für zukünftige Entwicklungen von Lern-Managementsystemen unterbreitet.

Kunden / Lernenden und Administratoren den Zugriff auf die für sie zulässigen Preismodelle und den Austausch von Modellen im laufenden Betrieb. Durch den Austausch der Modelle ändern sich die Abrechnungsmodalitäten für einen Kunden on-the-fly.<sup>178</sup>

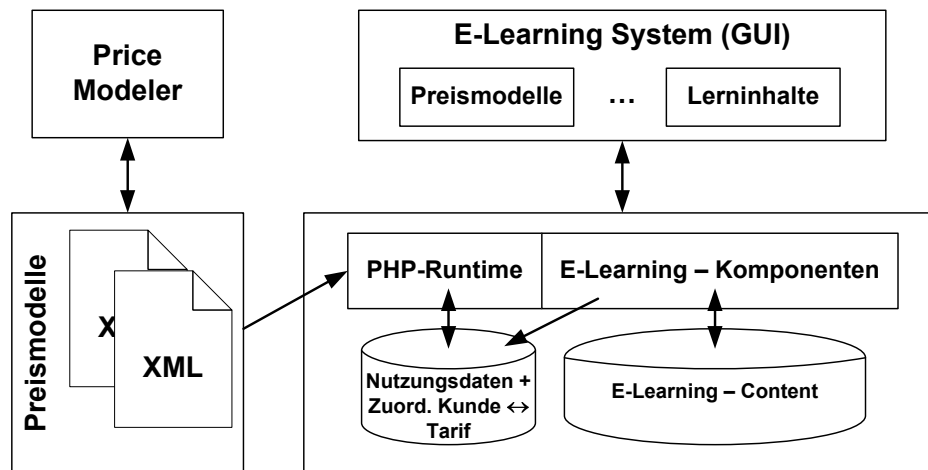


Abbildung 62: Zusammenspiel des Price-Modelers mit einem LMS über eine Laufzeitumgebung

### 7.1.2 Softwaretechnische Entwicklungsarchitektur des LMS

Das LMS wurde in PHP 5 ([www.php.net](http://www.php.net)) entwickelt. PHP ist eine serverseitige Skriptsprache, die zur dynamischen Erstellung von HTML-Webseiten verwendet wird. PHP-Skripte können mit Datenbanken kommunizieren. Derzeit werden unter anderem Oracle, DB2, PostgreSQL, MySQL und ODBC-konforme Datenbanken unterstützt. Darüber hinaus bietet PHP die für die Umsetzung des LMS erforderlichen Funktionen zur Authentifizierung von Nutzern, zum Parsing von XML, das für die Darstellung der Preismodelle benötigt wird<sup>179</sup>, und zur Erzeugung von grafischen Repräsentationen der Preismodelle (GD-Bibliothek).

Zur Implementierung des LMS muss folgende Software installiert sein:

- MySQL 5.0.20 ([www.mysql.com](http://www.mysql.com))
- PHP 5.1.1 ([www.php.net](http://www.php.net))
- Apache HTTPD 2.2.0 ([httpd.apache.org](http://httpd.apache.org))

<sup>178</sup> In einem professionellen Service-Provisioning-System sollte ein Preismodell nach Auswahl für eine vordefinierte Preisspanne, z. B. einen Monat, als nicht austauschbar gekennzeichnet werden.

<sup>179</sup> Die verwendeten DOM XML-Funktionen sind experimentell.

Für Probeinstallationen bietet sich die Installation von XAMPP (www.apachefriends.org) an, das die zuvor aufgeführte Software umfasst und nur minimale Anforderungen an die Konfiguration stellt.

### 7.1.3 Anwendungsfälle

Die Anwendungsfälle des LMS beschreiben sowohl die Verwaltung von Inhalten und Preismodellen, die außerhalb des Systems erstellt wurden, durch Administratoren, als auch die Möglichkeiten zur Nutzung des LMS durch einen Lernenden / Anwender. In den folgenden Kapiteln 7.1.4 und 7.1.5 werden Anwendungsfälle aus Sicht der Administratoren und der Benutzer des LMS dargestellt.

#### 7.1.4 Anwendungsfälle aus Administrator-Sicht

Hierzu gehören neben dem Anlegen neuer Benutzer die Zuordnung von Preismodellen zu Kursen, die Definition eines Kurskontextes (Kurs anlegen), das Hinzufügen von Lektionen zu einem Kurs (Lektion anlegen) und die Zuweisung von Lernobjekten zu Lektionen (Übung / Download anlegen).

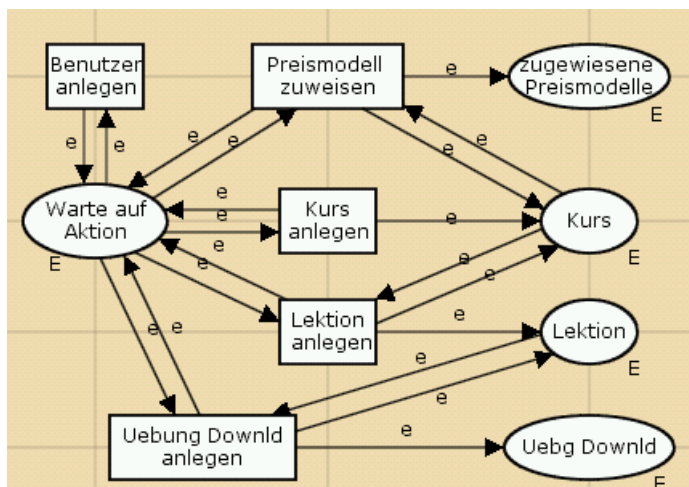


Abbildung 63: Anwendungsfälle aus der Sicht eines Administrators

##### 7.1.4.1 Anlegen eines Benutzers

Das Anlegen eines Benutzers wird durch Eingabe der erforderlichen Daten im Formular, das in Abbildung 64 dargestellt ist, unterstützt. Durch Erweiterung des Formulars und des Datenmodells können weitere Informationen erfasst werden. Denkbar wäre die Eingabe der Kundengruppe oder die Erfassung des Status eines

Lernenden (Anzahl der Fachsemester / Studienrichtung oder die Angabe, ob das Studium als Zweitstudium bestritten wird).

Benutzer anlegen		Benutzer löschen	
Benutzername:	<input type="text"/>	markus	<a href="#">löschen</a>
Kennwort:	<input type="text"/>		
Vorname:	<input type="text"/>		
Nachname:	<input type="text"/>		
<input type="button" value="Absenden"/>			

Abbildung 64: LMS - GUI zur Anlage eines Benutzers

### 7.1.4.2 Anlegen eines Kurses

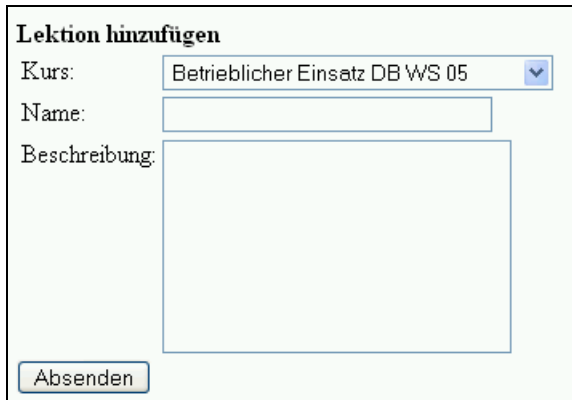
Das Formular zum Anlegen eines Kurses ist in Abbildung 65 dargestellt. Im prototypischen LMS stellen Kurse und Kursteile wie Lektionen und Übungen die Produkte dar. Erst nach dem Anlegen eines Kurses können dem Kurs Lektionen zugeordnet werden.

Kurs anlegen	
Name:	<input type="text"/>
Dozent:	<input type="text"/>
Startdatum:	<input type="text"/> Format: 2000-12-31 16:01:45
Enddatum:	<input type="text"/> Format: 2000-12-31 16:01:45
Syllabus:	<input type="text"/>
Kreditpunkte:	<input type="text"/>
<input type="button" value="Absenden"/>	

Abbildung 65: Anlegen eines Kurses

### 7.1.4.3 Lektion hinzufügen

Das Hinzufügen einer Lektion zu einem Kurs ist durch die Eingabe in dem folgenden Formular möglich.



**Lektion hinzufügen**

Kurs:

Name:

Beschreibung:

*Abbildung 66: Hinzufügen einer Lektion zu einem Kurs*

Eine Lektion kann eine beliebig umfangreiche Beschreibung enthalten. Die Erstellung von Inhaltsseiten zu einem Kurs wird durch ein weiteres Formular unterstützt (nicht dargestellt).<sup>180</sup>

#### **7.1.4.4 Preismodell zuweisen**

Die mit dem Price-Modeler erzeugten Preismodelle können den Kursen, die das zentrale Produkt im LMS bilden, zugewiesen werden (vgl. Abbildung 67). Nach der Zuweisung eines Preismodells zu einem Kurs, können Lernende dieses auswählen. Im Formular werden die zur Verfügung stehenden Modelle in einer Dropdown-Liste angezeigt. Eine Zuweisung mehrerer Preismodelle zu einem Kurs ist möglich. Zum Kurs „Betrieblicher Einsatz DB WS 05“ wurden neben einem Flat-Fee-Modell ein Preismodell für die verbrauchsabhängige Abrechnung und ein nach Kundengruppen differenziertes Preismodell eingerichtet.

---

<sup>180</sup> Nach dem Buchen eines Kurses stehen die Inhaltsseiten dem Lernenden zur Verfügung. Eine Einzelbuchung von Kursinhaltsseiten ist im Prototyp-System nicht möglich, könnte jedoch einfach nachgerüstet werden.

**Preismodell zuweisen**

Kurs:

Preismodell:

als

**Preismodell löschen**

Flat-Fee-Modell (FixedFee.prm1) [löschen](#)

Verbrauchsabhängiger Tarif (Usage.prm1) [löschen](#)

Tarif nach [löschen](#)

Kundengruppen (PreisdifferenziertesModell.prm1) [löschen](#)

Downloads nach Kundengruppe, Fixpreis pro Übung (PreisdifferenziertesModell.prm1) [löschen](#)

---

**Kurse / Modelle**

Kurs	Preismodell
Betrieblicher Einsatz DB WS 05	Flat-Fee-Modell (FixedFee.prm1) <a href="#">entfernen</a>
	Verbrauchsabhängiger Tarif (Usage.prm1) <a href="#">entfernen</a>
	Tarif nach Kundengruppen (PreisdifferenziertesModell.prm1) <a href="#">entfernen</a>
Grundzüge der Wirtschaftsinformatik	Flat-Fee-Modell (FixedFee.prm1) <a href="#">entfernen</a>
	Verbrauchsabhängiger Tarif (Usage.prm1) <a href="#">entfernen</a>
	Downloads nach Kundengruppe, Fixpreis pro Übung (PreisdifferenziertesModell.prm1) <a href="#">entfernen</a>

**Abbildung 67: LMS - GUI zum Zuweisen eines Preismodells zu einem Kurs sowie zum Löschen von Preismodellen**

### 7.1.4.5 Anlegen von Übungen und Downloads

Nachdem die Lektionen zu einem Kurs angelegt wurden, können den Lektionen Übungen und Downloads zugewiesen werden. Dazu müssen die Downloads und Übungen in vordefinierten Verzeichnissen des LMS abgelegt werden, z. B. in Form von PPT- oder PDF- Dateien. Das in Abbildung 68 dargestellte Formular unterstützt die Zuordnung der Dateien zu einer Lektion eines Kurses. Eine Zuordnung zu mehreren Kursen ist möglich. Im Fall der Verwendung eines Lernobjekts in mehreren Kursen, sollte dies nur einmal abgerechnet werden.

**Kurse-Lektionen-Übungen-Downloads**

Kurse	Lektionen
	<p><b>Übungen:</b></p> <p>Datei <input type="text" value="AchteUebung.ppt"/> mit Name <input type="text"/></p> <p><input type="button" value="Hinzufügen"/></p> <p>Übungsblatt 1 <a href="#">entfernen</a></p> <p>Übung 1 Lösung <a href="#">entfernen</a></p> <p>Wiederholung 1 <a href="#">entfernen</a></p> <p><b>Downloads:</b></p> <p>Datei <input type="text" value="Win1_Praktisch3_Lösung.pdf"/> mit Name <input type="text"/></p> <p><input type="button" value="Hinzufügen"/></p> <p>Termine der Vorlesung <a href="#">entfernen</a></p> <p>Termine der Übung <a href="#">entfernen</a></p> <p>Kapitel 1 als PDF <a href="#">entfernen</a></p> <p>Klausur der vorherigen Vorlesung (2004) <a href="#">entfernen</a></p> <p><b>Übungen:</b></p>
	Einführung <a href="#">entfernen</a>

**Abbildung 68: Hinzufügen von Übungen und Downloads zu einer Lektion**



### 7.1.5 Anwendungsfälle aus Sicht der Lernenden

Für Lernende wurden die folgenden Anwendungsfälle im LMS implementiert: Abonnieren eines Kurses, Buchung eines Lernobjektes (Übung, Download) und die Auswahl eines Preismodells für einen Kurs.

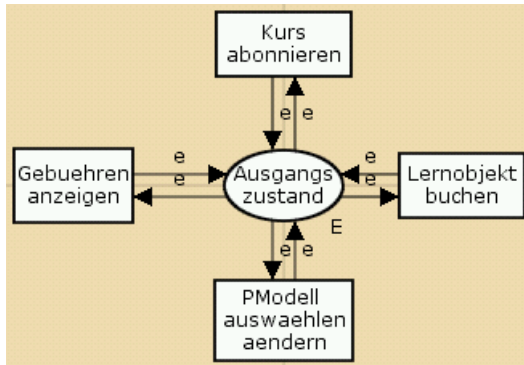


Abbildung 69: Anwendungsfälle aus Sicht eines Lernenden

#### 7.1.5.1 Buchung eines Kurses und Auswahl eines Preismodells

Der Lernende kann, nachdem ihm durch einen Administrator ein Benutzerkonto angelegt wurde und er sich im LMS angemeldet hat, einen Kurs buchen. Dazu wählt er im Übersichtsmenü einen Kurs aus und klickt in der Übersichtsseite zum Kurs auf „Diesen Kurs abonnieren“ (vgl. Abbildung 70). Nach dem „Abonnieren“ des Kurses, muss der Anwender ein Preismodell (im Prototypen als „Tariff Model“ bezeichnet) auswählen. Die Auswahl von Preismodellen zu einem bestehenden Kurs „Grundzüge der Wirtschaftsinformatik“ ist in Abbildung 71 dargestellt.

**E-Learning Portal - Grundzüge der Wirtschaftsinformatik**

user: martin | profile | logout

**Grundzüge der Wirtschaftsinformatik**

[Diesen Kurs abonnieren.](#)

Dozent: Prof. Dr. Roland Holten  
Kreditpunkte: 4

Der Kurs beginnt am 24.10.2004 und endet am 03.02.2005.

**Syllabus**  
Vermittelt werden soll eine Kombination aus Methoden- und Faktenwissen.  
Einbeziehung von akt. Tagesgeschehen.  
Lernziel: kritisches Hinterfragen von Modeerscheinungen.  
Erkennen von lfr. Trends.

**Lektionen:**

- [1 - Einleitung](#)
- [2 - Grundlagen HW und SW](#)
- [3 - Rechnernetze](#)
- [4 - Daten-, Funktions-, und Ablaufmodellierung](#)
- [5 - DB- und IS](#)
- [6 - Software Engineering Management](#)
- [7 - Programmierung](#)
- [8 - IM und betriebliche IS](#)
- [9 - E-Commerce und M-Commerce](#)
- [10 - Safety und Security](#)
- [11 - Rechtliche Problemstellungen](#)
- [12 - Langfristige Entwicklungen](#)
- [13 - WI Weiterentwicklung](#)

Fertig

Abbildung 70: Kurs buchen

**Tarif-Modell:**

Bitte wählen Sie ein Preismodell zu diesem Kurs.

- [Flat-Fee-Modell](#)
- [Verbrauchsabhängiger Tarif](#)
- [Downloads nach Kundengruppe](#)
- [Fixpreis pro Übung](#)

Abbildung 71: Preismodell auswählen für einen Kurs

### 7.1.5.2 Aktivierung der Nutzung von Lernobjekten

Im LMS muss ein Lernender explizit angeben, welche Lernobjekte (Übungen und Downloads) er nutzen möchte. Dazu muss der Lernende hinter dem gewünschten Lernobjekt auf „bereitstellen“ klicken (vgl. Abbildung 72 a)). Danach kann das Lernobjekt uneingeschränkt genutzt werden (vgl. Abbildung 72 b)).<sup>181</sup>



Abbildung 72: Buchung von Lernobjekten

### 7.1.6 Eine Laufzeitumgebung für Preismodelle

Die Laufzeitumgebung interpretiert Preismodelle, indem sie diese aus einem vorgegebenen Verzeichnis ausliest und die in den Preismodellen definierten Regeln auswertet. Der Algorithmus der Laufzeitumgebung ist in Abbildung 73 in Auszügen wiedergegeben. Wie in Kapitel 5.4.2 dargestellt werden Preismodelle von unten nach oben ausgewertet. Dazu müssen in einem ersten Schritt die Blattknoten, d.h. einfache und komplexe Preise, ausgewertet werden. Die Auswertung eines Blattknotens wird im oberen Block der Abbildung beschrieben. Durch die in den Preisen definierten SQL-Strings werden Tupel aus der

<sup>181</sup> Die Bereitstellung muss im LMS unabhängig von dem zugrunde liegenden Preismodell durchgeführt werden. Erst durch die „Bereitstellung“ kann eine verbrauchsabhängige Bepreisung für Kurse eingerichtet werden. Aus Gründen der Vereinfachung muss der Lernende auch im Falle einer Flat-Rate Inhalte „bereitstellen“, obwohl nur die in der Flat-Rate angegebene Grundgebühr verrechnet wird.

Datenbank des LMS ausgelesen.<sup>182</sup> Diese werden mit Preiszählern bewertet und anschließend durch einen AccLan-Ausdruck verdichtet. Eine Zuordnung zu einzelnen Lernenden erfolgt durch die Definition der User-Attribute in den SQL-Strings.<sup>183</sup> Anschließend werden Aggregationsregeln rekursiv ausgewertet. Existiert zu einer Aggregationsregel keine weitere, übergeordnete Regel, so ist der Endpreis gefunden, da es sich um den Wurzelknoten des Preismodells handelt.

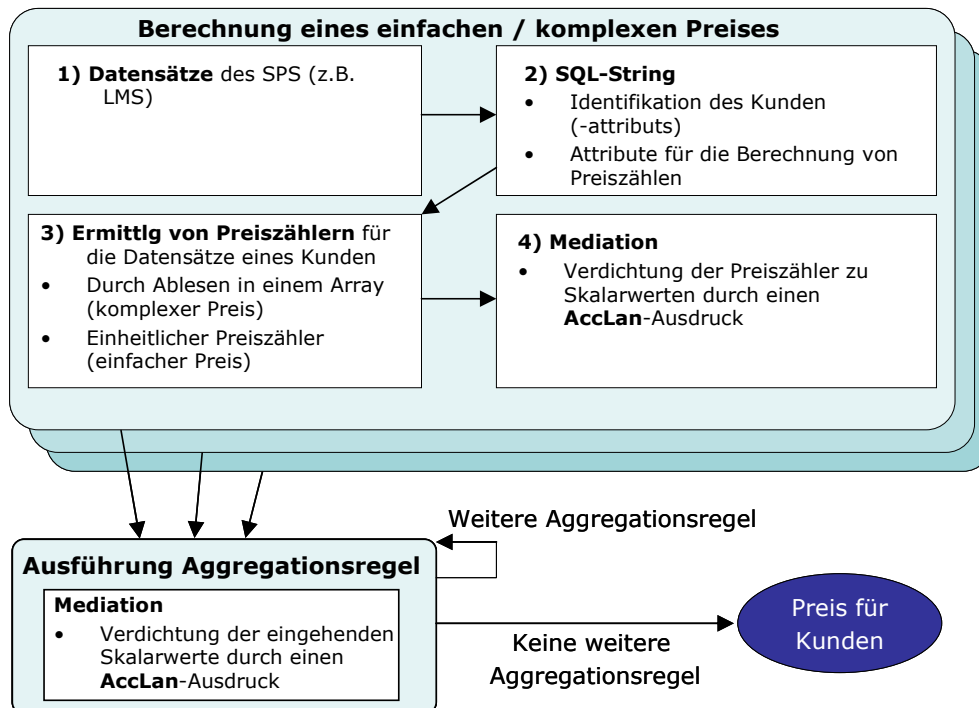


Abbildung 73: Berechnung eines Endpreises mit einem Preismodell

Nach der Wahl eines Preismodells im LMS durch den Lernenden, wird dieses dem Lernenden verkleinert in der Inhaltsseite des Kurses angezeigt. Dabei ähnelt die Darstellung im LMS der Darstellung im Price-Modeler und enthält zusätzlich die für den Lernenden durch die Laufzeitumgebung ermittelten Einzelpreise. Nach Klicken auf das verkleinerte Preismodell wird dieses in einem eigenen Browserfenster angezeigt (vgl. Abbildung 75). Der Lernende erhält so unmittelbar einen Überblick über die zu seinen Verbräuchen entstandenen Gebühren.

<sup>182</sup> Gilt nicht für zeitabhängige Preise und One-Time-Charges.

<sup>183</sup> Bei zeitabhängigen Preisen und One-Time-Charges ist eine solche Zuordnung nicht erforderlich.

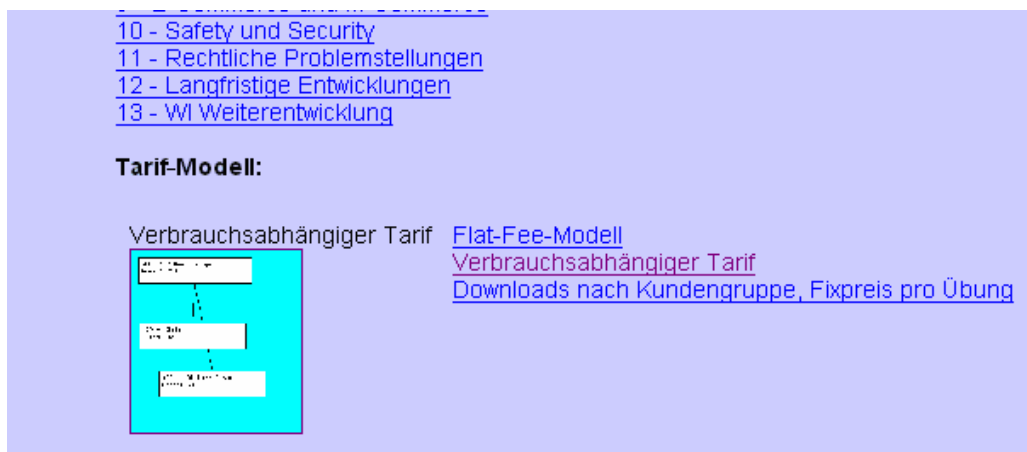


Abbildung 74: verkleinerte Anzeige eines Preismodells im LMS

In der folgenden Grafik ist das Preismodell „Verbrauchsabhängiger Tarif“ für den Lernenden mit der ID „martin“ angegeben. Durch Wahl des Preismodells würde demnach ein Endpreis von € 210,00 zu entrichten sein, der sich durch einen zeitabhängigen (Zeit) und einen Download-Preis (DL User Group) errechnet.<sup>184</sup>

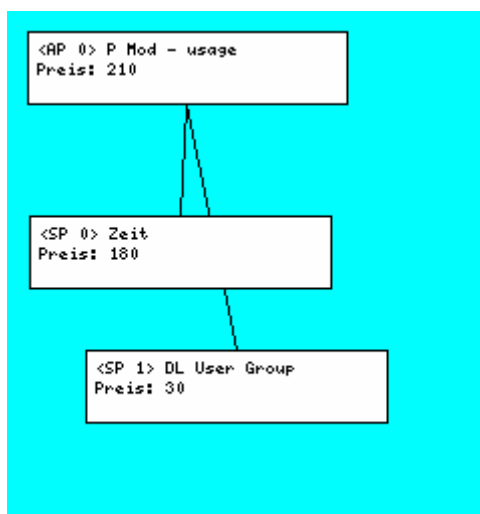


Abbildung 75: Auswertung eines Preismodells für einen Lernenden

## 7.2 Portal-Architektur eines Workflow-gestützten LMS

### 7.2.1 Workflows im E-Learning

Professionelle LMS bieten über die zuvor beschriebenen Funktionen des Prototyps hinaus weitere Möglichkeiten zur Planung und Betreuung von

Lehrveranstaltungen durch die Verwaltung von Kalendern, Funktionen zum Austausch von E-Mails zwischen den Lernenden und Lehrenden und zur Kommunikation von Lernenden untereinander (Foren). Obwohl die Arbeitsabläufe zur Erstellung von Lerninhalten und zur Betreuung von Lernenden oft ein hohes Maß an Kreativität erfordern, kann die Einführung von Workflow-Managementsystemen Lernenden und Lehrenden bei der Durchführung wiederkehrender Aufgaben sowie bei der Einhaltung von Terminen und Fristen Unterstützung bieten.

Eine Sammlung von allgemeinen Prozessen z. B. zur Erstellung von E-Learning-Inhalten, zur Organisation des Betriebs und der Nutzung von E-Learning und zur Evaluation von Lehrveranstaltungen, die als Vorlage zur Definition von Workflows dienen kann, ist in den Prozessmodellen der Standards (ISO/IEC19796-1 2005) und (DIN1032-1 2004) zu finden. Die Anpassung der in den Standard-Dokumenten beschriebenen Prozesse an die Bedürfnisse eines Bildungsproviders ist erforderlich, da die dargestellten Prozesse lediglich abstrakt beschrieben sind.

In den folgenden Petri-Netzen sind idealtypische Lehrprozesse abgebildet. Die systematische Beschreibung der Arbeitsabläufe kann außerdem dabei helfen, bestehende Abläufe zu analysieren und diese zu verbessern. Abbildung 76 zeigt ein einfaches, informelles Petri-Netz, das einen Prozess der Teilnahme an einem E-Learning-Kurs darstellt. Daneben sind Accounting-, Charging- und Billing-Prozesse, die zur Bepreisung und Abrechnung des Ressourcenverbrauchs durch Lernende dienen, abgebildet. Im vorliegenden Lernprozess muss der Lernende nach der Buchung eines Kurses ein Assessment absolvieren. Durch das Assessment werden sein Vorwissen festgestellt und abhängig davon die zu belegenden Kurslektionen ermittelt. Der Lernende muss anschließend unterschiedliche Lektionen absolvieren, die jeweils von einem Test in Form eines zu lösenden Aufgabenblatts beendet werden. Nach dem letzten Test wird ein Zertifikat generiert, das eine detaillierte Aufstellung der Lerninhalte und des Lernerfolgs enthält.

---

<sup>184</sup> Alternative Implementierungen der grafischen Darstellung wären durch Auflistung der berechneten Preiskomponenten ähnlich einer Ordnerstruktur zur Darstellung von Dateien oder durch Text-Generierung möglich.

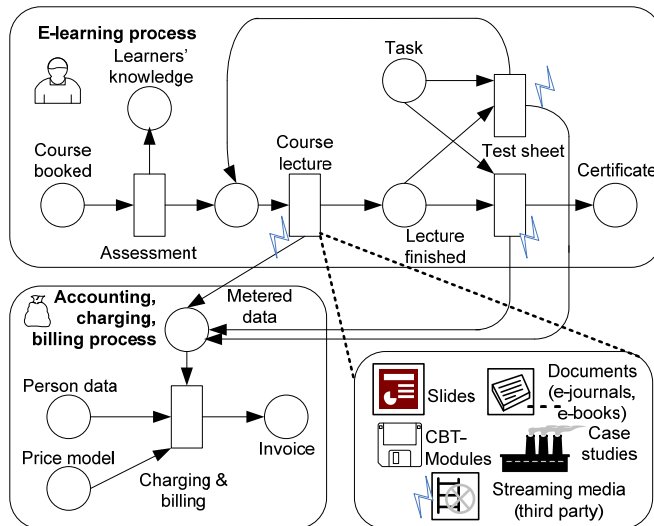


Abbildung 76: E-Learning- und Pricing-Prozesse<sup>185</sup>

Eine Preisberechnung kann wie in Abbildung 76 beschrieben ausgehend von den nachgefragten Objekten (Folien, Modulen etc.) und / oder abhängig von den genutzten Services erfolgen. Dabei kann ein *Dienst / Service* einer einzelnen Aktivität des abgebildeten Prozesses oder auch dem gesamten beschriebenen E-Learning-Prozess entsprechen.<sup>186</sup> Der verkürzt dargestellte Accounting-, Charging- und Billing-Prozess im unteren Teil der Abbildung bewertet die Verbrauchsdaten der Lernenden und generiert abhängig von persönlichen Daten der Lernenden (Gruppenzugehörigkeit u. a.) und der durch die Lernenden gewählten Preismodelle Rechnungen.

Für die Instanziierung der Prozesse in einem Workflow-Managementsystem müssen diese verfeinert werden. In Abbildung 77 ist das CP-Netz zur Korrektur eines Aufgabenblattes mit fünf Aufgaben dargestellt. Dieses bildet den Prozess der Benotung der durch den Lernenden in Abbildung 76 absolvierten Tests ab, der mehrfach in Lehrveranstaltungen an der VGU durchgeführt wurde (s. Kapitel 2.5.6.3, S. 52).<sup>187</sup>

<sup>185</sup> Quelle: (Gruene und Oberweis 2005).

<sup>186</sup> Es bietet sich an, nur solche Dienste zu bepreisen, mit denen der Lernende einen Wert verbindet. Die Beantwortung einer umfangreichen inhaltlichen Frage des Lernenden durch einen Tutor bietet sich eher zur Bepreisung an, als der dafür erforderliche Austausch von E-Mails.

<sup>187</sup> Z. B. im Software Engineering Management im Wintersemester 2002/2003 und im Information Management im Wintersemester 2003/2004.

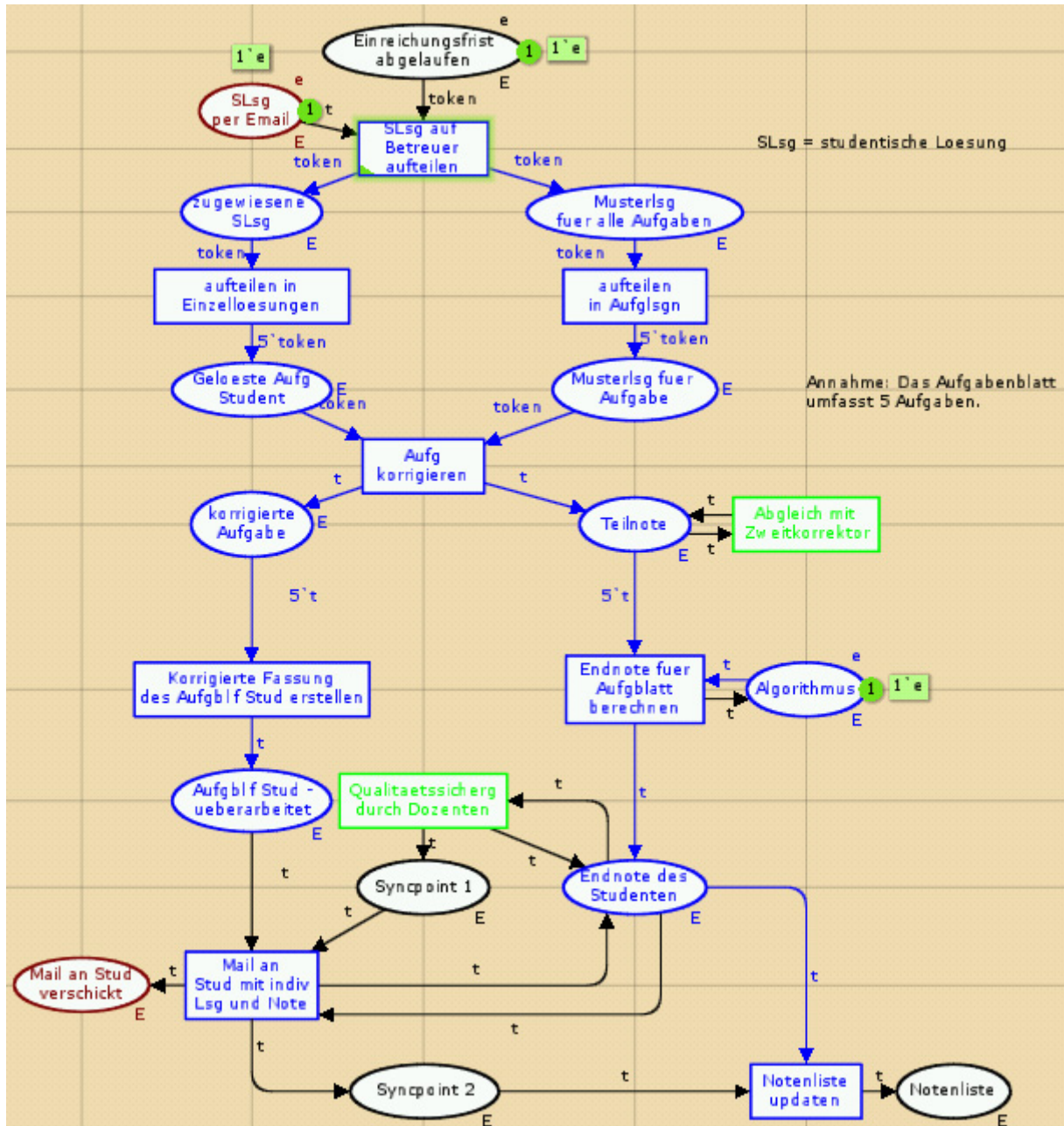


Abbildung 77: Korrektur eines Aufgabenblatts

Die Korrektur eines Tests läuft dabei wie folgt ab:

Nachdem die Frist zur Einreichung des Tests durch die Studierenden verstrichen ist, werden die eingegangenen Lösungen der Studierenden auf die Betreuer des Kurses aufgeteilt und die Aufgaben anhand einer Musterlösung bewertet. Die Bewertungshinweise zu den Aufgabenlösungen werden gesammelt und die Teilnoten durch einen Zweitkorrektor geprüft. Teilnoten werden mit Hilfe eines Algorithmus, z. B. Gewichtung nach dem Schwierigkeitsgrad der Aufgabe, zu einer Endnote für das Aufgabenblatt verdichtet. Am Ende des Benotungsprozesses findet eine Kontrolle durch den Dozenten statt, der die Lehrveranstaltung leitet.



Erst nach dieser Kontrolle wird eine Mail mit individuellen Lösungshinweisen und der Endnote an den Studierenden geschickt. Nach dem Versand der Mail wird die Note in die Notenliste übernommen.

### 7.2.2 Portalarchitektur eines Lern-Managementsystems

Die Verfügbarkeit eines WfMS ist eine notwendige Voraussetzung für die Implementierung eines Lern-Managementsystems als Portal (Cesarini u.a. 2004; Gruene u.a. 2004). Portal-Server ermöglichen die Integration verteilter Anwendungen sowie die Bereitstellung von Sichten für unterschiedliche Nutzergruppen, die so genannten *Portale*. Dabei werden Anwendungen durch eine serviceorientierte Architektur auf Basis von Web-Services miteinander „lose“ gekoppelt und in einem webbasierten Informationssystem bereitgestellt (Liu u.a. 2003; Reklaitis u.a. 2003; Vossen und Westerkamp 2003). Als Standard zur Orchestrierung von Web-Services hat sich die Business Process Execution Language for Web-Services (BPEL4WS) etabliert.<sup>188</sup> Es ist zu erwarten, dass die Kombination von Web-Services über BPEL4WS in Zukunft die Kommunikation zwischen Komponenten und die Integration neuer Komponenten in LMS erleichtern wird. Web-Services verfügen leider nicht über grafische Elemente, die die Darstellung der in ihnen gekapselten Funktionen erlauben. Der Portlet-Standard erweitert Web-Services um grafische Konzepte. Portal-Server erweitern Application-Server, auf denen Web-Services gehostet werden können, indem sie die Portlet-Spezifikationen WSRP und JSR 168 unterstützen (Abdelnur und Hepper 2003; OASIS 2003; Bellas 2004). Portlets sind kleine, „self-contained“ Applikationen, die in personalisierte, webbasierte Informationssysteme (Portale) integriert werden können (Clarke 2004). Sie erlauben die Kommunikation untereinander über J2EE Session-Kontexte (vgl. Java 2 Platform Enterprise Edition; <http://java.sun.com/j2ee/>) und unterstützen *Single-Sign-On-Mechanismen* (Díaz und Paz 2005).

---

<sup>188</sup> <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>

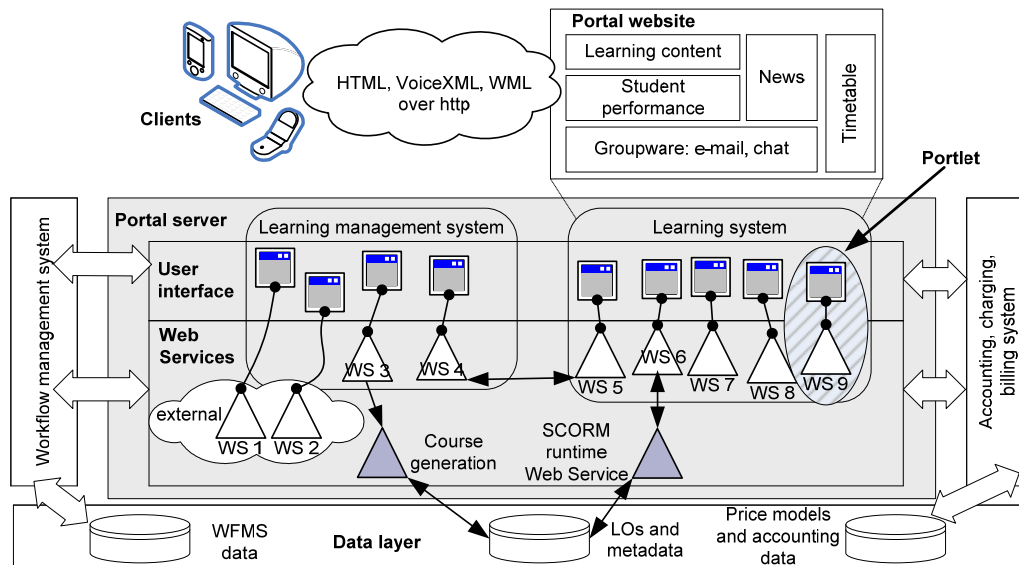


Abbildung 78: Portal-Architektur eines verteilten LMS<sup>189</sup>

In Abbildung 78 wird eine Portal-Architektur für ein LMS vorgestellt. Neben einem Portal für die Lernenden (Learning System) wird ein Portal für die Verwaltung von Lerninhalten (Learning Management System) bereitgestellt. Portlets von Third-Party-Providern können durch *Proxies* integriert werden.<sup>190</sup> Web-Services WS1 und WS2 werden nebst ihrer Layout-Informationen auf Remote-Servern gehostet. Die Kommunikation zwischen Lernenden und Lehrenden über das LMS wird durch Web-Services in der Workflow-Ebene der Portal-Architektur implementiert. Sobald ein Lehrender eine E-Mail über das Portal an einen Lernenden absendet, werden die Inhalte der Nachricht durch den Web-Service WS4 verarbeitet und an WS5 übergeben. Daraufhin stellt das zu WS5 gehörige Portlet die Inhalte der E-Mail dem Lernenden in seinem Portal dar. Web-Services als Teil von Portlets und Web-Services für Hintergrund-Aufgaben, z. B. zur Ermittlung von Noten, können mit den zugrunde liegenden Datenbanken des LMS kommunizieren. Zum Ablegen von Lernobjekten im LMS kann ein Lehrender über das Portal auf WS3 zugreifen. WS3 prüft, ob die Lerninhalte standardkonform beschrieben wurden und übernimmt diese anschließend in die Datenbank für Lerninhalte. Ein Lernender kann sich die Lerninhalte über WS6, den „SCORM Runtime Web-Service“ anzeigen lassen.

<sup>189</sup> Quelle: (Gruene und Oberweis 2005).

<sup>190</sup> Ein Proxy gestattet die Einrichtung einer Kommunikation über SOAP-Nachrichten mit WSRP-konformen Portlets auf einem entfernten Server. Dabei werden auch die Informationen für das Rendering abgerufen.

### 7.2.3 Bewertung der Portalarchitektur für Lern-Managementsysteme

Die Gestaltung eines Lern-Managementsystems durch Anwendung einer Portalarchitektur bietet die folgenden Vorteile:

- Web-Services-Architekturen als Teil von Portalen unterstützen die Wiederverwendung von *Altsystemen*, indem diese über standardisierte Schnittstellen beschrieben und zugreifbar gemacht werden (vgl. (Sneed 2006)).
- Portale und Web-Services-Architekturen eignen sich zur Darstellung *verteilter Systeme* (Kumar 2005). E-Learning-Provider können die bestehenden Systeme anderer Provider nutzen und müssen diese nicht erneut implementieren (Vossen und Westerkamp 2005). Gerade bei kostenintensiven Anwendungen, die hohe Anforderungen an Hardware / Software und hohe personelle Anforderungen an die Wartung stellen, ist dies aus Gründen der Effizienz sinnvoll.<sup>191</sup>
- Portlets können an die *grafische Oberfläche* / das Corporate Design des Bildungsproviders angepasst werden. Bei Bedarf lassen sich außerdem Portale für neue *Nutzergruppen* erstellen, die auf den bestehenden Web-Services / Portlets aufsetzen. Die gleiche Funktionalität kann so z. B. regulären Studierenden und Executive Students einer Universität unter unterschiedlichen „Marken“ angeboten werden.
- Die Beschreibung bestehender Systeme als serviceorientierte Architektur (SOA) kann helfen, *Schwachstellen*, z. B. Redundanzen in der Systemlandschaft eines Bildungsproviders, zu *identifizieren*.
- Die Kommunikation zwischen Teilsystemen kann durch *Workflows*, z. B. mittels BPEL4WS, automatisiert werden. Prüfungsergebnisse können so bspw. durch das LMS an ein Prüfungsverwaltungssystem übermittelt werden.

Neben den zuvor genannten Vorteilen können Probleme bei der Implementierung eines Portals entstehen:

---

<sup>191</sup> (Ramboll 2004) stellen fest, dass über 50% der europäischen Hochschulen mit anderen Hochschulen oder Bildungsanbietern kooperieren.

- Die Einrichtung eines Portals erfordert die Verfügbarkeit geschulter Mitarbeiter, die auch die Betreuung des laufenden Systems übernehmen und bei Bedarf neue Anwendungen in die serviceorientierte Architektur integrieren. Die Organisationsstruktur des Bildungsproviders muss entsprechend ausgelegt sein bzw. auf die wachsenden Bedürfnisse abgestimmt werden.
- Daten, die zwischen Anwendungen in einer SOA im E-Learning ausgetauscht werden, sind oft personenbezogen und machen so eine verschlüsselte Übertragung erforderlich. Ein entsprechendes Sicherheitskonzept muss vom Bildungsprovider entwickelt werden. Durch die Einbindung sicherheitskritischer Systeme in eine SOA entstehen Angriffsmöglichkeiten. Eine Absicherung der Kommunikation in SOA durch XML-Signaturen wird in (Hermann und Kessler 2005) beschrieben.
- Um Anwendern einen zentralen Zugriff auf Informationen über die in Portlets gekapselten Anwendungssysteme zu ermöglichen (Single-Sign-On-Verfahren), muss eine zentrale Datenbank mit Kennwörtern und Nutzerkonten eingerichtet werden, aus der der Portal-Server die erforderlichen Berechtigungen für die Anwendungen abfragen kann.
- Für die Nutzung kommerzieller Portal-Server, z. B. IBM-WebSphere ([www.ibm.com/software/webservers/portal](http://www.ibm.com/software/webservers/portal)) und BEA Weblogic Portal ([www.bea.com/products/weblogic/portal](http://www.bea.com/products/weblogic/portal)) und für Entwicklungsumgebungen zur Erstellung von Portalen fallen Lizenzkosten an.<sup>192</sup>

---

<sup>192</sup> Obwohl für Portal-Server im Open Source keine Gebühren anfallen, sind Beratungsleistungen in der Regel kostenpflichtig (s. z. B. [http://www.sun.com/software/products/portal\\_srvr/index.xml](http://www.sun.com/software/products/portal_srvr/index.xml)).

## 8 Schlussbetrachtung

### 8.1 Zusammenfassung

In der Arbeit wurde ein Konzept entwickelt, das eine modellbasierte Betrachtung des Pricings digitaler Produkte ermöglicht.

Dazu wurde neben einer Beschreibungssprache *PriMoL* ein Prototyp zur Erzeugung von grafischen Preismodellen erstellt.

Durch den Entwurf der *Beschreibungssprache PriMoL* wird eine *Orthosprache* definiert, mit der das Ziel verfolgt wird, *sprachliche Konzepte im Pricing aus dem Blickwinkel der Wirtschaftsinformatik zu normieren*. Dabei lassen die formale Definition von PriMoL und die Darstellung der zugrundeliegenden Methodik, die bei der Sprachdefinition verwendet wird, eine Weiterentwicklung durch den Anwender zu. Dieser wird durch die Arbeit in die Lage versetzt, eigenständig neue Konzepte in PriMoL einzubetten und PriMoL entlang seiner Bedürfnisse zu erweitern.<sup>193</sup> Das gleiche gilt für den Entwurf neuer grafischer Repräsentationselemente der Sprache. Um eine Unabhängigkeit der Modellierungssprache von Anwendungsdomänen zu erreichen wurde neben der *Digitalitätsannahme der Dienste und Produkte* weitgehend darauf verzichtet, Annahmen über die zugrundeliegenden Technologien und Service-Provisioning-Systeme in die Sprache aufzunehmen. Die Beschreibungssprache ist daher als *konzeptuelle Sprache* zu betrachten. Es ist jedoch für den Nachweis der Anwendbarkeit unumgänglich gewesen, die Sprache auch um so genannte SQL-Meter, d.h. Messpunkte („Datenfühler“), die in SQL definiert werden, zu erweitern.

In einem E-Learning-Setting wurde anhand eines *selbst erstellten Lern-Managementsystems* die Anwendbarkeit im Sinne der *Implementierbarkeit der erzeugten Modelle* überprüft. Dazu wurde eine Laufzeitumgebung in PHP umgesetzt, mit der Preismodelle instanziiert werden und die eine Einbettung in PHP-basierte Systeme zulässt. Im prototypischen E-Learning-Setting kann gezeigt werden, wie Anwender on-the-fly Preismodelle austauschen können und mittels des Price-Modeler-Prototyps neue Modelle, z.B. auf Basis bestehender Modelle entwickelt werden können. Damit einher gehen eine Reduzierung der Kosten des

---

<sup>193</sup> Die Sprachdefinition umfasst neben der Syntax auch die Semantik.

Pricings selbst und eine Beschleunigung des Prozesses der Einführung neuer Tarifsysteme.<sup>194</sup>

Die *Immaterialität der betrachteten Produkte* stellt dabei erhöhte Anforderungen an die zugrundeliegenden Abrechnungssysteme im E-Learning, bietet jedoch auch Vorteile gegenüber konventionellen Produkten:

Bei digitalen Produkten ist durch die Immaterialität ein *unbegrenzt* Produktionspotenzial gegeben. Diese Annahme ist bei digitalen Dienstleistungen / Prozessen *im E-Learning eher unzutreffend*, da sich E-Learning-Prozesse meist durch einen hohen Grad an Interaktion menschlicher Akteure mit einem Lern-Managementsystem bzw. der menschlichen Akteure untereinander auszeichnen. Die Einführung von *Workflow-Managementsystemen* kann jedoch einen Anreiz bieten, Prozesse im E-Learning zu „optimieren“, indem der Grad der Automatisierung z. B. in der Betreuung eines Kurses erhöht wird. Als Vision wurde dazu im Kapitel 7 die Architektur eines Portal-basierten Lern-Managementsystems mit Workflowunterstützung skizziert.

Durch die Immaterialität ist zudem die *Messung von Verbräuchen* und Dienstleistungsansprüchen durch die Kunden gewährleistet. Der Konsum eines digitalen Produktes erzeugt stets Daten in einem Service-Provisioning-System.<sup>195</sup>

Da die Granularität der Produkte im E-Learning sehr unterschiedlich sein kann, kann eine Vielzahl von „Datenpunkten“ entstehen, die für eine Bepreisung berücksichtigt werden müssen. Eine Beschreibungssprache muss dafür geeignete Mechanismen vorhalten, die alle denkbaren *Verdichtungsregeln* für eine Preisermittlung über eine große Anzahl von Datenpunkten unterstützen.

Auch der *Produktionsprozess* ist im E-Learning oft über mehrere Anwendungen verteilt, z. B. greifen Lernende neben dem Lern-Managementsystem auf digitale Bibliotheken und – gerade im Fachgebiet Wirtschaftsinformatik – auf die durch den Bildungsprovider gehosteten Serversysteme (Datenbanken, Datawarehouse-Systeme etc.) zu, um ihr Wissen zu vertiefen. Um dem Abnehmer der Bildungsleistung eine einheitliche Bepreisung quer über die Vielzahl beteiligter Anwendungen mittels eines Preismodells zu bieten, muss ein Modellierungswerkzeug und auch die Laufzeitumgebung *Schnittstellen* zu den vorgenannten

---

<sup>194</sup> Zur Problematik der Kosten einer Preisänderung vgl. (Bergen u.a. 2003).

<sup>195</sup> Selbst wenn keine Daten erzeugt werden, kann durch Einführung von Logging-Mechanismen eine Datengenerierung implementiert werden.

Systemen bereitstellen. Dabei wurde im Rahmen der Arbeit vereinfachend davon ausgegangen, dass die Nutzung durch den Bildungsabnehmer in einer relationalen Datenbank protokolliert wird, auf die zwecks Auswertung zugegriffen werden kann. Der Price-Modeler-Prototyp unterstützt zurzeit die Definition des *Datenzugriffs über beliebige relationale Datenbanken*, solange für diese ein JDBC-Interface verfügbar ist. Zu Testzwecken können die Datensätze, die mittels SQL-Metern definiert wurden, im Modeler ausgelesen werden. Eine Erweiterung zur Messung von XML-gespeicherten Daten ist durch Austausch von Programmpaketen aufgrund der *Modularität des Java-Prototyps* durch geringen Aufwand möglich (vgl. Abbildung 61, S. 171).

Die Entwicklung der Beschreibungssprache und des Prototyps ist stark durch Ergebnisse im Forschungsprojekt „PREMIUM“ an der Johann Wolfgang Goethe-Universität Frankfurt vorangetrieben worden. Die Diskussion mit Forschungspartnern im Projekt ermöglichte neben der Weiterentwicklung der Sprache auch einen Blick auf andere Verfahren der Preisbildung, die aktuell im wissenschaftlichen Interesse stehen, z. B. dynamische Preisermittlungsverfahren (Reverse Auctions, kombinatorische Auktionen etc.) Daneben wurden mit Experten der Preisbildung im Versicherungs- und im Telekommunikationsbereich Konzepte der Sprache diskutiert sowie eine Fortführung der Entwicklung durch eine Einbindung der erstellten Software in professionelle, nicht modellbasierte Abrechnungssysteme angedacht.

Die Beschreibung von Preismodellen und die Verwendung der Beschreibungssprache PriMoL wurden im E-Learning demonstriert. PriMoL ist jedoch nicht auf diesen Anwendungsbereich beschränkt. Eine *Verallgemeinerbarkeit* der Ergebnisse wurde zudem durch die strukturelle Betrachtung von Tarifen und deren Inbezugsetzung zu Preismodellen im Versicherungsbereich sowie im Mobilfunk untersucht.

## **8.2 Trends und Fortentwicklung der Konzepte**

Bei der Untersuchung der Preisbildungsmechanismen im E-Learning konnte festgestellt werden, dass Preismodelle im E-Learning tendenziell aus fixen Preisen für Teilleistungen bestehen (Nutzung von Ressourcen, Prüfungsgebühren etc.).

Dies führt dazu, dass Kunden i. d. R. für ein Komplettangebot zahlen und nicht für Teile eines E-Learning-Kurses.

Durch die Verfügbarkeit einer Vielzahl von Lernangeboten im Internet wird der Kunde jedoch in die Lage versetzt, zur Erlangung von Zertifikaten Leistungen bei unterschiedlichen Bildungsprovidern nachzufragen. Auch das Auftreten von Dienstleistern, die Kunden bei der individuellen Zusammenstellung von Lerninhalten durch Kombination von Lernmodulen unterschiedlicher Bildungsprovider unterstützen, ist denkbar.

Für Hochschulen bietet sich die *Gestaltung neuer Geschäftsmodelle* an, die den Vertrieb der Bildungsprodukte z. B. im Executive Training ermöglichen.

Die vorgenannten Punkte verlangen von Bildungsprovidern auch die Lizenzierung, Zertifizierung und Bepreisung von Modulen (Ausschnitten von Kursen) sowie die Gestaltung feingranularer Preismodelle, die auch Preise für diese Module umfassen.

### ***Technische und methodische Fortentwicklung***

Für einen industriellen Einsatz der vorgestellten Konzepte ist eine technische Weiterentwicklung des Prototyps notwendig. Es bietet sich an, insbesondere die Laufzeitumgebung, die zurzeit in PHP formuliert ist, in einen *Web-Service* zu überführen, der sich nahtlos in bestehende Service-Provisioning-Systeme im Web einbinden lässt („*Pricing-out-of-a-Box*“). Die im E-Learning-Setting vorgestellten Funktionen könnten dann z. B. in Web-Shops auch für die Bepreisung materieller Produkte genutzt werden.

Daneben müssen Kommunikationsschnittstellen für die *Anbindung an Billing-Systeme* geschaffen werden, die eine rechtssichere Zahlungsabwicklung ermöglichen.

Als funktionale Erweiterung wäre ein *Simulationswerkzeug* wünschenswert, das auf Basis von Kundendaten und Verhaltensannahmen Prognosen für Erlösänderungen bei Wechsel von Preismodellen zulässt.

Neben der programmatischen Erweiterung fehlt eine *Methode, die den Prozess des Pricings aus Sicht des Anwenders* strukturiert. Eine solche Methode sollte Hinweise bereitstellen, wie die Parameter eines Preismodells für einen konkreten Anwendungsbereich zu ermitteln sind und wie Differenzierungsstrategien in einem Preismodell formuliert werden können.



Durch die Verbreitung einer Beschreibungssprache wie PriMoL wird der Entwurf von Preismodellen erheblich vereinfacht. Es ist daher anzunehmen, dass die Anzahl der durch einen Diensteanbieter im Internet zu verwaltenden Preismodelle zunehmen wird. Eine große Anzahl von Preismodellen muss dann verwaltet und über geeignete *Wissensdatenbanken* zugänglich gemacht werden. Dabei sollte für einen Provider durch eine Wissensdatenbank die Suche nach ähnlichen Preismodellen bspw. ausgehend von den Eigenschaften digitaler Produkte (vgl. Kapitel 2.2.3) und die gezielte Suche nach bereits definierten Komponenten von Preismodellen unterstützt werden. Eine Möglichkeit zur Verwaltung von Preismodellen besteht in der Klassifizierung von Preismodellen mittels *Ontologien*.<sup>196</sup> In Ontologien werden mit Sprachen wie der Web Ontology Language (OWL) Taxonomien / Hierarchien über Elementklassen und Elemente eines Anwendungsbereichs definiert und die Taxonomien durch Querbeziehungen zwischen Elementen und Elementklassen erweitert. Durch Abfragesprachen kann dann auf die Wissensbasis zugegriffen werden. Abfragen wie „Suche mir alle verbrauchsabhängigen Preismodelle für E-Content.“ oder „Welche Produkte sind ähnlich strukturiert wie das Produkt Y?“ sind denkbar.<sup>197</sup>

Obwohl die Sprache PriMoL generisch für Preisermittlungsmechanismen entwickelt wurde, können die dargestellten Konzepte auch für die Implementierung von Berichtssystemen „zweckentfremdet“ werden. Neben dem Zugriff auf eine Vielzahl von Datenbasen unterstützt die Sprache eine Aggregation von Kennzahlen über mehrere Ebenen.

Eine Überprüfung der beschriebenen Ansätze und eine Weiterentwicklung der Sprache werden im PREMIUM-Projekt verfolgt. Dazu sind u. a. Einzelprojekte mit Praxispartnern aus dem Finanz- und Rückversicherungssektor in Vorbereitung. Zur Abbildung der Prozessunterstützung für Pricing-Prozesse wurde in einer Diplomarbeit ein auf Microsoft Visio basierendes Petri-Netzwerkzeug zur Manipulation von Daten in ODBC-konformen Datenbanken

---

<sup>196</sup> Nach (Gruber 1993) sind Ontologien aus Sicht der Informatik explizite Spezifikationen von Entities (Objekten und Konzepten) und der Beziehungen, die zwischen diesen gelten in einer abstrakten und vereinfachten Sicht eines betrachteten Realitätsausschnitts. Ontologien werden stets mit dem Ziel einer späteren Auswertung aufgestellt.

<sup>197</sup> Das in der Arbeit entwickelte Metamodelle nebst Orthosprache kann als Ontologie angesehen werden.

erstellt, mit dem Prozesse im Pricing beschrieben und instanziiert werden können (vgl. (Basoglu 2006)). Das in der Diplomarbeit erstellte Werkzeug kann somit als Definitionsoberfläche und Ablaufunterstützung in eine Web-Service-basierte Laufzeitumgebung für Preismodelle integriert werden. Eine flexible Anpassung von Prozessen im Pricing an die Erfordernisse eines Diensteanbieters, wie in Kapitel 7.2 am Beispiel des E-Learnings skizziert, wird somit möglich.

## Anhänge

### A.1 Virtuelle Studiengänge, virtuelle Universitäten und Suchmaschinen für E-Learning

Die folgenden Tabellen umfassen eine Linksammlung zu E-Learning und verwandten Themengebieten (Datum: 03.07.2006) ohne Anspruch auf Vollständigkeit.

<b>Studium im Netz</b>	
<b>Virtuelle Studiengänge und Studiengänge mit Präsenzphasen</b>	
Multimediales Fernstudium Medizinische Informatik: Projekt im BMBF-Förderprogramm „Neue Medien in der Hochschullehre“	<a href="http://www.medin.info">www.medin.info</a>
Virtueller Weiterbildungsstudiengang Wirtschaftsinformatik: Abschluss mit dem MSc	<a href="http://www.vawi.de">www.vawi.de</a>
Bachelor of Arts „Soziale Arbeit“: Fernstudium der HS Fulda, FH Koblenz, FH Potsdam und FH Münster; gefördert durch das BMBF-Programm „Multimedia in der Hochschule“	<a href="http://www.basa-online.de">www.basa-online.de</a>
Technische Akademie Südwest e.V. Weiterbildungsstudiengänge „Bauschäden“ und „Grundstücksbewertung“ mit unterschiedlichen Abschlüssen	<a href="http://www.tas-kl.de">www.tas-kl.de</a>
Fachakademie Saar für Hochschulfortbildung: juristische und betriebswirtschaftliche, berufsbegleitende Studiengänge im Fernstudium	<a href="http://www.e-fsh.de">www.e-fsh.de</a>
European School of Business Reutlingen: MBA-Programm im Fernstudium mit Präsenzphasen, FIBAA-akkreditiert	<a href="http://www.esb-reutlingen.de/mba.html">www.esb-reutlingen.de/mba.html</a>
Rheinahrcampus MBA	<a href="http://www.rheinahrcampus.de/mba/">www.rheinahrcampus.de/mba/</a>
College of Estate Management: Angebot eines MBA in Construction and	<a href="http://www.cem.ac.uk">www.cem.ac.uk</a>

Real Estate sowie eines MSc in Real Estate	
Zentrum für Fernstudien und universitäre Weiterbildung TU Kaiserslautern: Vielzahl von Studiengängen, die dem ECTS unterliegen	<a href="http://www.zfuw.de">www.zfuw.de</a>
Master of Public Policy: Ausbildung von Management-Nachwuchs für die Öffentliche Verwaltung und Non-Profit-Organisationen	<a href="http://publicpolicy.uni-erfurt.de/master.html">publicpolicy.uni-erfurt.de/master.html</a>
Educatis Graduate School of Management: Schweizer Online-Universität mit unterschiedlichen Abschlüssen (MBA, MSc, Dr. und PhD); Voraussetzung: abgeschlossenes Hochschulstudium	<a href="http://www.educatisgsm.de">www.educatisgsm.de</a>
<b>Berufsbegleitend</b>	
Verbundstudium NRW: Das Verbundstudium ermöglicht den Erwerb des Fachhochschulabschlusses.	<a href="http://www.verbundstudium.de/vs_studieren/index.htm">www.verbundstudium.de/vs_studieren/index.htm</a>
<b>Virtuelle Universitäten</b>	
Virtual Globale University, Master of Business Informatics (s. Kapitel 2.5.6.3)	<a href="http://www.vg-u.de">www.vg-u.de</a>
Open Universiteit, Niederlande: akademische Studienprogramme (Master), Erwachsenenbildung und angepasste Inhouse Trainings für Firmen	<a href="http://www.ou.nl">www.ou.nl</a>
Open University, UK: Erwerb von Zertifikaten, Diplomas, BA-, BSc-, MBA- und MSc-Degrees	<a href="http://www.open.ac.uk">www.open.ac.uk</a>
African Virtual University: Zertifikate, Degree- und Diploma-Programme	<a href="http://www.avu.org">www.avu.org</a>
Canadian Virtual University: Zusammenschluss kanadischer Universitäten; Vielzahl unterschiedliche Abschlüsse (bis zur Promotion) und Zertifikate	<a href="http://www.cvu-uvc.ca">www.cvu-uvc.ca</a>
Finnish Virtual University: Zusammenschluss finnischer Universitäten; Angebot unterschiedlicher Kurse im Graduate- und Postgraduate-Studium	<a href="http://www.virtuaaliyliopisto.fi">www.virtuaaliyliopisto.fi</a>

Global Virtual University: zwei Masterstudiengänge und Kurse in Umweltmanagement	<a href="http://www.gvu.unu.edu">www.gvu.unu.edu</a>
EURO-Mediterranean Open Sky University: zwei University Degrees (IT und Mechanical Engineering) sowie 13 Kurse auf Volkshochschulniveau (Vocational Training)	<a href="http://www.uninettuno.it/mednetu/e/skins/mednetu/home/index_1024.asp">www.uninettuno.it/mednetu/e/skins/mednetu/home/index_1024.asp</a>
Virtuelle Hochschule Bayern: virtueller Verbund von (Fach-)Hochschulen und Universitäten; Kurse nach ECTS bewertet; Studierende der Trägerhochschulen können die Kurse kostenfrei nutzen	<a href="http://www.vhb.org">www.vhb.org</a>
Hamburger Fernhochschule: Bachelor- und Masterstudiengänge mit Präsenzphasen	<a href="http://www.fern-fh.de">www.fern-fh.de</a>
Teleakademie Hochschule Furtwangen: Online- Kurse zu überwiegend informatischen Themenstellungen	<a href="http://www.tele-ak.de">www.tele-ak.de</a>
Fernfachhochschule Riedlingen: Unterschiedliche Fernstudiengänge, Präsenzphasen und Offline-Lernmaterial	<a href="http://www.fh-riedlingen.de">www.fh-riedlingen.de</a>

### **Förderung von E-Learning**

Portal zur BMBF-Förderung „Neue Medien in der Bildung“	<a href="http://www.medien-bildung.net">www.medien-bildung.net</a>
E-Learning Europa	<a href="http://www.elearningeuropa.info">www.elearningeuropa.info</a>
eContentPlus: EU-Förderprogramm, um die Zukunftsfähigkeit von Online-Lerninhalten zu erhöhen	<a href="http://europa.eu.int/information_society/activities/econtentplus/index_en.htm">europa.eu.int/information_society/activities/econtentplus/index_en.htm</a>
Eumedis: EU-gefördertes Programm zur Weiterentwicklung der „Information Society“ im Mittelmeerraum, insb. im Bereich „Education“	<a href="http://www.eumedis.net/en/">www.eumedis.net/en/</a>
Überblick über europäische Programme der European Association of Distance Teaching Universities	<a href="http://www.eadtu.nl/default.asp?hId=5&amp;mmId=20">www.eadtu.nl/default.asp?hId=5&amp;mmId=20</a>
European eLearning Industry Group: Konsortium von IT-Firmen und Con-	<a href="http://www.elig.org">www.elig.org</a>

tent-Providern, das sich zum Ziel gesetzt hat, den Einsatz von E-Learning in Europa zu fördern

<b>Suchmaschinen</b>	
Ploteus: Suchportal für Aus- und Weiterbildungsmöglichkeiten in Europa	<a href="http://europa.eu.int/ploteus/portal/home.jsp">europa.eu.int/ploteus/portal/home.jsp</a>
Hobson's Distance Learning UK Courses: Online-Suchmaschine für Studiengänge, bietet auch Recruiting an	<a href="http://www.distancelearning.hobsons.com">www.distancelearning.hobsons.com</a>
Hyperstudy: Study Abroad Program Searches: für Universitäts-, Sprachkurse	<a href="http://www.hyperstudy.com">www.hyperstudy.com</a>
Distance Learning Course Finder:	<a href="http://www.dlcoursefinder.com">www.dlcoursefinder.com</a>
Study2U: für Distance Learning und Online Degree-Programme (Bachelors, Masters, MBAs, Doctorate)	<a href="http://www.study2u.com">www.study2u.com</a>
List of Virtual Universities: Liste mit virtuellen Universitäten	<a href="http://www.adm.hb.se/~ubj/vus.html">www.adm.hb.se/~ubj/vus.html</a>
American Distance Education Consortium: Umfangreiche Übersicht über amerikanische Fernlernprogramme	<a href="http://www.adec.edu/virtual.html">www.adec.edu/virtual.html</a>
geteducated.com: Auflistung von Studienprogrammen, die den Qualitätsstandards von geteducated genügen	<a href="http://www.geteducated.com">www.geteducated.com</a>
Studieren.de: Suchmaschine für Studienprogramme, u. a. Fern- und Onlinestudien, gesponsert von der FH München	<a href="http://www.studieren.de">www.studieren.de</a>
Studieren-im-Netz.de: Universitäten können selbstständig Informationen zu den eigenen Online-Studienangeboten einstellen; gefördert von der Bund-Länder-Kommission	<a href="http://www.studieren-im-netz.de">www.studieren-im-netz.de</a>

<b>Organisationen</b>	
European Association of Distance Teaching Universities: Organisation der freien und Distance Learning-Universitäten in Europa.	<a href="http://www.eadtu.nl">www.eadtu.nl</a>
United States Distance Learning Association: amerikanische Non-Profit-Organisation für Distance Learning	<a href="http://www.usdla.org">www.usdla.org</a>
European University Association: Organisation europäischer Universitäten und der nationalen Rektorenkonferenzen; Ziel: Entwicklung eines „kohärenten“ Systems der Hochschulbildung und -forschung	<a href="http://www.eua.be">www.eua.be</a>
Deutsche Gesellschaft für Wissenschaftliche Weiterbildung und Fernstudium e.V.: Vereinszweck besteht in der Förderung, Koordinierung und Repräsentation der von Hochschulen getragenen Weiterbildung und des Fernstudiums	<a href="http://www.dgwf.net">www.dgwf.net</a>
Arbeitsgruppe Fernstudium in der Deutschen Gesellschaft für Weiterbildung und Fernstudium: Ziel liegt in der Vertretung der Mitglieder (Hochschulen) auf europäischer Ebene, in der Verbreitung von Angeboten und im Informationsaustausch	<a href="http://www.ag-fernstudium.de">www.ag-fernstudium.de</a>
Hochschulrektorenkonferenz: Informationen zu staatlich anerkannten Hochschul-Studiengängen in Deutschland	<a href="http://www.hochschulkompass.de">www.hochschulkompass.de</a>
Zentralstelle für Fernstudien an Fachhochschulen: Informationen zum Fernstudium an Fachhochschulen	<a href="http://www.zfh.de">www.zfh.de</a>

<b>Beratung</b>	
Hochschulverbund Distance Learning: Interessensgemeinschaft von Hochschulen und Weiterbildungszentren in Deutschland	<a href="http://www.aww-brandenburg.de/HDL/">www.aww-brandenburg.de/HDL/</a>
Institut für Verbundstudien NRW: technische und fachliche Beratung der Fachhochschulen, insbesondere bei der Entwicklung von berufsbegleitenden Studienangeboten (so genanntes Verbundstudium)	<a href="http://www.ifv-nrw.de">www.ifv-nrw.de</a>

<b>Repositorien</b>	
Medienquadrat: umfasst u. a. einen Pool an Lernmaterialien für Mediengestaltung, Mediensysteme und Medienkultur	<a href="http://www.uni-weimar.de/~m2">www.uni-weimar.de/~m2</a>
Information Technology Online: Umfasst eine Auflistung von Lernmodulen	<a href="http://www.ias.uni-stuttgart.de/projekte/ito">www.ias.uni-stuttgart.de/projekte/ito</a>
Lernobjekte der Open University	<a href="http://www.ouw.co.uk">www.ouw.co.uk</a>



## A.2 Sprachdefinition

### A.2.1 Accounting-Language-Definition in EBNF

```
(* AccLan -- Accounting Language-Definition *)

<Codeblock> ::= <Variablenblock>
              <Verarbeitungsblock>
              <Returnanweisung>;

(* Hauptblöcke *)

<Variablenblock> ::= {<Variablendeklaration>};

<Verarbeitungsblock> ::= {<If-Block>|<Zuweisung>};

  <If-Block> ::=      "IF "<Vergleichsausdruck>
                    <Verarbeitungsblock>
                    "ENDIF;"

  <Zuweisung> ::= <Variablenbezeichner>'='(<Zahl>|<Variablenbezeichner>);';

  <Vergleichsausdruck> ::= ( (<Konstante>|<Variablenbezeichnung>|<Integerzahl>
                             |<Rationzahl>)
                             <Bedingung>

                             (<Konstante>|<Variablenbezeichnung>|<Integerzahl>
                             |<Rationzahl>) )
                             -
                             (<Integerzahl><Bedingung><Ratiozahl>)
                             -
                             (<Ratiozahl><Bedingung><Integerzahl>);

<Returnanweisung> ::= "RETURN "<Variablenbezeichner>;

(* Definitionen für einfache Typen *)

<Variablendeklaration> ::= "VAR "<Variablenbezeichner> ['='<Zahl>]';';

  <Variablenbezeichner> ::= <Buchstabe> {<Buchstabe>|<Ziffer>};

  <Zahl> ::= <Integerzahl>|<Ratiozahl>;

  <Integerzahl> ::= ["-"]<ZifferAußerNull>{<Ziffer>};

  <Rationzahl> ::= <Integerzahl>.'{<Ziffer>};

<Bedingung> ::= '='|>='|<='|<'|>'|<>';

<Konstante> ::= 'COUNT'|'MIN'|'MAX'|'SUM';

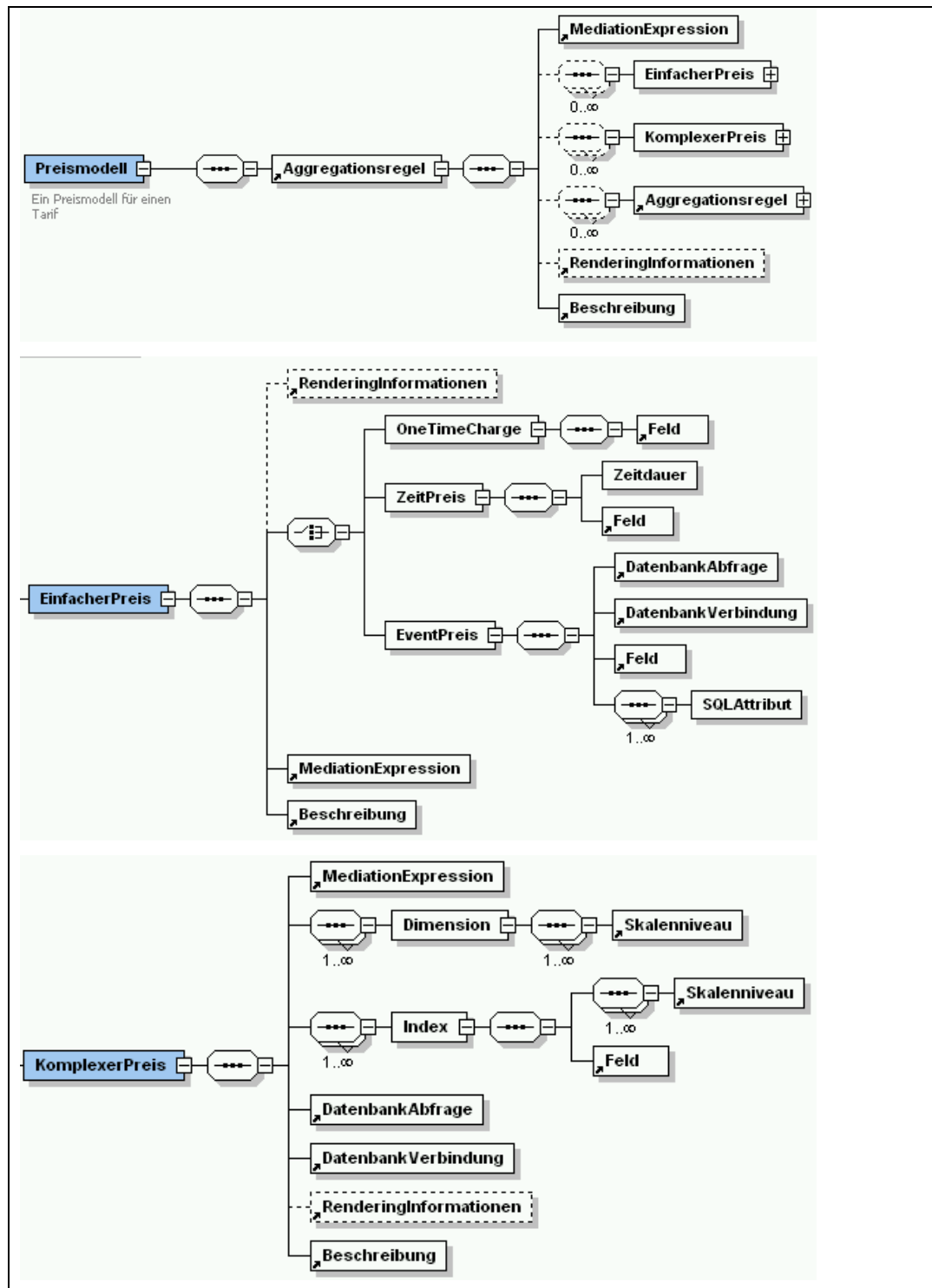
<Buchstabe> ::= 'a'|'b'|'c'|'d'|'e'|'f'|'g'|'h'|'i'|'j'|'k'|'l'|'m'|'n'|
'o'|'p'|'q'|'r'|'s'|'t'|'u'|'v'|'w'|'x'|'y'|'z'|'A'|'B'|'C'|'D'|'E'|'F'|
'G'|'H'|'I'|'J'|'K'|'L'|'M'|'N'|'O'|'P'|'Q'|'R'|
'S'|'T'|'U'|'V'|'W'|'X'|'Y'|'Z';

<ZifferAußerNull> ::= '1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9';

<Ziffer> ::= '0' | <ZifferAußerNull>;
```

Abbildung 79: Accounting Language AccLan als Erweiterte Backus-Naur-Form

## A.2.2 XML-Schema von PriMoL

Abbildung 80: XML-Schema von PriMoL in XML-Spy-Notation<sup>198</sup><sup>198</sup> XML-Spy ist ein XML-Editor der Firma Altova (www.altova.com).

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="Preismodell"><xs:complexType>
  <xs:sequence><xs:element ref="Aggregationsregel"/></xs:sequence>
  <xs:attribute name="Nr" type="xs:anySimpleType"/>
  <xs:attribute name="Ersteller" type="xs:string"/>
</xs:complexType></xs:element>

<xs:element name="MediationExpression"><xs:complexType>
  <xs:attribute name="Nr" type="xs:anySimpleType"/>
  <xs:attribute name="CodeBlock" type="xs:string"/>
</xs:complexType></xs:element>

<xs:element name="Aggregationsregel"><xs:complexType>
<xs:sequence>
  <xs:element ref="MediationExpression"/>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="EinfacherPreis">
<xs:complexType><xs:sequence>
<xs:element ref="RenderingInformationen" minOccurs="0"/>
<xs:choice>

<xs:element name="OneTimeCharge"><xs:complexType><xs:sequence><xs:element
ref="Feld"/>
  </xs:sequence>
  <xs:attribute name="Datum" type="xs:anySimpleType" use="optional"/>
  </xs:complexType>
</xs:element>

<xs:element name="ZeitPreis"><xs:complexType><xs:sequence>
  <xs:element name="Zeitdauer">
    <xs:complexType>
<xs:attribute name="Jahre" type="xs:integer" use="optional" default="0"/>
<xs:attribute name="Monate" type="xs:integer" use="optional" default="0"/>
<xs:attribute name="Tage" type="xs:integer" use="optional" default="0"/>
<xs:attribute name="Stunden" type="xs:integer" use="optional" default="0"/>
<xs:attribute name="Minuten" type="xs:integer" use="optional" default="0"/>
<xs:attribute name="Sekunden" type="xs:integer" use="optional" default="0"/>
    </xs:complexType>
  </xs:element>
  <xs:element ref="Feld"/>
</xs:sequence>
<xs:attribute name="StartDatum" type="xs:dateTime" use="required"/>
<xs:attribute name="EndDatum" type="xs:dateTime" use="optional"/>
</xs:complexType>
</xs:element>

<xs:element name="EventPreis"><xs:complexType><xs:sequence>
  <xs:element ref="DatenbankAbfrage"/>
  <xs:element ref="DatenbankVerbindung"/>
  <xs:element ref="Feld"/>
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="SQLAttribut"><xs:complexType>
      <xs:attribute name="Alias" type="xs:anySimpleType" use="required"/>
    </xs:complexType></xs:element></xs:sequence>
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:choice>
<xs:element ref="MediationExpression"/>
<xs:element ref="Beschreibung"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>

<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="KomplexerPreis">
<xs:complexType>
<xs:sequence>
<xs:element ref="MediationExpression"/>
<xs:sequence maxOccurs="unbounded">
<xs:element name="Dimension">
<xs:complexType>
<xs:sequence maxOccurs="unbounded">

```

```

<xs:element ref="Skalenniveau"/>
</xs:sequence>
<xs:attribute name="Name" type="xs:anySimpleType"/>
<xs:attribute name="SQLAttributName" type="xs:anySimpleType"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:sequence maxOccurs="unbounded">
<xs:element name="Index">
<xs:complexType>
<xs:sequence>
<xs:sequence maxOccurs="unbounded">
<xs:element ref="Skalenniveau"/>
</xs:sequence>
<xs:element ref="Feld"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:element ref="DatenbankAbfrage"/>
<xs:element ref="DatenbankVerbindung"/>
<xs:element ref="RenderingInformationen" minOccurs="0"/>
<xs:element ref="Beschreibung"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element ref="Aggregationsregel"/></xs:sequence>

<xs:element ref="RenderingInformationen" minOccurs="0"/>
<xs:element ref="Beschreibung"/>
</xs:sequence>

<xs:attribute name="Nr" type="xs:anySimpleType" use="required"/>
<xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>

<xs:element name="Skalenniveau">
<xs:complexType>
<xs:attribute name="vonInklusive" type="xs:anySimpleType"/>
<xs:attribute name="bisExklusive" type="xs:anySimpleType"/>
<xs:attribute name="DimensionName" type="xs:anySimpleType"/>
<xs:attribute name="DimensionSQLAttributName" type="xs:anySimpleType"/>
</xs:complexType>
</xs:element>

<xs:element name="DatenbankAbfrage">
<xs:complexType><xs:attribute name="CustomerIDAlias" type="xs:anySimpleType"/>
</xs:complexType></xs:element>

<xs:element name="DatenbankVerbindung"><xs:complexType>
<xs:attribute name="Server" type="xs:anySimpleType"/>
<xs:attribute name="Port" type="xs:anySimpleType"/>
<xs:attribute name="Passwort" type="xs:anySimpleType"/>
<xs:attribute name="Benutzername" type="xs:anySimpleType"/>
<xs:attribute name="Datenbank" type="xs:anySimpleType"/>
<xs:attribute name="DatenbankType" type="xs:anySimpleType"/>
<xs:attribute name="MountpointName" type="xs:anySimpleType"/>
</xs:complexType></xs:element>

<xs:element name="Feld"><xs:complexType>
<xs:attribute name="Wert" type="xs:anySimpleType"/>
</xs:complexType></xs:element>

<xs:element name="RenderingInformationen"><xs:complexType>
<xs:attribute name="X" type="xs:decimal" use="optional" default="0"/>
<xs:attribute name="Y" type="xs:decimal" use="optional" default="0"/>
<xs:attribute name="Breite" type="xs:decimal" use="optional" default="0"/>
<xs:attribute name="Hoehe" type="xs:decimal" use="optional" default="0"/>
</xs:complexType></xs:element>

<xs:element name="Beschreibung"/>
</xs:schema>

```

Abbildung 81: XML-Schema-Repräsentation von PriMoL in Textform

## A.3 Petri-Netze und Software Engineering

### A.3.1 Struktureigenschaften von Petri-Netzen

Da jedes Netz  $N$  einem Graphen im Sinne der Graphentheorie entspricht, können Netze anhand ihrer graphentheoretischen Eigenschaften klassifiziert werden.

#### *Schlichtheit*

Eine Netz  $N$  heißt schlicht, wenn gilt  $\forall x, y \in X: \bullet x = \bullet y \wedge x\bullet = y\bullet \Rightarrow x = y$ . Schlichte Netze haben keinen Knoten mit identischer Umgebung.

#### *Azyklische Netze*

Ein Netz  $N$  ist azyklisch, wenn für jedes  $x \in X$ ,  $(x, x) \notin F^+$ , wobei  $F^+$  die transitive Hülle von  $F$  bildet.<sup>199</sup> Ein Knoten  $x$  kann nicht wieder durch Schaltvorgänge erreicht werden.

#### *Isolierte Stellen*

Ein Netz  $N$  hat keine isolierten Stellen, wenn  $\forall s \in S, \mathbf{nbh}(s) \neq \emptyset$ . Eine Stelle muss also mindestens einen Vor- oder Nachbereich haben.

#### *Verzweigungen*

Ein Knoten ist vorwärtsverzweigt, wenn sein Nachbereich mindestens zwei Knoten umfasst, rückwärtsverzweigt, sofern der Vorbereich mindestens zwei Knoten umfasst.

#### *Transitions- und stellenberandete Teilnetze*

Als Rand eines Teilnetzes  $N'$  wird die Menge der Knoten bezeichnet, die mittels gerichteter Kanten mit Knoten verbunden sind, die nicht zum Teilnetz gehören.

Ein Netz  $N' = (S', T', F')$  heißt Teilnetz eines Netzes  $N = (S, T, F)$ , falls gilt:

- $S' \subseteq S$ ,
- $T' \subseteq T$  und
- $F' = F \cap ((S' \times T') \cup (T' \times S'))$ .

Der Rand des Teilnetzes  $N'$  ist definiert durch

$$\mathit{rand}(N', N) = x \in (S' \cup T') \mid \bullet x \cup x\bullet \setminus (S' \cup T') \neq \emptyset.$$

---

<sup>199</sup> Die transitive Hülle umfasst im hier dargestellten Fall sämtliche Flussrelationen, die sich aus den bestehenden Flussrelationen  $F$  des Netzes herleiten lassen. Gilt z.B.  $(a, b)$  und  $(b, c)$  als Flussrelation zwischen den Knoten  $a, b$  und  $c$ , so gilt auch  $(a, c)$ , da sich  $(a, c)$  durch Verkettung von  $(a, b)$  und  $(b, c)$  ableiten lässt.

Der Vorbereich und der Nachbereich beziehen sich dabei auf das Netz  $N$ . Das Teilnetz heißt transitionsberandet, falls  $\text{rand}(N', N) \subseteq T$ , und stellenberandet, falls  $\text{rand}(N', N) \subseteq S$ .

### A.3.2 Dynamische Eigenschaften von Petri-Netzen

Dynamische Eigenschaften beziehen sich auf die Markierung(en) eines Netzes. Im Falle Elementarer Petri-Netze beschreibt eine *Markierung* als Untermenge von  $S$  die Gesamtheit aller gleichzeitig markierten Stellen. Mit  $M_a$  sei im Folgenden die *Startmarkierung* / *Ausgangsmarkierung* eines Netzes bezeichnet. Im Folgenden sei  $N = (S, T, F, M_a)$  ein Petri-Netz mit gegebener Ausgangsmarkierung  $M_a$  und  $t \in T$  eine beliebige Transition.

#### **Folgemarkierung**

- $t$  kann bei gegebener Markierung  $M \subseteq S$  schalten ( $t$  ist aktiviert,  $t$  kann feuern), falls  $\bullet t \subseteq M$  und  $t \bullet \cap M = \emptyset$ , kurz<sup>200</sup>:  $t \mathbf{con} M$
- Sei  $M' \subseteq S$ . Dann schaltet  $t$  von  $M$  nach  $M'$ , wenn  $t \mathbf{con} M$  und  $M' = (M - \bullet t) \cup t \bullet$ , geschrieben als  $M [t \triangleright M'$ .
- $M'$  ist die *Folgemarkierung* von  $M$ .

#### **Schaltfolge, erreichbare Markierung, Lebendigkeit, Deadlock**

- Seien  $t_1 \dots t_n \in T$  mit  $n \geq 0$  eine *Folge von Transitionen*. Seien  $M, M' \in S$ . Dann schalten  $t_1 \dots t_n$  von  $M$  nach  $M'$ , falls es Markierungen  $M_0, M_1, \dots, M_n$  gibt mit  $M_0 = M_a$  und  $M_n = M'$  und  $M_{i-1} [t_i \triangleright M_i$ , für die gilt  $1 \leq i \leq n$ . Verkürzt kann dann geschrieben werden  $M [t_1 \dots t_n \triangleright M'$ .
- Sei  $x$  eine Folge von Transitionen und  $M \subseteq S$ . Dann kann  $x$  in  $M$  schalten, falls es ein  $M' \subseteq S$  gibt, so dass gilt:  $M [x \triangleright M'$ . Verkürzt:  $x \mathbf{con} M$ .
- Die Folge von Transitionen  $x$  wird *Schaltfolge zu  $N$*  genannt, falls  $x \mathbf{con} M_a$  gilt. Die *Menge aller Schaltfolgen* wird mit  $\mathbf{FS}(N)$  bezeichnet.
- $M$  ist eine *erreichbare Markierung* von  $N$ , wenn ein  $x \in \mathbf{FS}(N)$  existiert, für das gilt:  $M_a [x \triangleright M$ . Die Menge aller erreichbaren Markierungen wird mit  $C_N$  bezeichnet.

---

<sup>200</sup> Von engl.:  $t$  has concession in  $M$ , s. auch Kapitel 0

- $t$  ist lebendig in  $N$ , sofern für jedes  $M$  in  $C_N$  ein  $x$  existiert, so dass  $xt \text{ con } M$ .
- $M$  heißt *Deadlock*, wenn unter  $M$  keine Transition aktiviert ist.
- $t$  ist tot, wenn sie unter keiner Markierung  $M$  mit  $M_a[x > M$  aktiviert ist.
- Ein Netz  $N$  ist *stark lebendig*, wenn aus jeder erreichbaren Markierung aus  $C_N$  jede Transition aktivierbar ist.

### Nebenläufigkeit

Zwei Transitionen, die über keine gemeinsamen Stellen im Vor- und Nachbereich verfügen, können nebenläufig schalten.

- $U \subseteq T$  umfasst eine disjunkte Menge von Transitionen, wenn  $U \neq \emptyset$  und außerdem für zwei beliebige Transitionen  $t_1$  und  $t_2 \in U$  gilt:

$$\text{nbh}(t_1) \cap \text{nbh}(t_2) = \emptyset. \text{ Verkürzt: } \mathbf{disj}(U).$$

- $U$  kann in  $M \subseteq S$  schalten, wenn gilt  $\mathbf{disj}(U)$ , wenn der Vorbereich von  $U$  in  $M$  liegt ( $\bullet U \subseteq M$ ) und der Nachbereich nicht in  $M$  liegt  $U \bullet \cap M = \emptyset$ . Ausgedrückt durch  $U \text{ con } M$ .
- Seien  $M, M' \subseteq S$ .  $U$  kann von  $M$  nach  $M'$  schalten ( $M [U > M'$ ), wenn  $U \text{ con } M$  und  $M' = (M - \bullet U) \cup U \bullet$ . Wenn die Anzahl der Elemente in  $U \geq 2$ , dann umfasst  $U$  mindestens einen nebenläufigen Schaltvorgang von  $M$  nach  $M'$ .

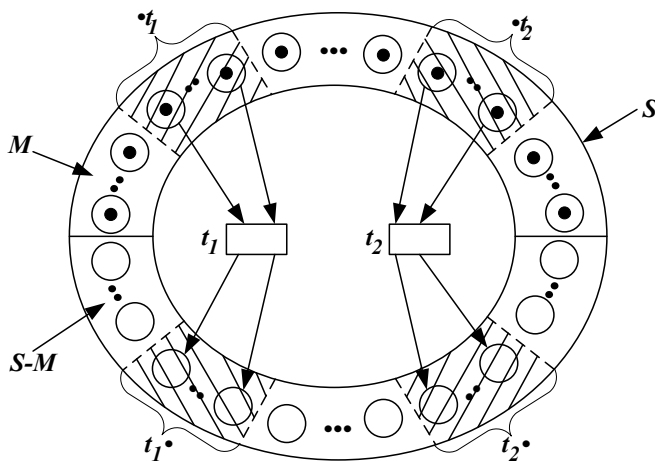


Abbildung 82: Nebenläufigkeit in Petri-Netzen<sup>201</sup>

In Abbildung 82 können  $t_1$  und  $t_2$  nebenläufig schalten. Die Vorbereiche  $\bullet t_1$  und  $\bullet t_2$  befinden sich in der Ausgangsmarkierung  $M$ . Durch den nebenläufigen

Schaltvorgang werden die Markierungen der Stellen in den Vorbereichen gelöscht und  $t_1 \bullet$  und  $t_2 \bullet$ , die Stellen in den Nachbereichen, gesetzt.

### Kausalität

Die *Kausalität* zweier Transitionen  $t_1$  und  $t_2$  wird in den folgenden beiden Abbildungen dargestellt.  $t_2$  kann erst schalten, wenn  $t_1$  geschaltet hat, siehe Abbildung 83. Es handelt sich um eine so genannte *Input-Concession*. Im Gegensatz dazu muss in Abbildung 84 erst  $t_1$  schalten, damit die vorgelagerte Transition  $t_2$  schalten kann (*Output-Concession*).  $t_1$  ist also in beiden Abbildungen Ursache für  $t_2$ . Kausalität entsteht immer dann, wenn sich Vor- und Nachbereich überdecken. Im ersten Fall gilt  $t_1 \bullet \cap \bullet t_2 \neq \emptyset$ , im zweiten  $\bullet t_1 \cap t_2 \bullet \neq \emptyset$ .

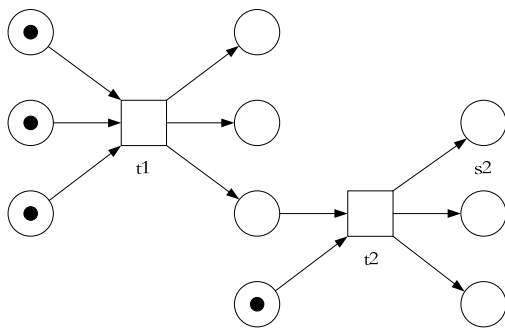


Abbildung 83: Kausalität zweier Transitionen (*Input-Concession*)<sup>202</sup>

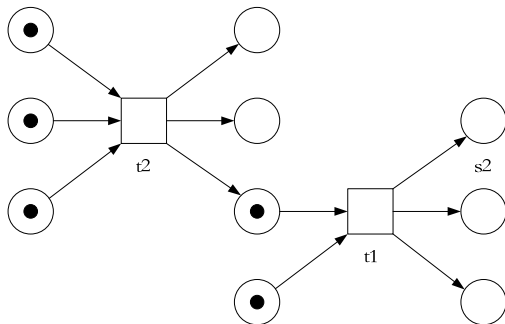


Abbildung 84: Kausalität zweier Transitionen (*Output-Concession*)<sup>203</sup>

### Konflikt

Konflikte entstehen, wenn (mindestens) zwei Transitionen gleiche Stellen im Vor- bzw. Nachbereich haben ( $\bullet t_1 \cap \bullet t_2 \neq \emptyset$  oder  $t_1 \bullet \cap t_2 \bullet \neq \emptyset$ ) und somit nicht nebenläufig schalten können. Einen Konflikt, bei dem sich Vorbereiche überdecken, nennt man *Input-Konflikt* (siehe Abbildung 85 (a)); bei

<sup>201</sup> Quelle: modifiziert nach (Rozenberg und Engelfriet 1998).

<sup>202</sup> Quelle: modifiziert nach (Rozenberg und Engelfriet 1998).

<sup>203</sup> Quelle: modifiziert nach (Rozenberg und Engelfriet 1998).



Nachbereichen wird der Konflikt als *Output-Konflikt* (siehe Abbildung 85 (b)) bezeichnet.

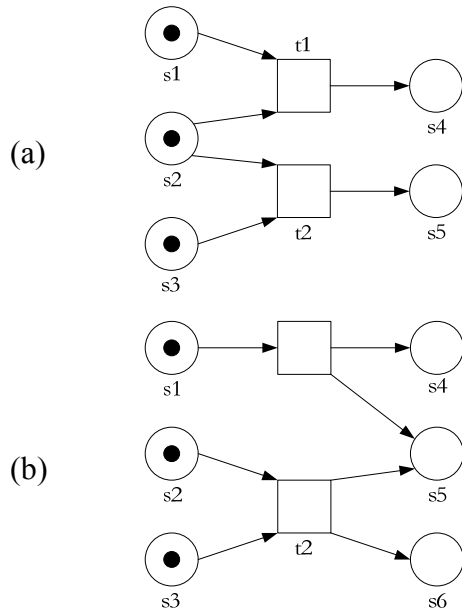


Abbildung 85: Input- und Output-Konflikt

### Kontakt

$N$  ist *kontaktfrei*, wenn für alle  $t \in T$  und  $M \in C_N$  gilt:

wenn  $\bullet t \subseteq M$ , dann auch  $t \bullet \cap M = \emptyset$ .

In Abbildung 86 liegt ein Kontakt vor, da mindestens eine der Stellen im Nachbereich von  $t$  markiert ist.  $t$  kann also erst schalten, wenn die Markierung der Stelle(n) im Nachbereich durch Schalten einer weiteren Transition aufgehoben wird.

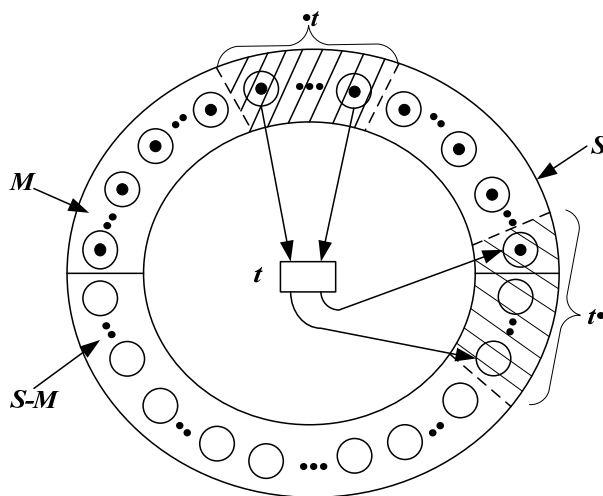


Abbildung 86: Kontakt in Elementaren Petri-Netzen

### Sequenz, Synchronisation und Zyklus

Um die Beschreibung von Petri-Netzen zu erleichtern, können neben den zuvor definierten Ablaufstrukturen Nebenläufigkeit, Kontakt und Konflikt folgende Ablaufstrukturen identifiziert werden:

Ablaufstruktur	Beispielnetz	Beschreibung
Sequenz	<p>s1      t1      s2      t2      s3</p>	Das Schalten einer Transition t2 setzt voraus, dass bereits eine andere Transition t1 geschaltet hat.
Synchronisation	<p>s1      t1      s2      t3      s5 s3      t2      s4</p>	Zur Aktivierung von t3 müssen sowohl t1, als auch t2 geschaltet haben.
Zyklus	<p>s1      t1      s2 t2</p>	Transitionen t1 und t2 finden im Wechsel statt.

Tabella 19: Ablaufstrukturen in Petri-Netzen

### A.3.3 Stellen/Transitionen-Netze

Grundlegende Konzepte von Petri-Netzen wurden anhand der in Kapitel 3.1 eingeführten Elementaren Petri-Netze beschrieben. Da Elementare Petri-Netze zur Darstellung komplexer Systeme weniger geeignet scheinen, wurden diese in den 1970er Jahren zu *Stellen/Transitionen-Netzen* (Reisig 1986b; Desel und Reisig 1998), die in diesem Abschnitt beschrieben werden, weiterentwickelt.<sup>204</sup> Stellen/Transitionen-Netze erlauben ebenso das Markenspiel mit ununterscheidbaren Marken, können aber z. B. im Gegensatz zu Elementaren Petri-Netzen über mehrere Marken in den Stellen verfügen, die dann als Behälter für Marken interpretiert werden. Ein Stellen/Transitionen-Netz, dessen Stellen alle als

<sup>204</sup> Es existieren zwei Auffassungen: Zum einen können S/T-Netze als Generalisierung von Elementaren Petri-Netze angesehen werden. Andererseits können S/T-Netze als „Kurzschreibweise“ und Abstraktion von Elementaren Petri-Netzen betrachtet werden, bei der

Bedingungen interpretiert werden können, entspricht einem Bedingungs/Ereignis-Netz (Desel und Oberweis 1996).

### **Formale Definition**

Ein Tupel  $N = (S, T, F, K, W, M_0)$  heißt *Stellen/Transitionen-System* (S/T-System), wenn gilt:

- (1)  $(S, T, F)$  ist ein Netz, bei dem  $S$  die Menge der Stellen und  $T$  die Menge der Transitionen bezeichnet.
- (2)  $K: S \rightarrow \mathbb{N}^+$ . Die Abbildung  $K$  ordnet jeder Stelle aus  $S$  eine Kapazität zu.
- (3)  $W: F \rightarrow \mathbb{N}^+$ . Die Gewichtungsfunktion  $W$  ordnet jeder Kante eine Kantengewichtung zu<sup>205</sup>.
- (4)  $M_0: S \rightarrow \mathbb{N}$  mit  $\forall s \in S: M_0(s) \leq K(s)$ .  $M_0$  heißt Anfangsmarkierung bzw. Anfangsmarkierungsfunktion des Netzes.

Als *S/T-Netz* wird das Tupel  $N = (S, T, F, K, W)$  bezeichnet.

### **Eigenschaften**

Gegeben sei das S/T-System aus dem vorherigen Abschnitt. Die strukturellen Eigenschaften von Elementaren Petri-Netzen, die in Abschnitt 0 beschrieben wurden, gelten analog bei S/T-Netzen, müssen jedoch in ihrer Definition entsprechend angepasst werden.

Es gilt:

- (1)  $M: S \rightarrow \mathbb{N}^0$  wird *Markierung* von  $N$  genannt mit  $\forall s \in S: M(s) \leq K(s)$ .
- (2) Eine *Transition*  $t$  ist *aktiviert* unter der Markierung  $M$  ( $t \text{ con } M$ ), wenn
  - a.  $\forall s \in \bullet t: M(s) \geq W(s, t)$  und
  - b.  $\forall s \in t \bullet: M(s) + W(t, s) \leq K(s)$ .
- (3) Schaltregel: Wenn  $t$  unter  $M$  aktiviert ist, dann entsteht beim Schalten von  $t$  eine *Folgemarkierung*  $M'$ , für die gilt:

$$M'(s) = \begin{cases} M(s) - W(s, t) & , \text{ falls } s \in \bullet t \setminus t \bullet \\ M(s) + W(t, s) & , \text{ falls } s \in t \bullet \setminus \bullet t \\ M(s) - W(s, t) + W(t, s) & , \text{ falls } s \in \bullet t \cap t \bullet \\ M(s) & \text{sonst} \end{cases}$$

Dieser Vorgang kann auch verkürzt dargestellt werden als:  $M[t \triangleright M'$

mehrere Bedingungen / Stellen zu einer Stelle und mehrere Ereignisse / Transitionen zu einer Transition zusammengefasst werden. (Reisig 1986b), S. 118.

<sup>205</sup> Nicht existente Kanten können auch als Kanten mit Kantengewichtung = 0 angesehen werden.

(4)  $[M_0>$  ist die Menge der *Folgemarkierungen* zu  $M_0$ , die durch Schaltvorgänge erreicht werden. Wenn  $M_1 \in [M_0>$  and  $M_1[t>M_2$  für ein beliebiges  $t \in T$ , dann gilt auch  $M_2 \in [M_0>$ . Außerdem:  $M_0 \in [M_0>$ .

(5) Eine endliche *Markierungsfolge* (*Occurrence-Sequence*) von  $N$  kann angegeben werden als:

$$\sigma = M_0 t_1 M_1 t_2 \dots t_n M_n,$$

falls gilt:  $\forall i, 1 \leq i \leq n: M_{i-1}[t_i>M_i$ .

Eine unendliche Markierungsfolge als:

$$\sigma = M_0 t_1 M_1 t_2 \dots ,$$

falls  $\forall i, 1 \leq i: M_{i-1}[t_i>M_i$ .

(6) Eine Folge  $t_1 t_2 \dots$  von Transitionen ist eine *Schaltfolge* (*Transition-Sequence*) zu  $N$ , wenn eine Markierungsfolge  $M_0 t_1 M_1 t_2 \dots$  zu  $N$  existiert.

(7) Zwei Transitionen  $t_1$  und  $t_2$  sind *nebenläufig aktiviert* bei gegebener Markierung  $M$ , wenn

$$\forall s \in S: W(s, t_1) + W(s, t_2) \leq M(s) \leq K(s) - W(t_1, s) - W(t_2, s).$$

$s$  kann hier als Stelle im Vorbereich und als Stelle im Nachbereich interpretiert werden. Für eine Stelle im Nachbereich von  $t$  gilt, dass die Anzahl der durch die eingehenden Kanten hinzuzufügenden Marken addiert zu den Marken, die bereits in der Stelle liegen, maximal gleich der Kapazität der Stelle sein darf. Für eine Stelle im Vorbereich gilt, dass die ausgehenden Kanten maximal die vorhandenen Markierungen  $M(s)$  entnehmen dürfen.

(8) Zwei Transitionen stehen im *Konflikt*, falls beide aktiviert sind, aber nicht nebenläufig schalten können.

(9) Eine Transition  $t$  hat *Kontakt* in der Markierung  $M$ , wenn  $s \in S: W(s, t) \leq M(s), \exists s_0 \in S$  mit  $M(s_0) > \{K(s_0) - W(t, s)\}$ . In einem kontaktfreien Netz stellen die Kapazitäten keine Beschränkungen dar. Sie können weggelassen werden.

Auch S/T-Netze sind zur Beschreibung komplexer Systeme nur bedingt geeignet. Ein wesentlicher Nachteil besteht darin, dass Objekte nicht unterscheidbar sind. In den folgenden Kapiteln werden Höhere Petri-Netze beschrieben, die die Definition individueller Marken / Objekte erlauben.

### A.3.4 Petri-Netze im Software Engineering

Obwohl das Software Engineering, als problemorientierte Disziplin, und Petri-Netze, als lösungsorientierte Disziplin, unterschiedliche Blickwinkel in der Entwicklung von Software und Informationssystemen einnehmen, werden Petri-Netz-Varianten immer wieder zu Analysezwecken und zur Modellierung von Systemverhalten im Software Engineering eingesetzt (Denaro und Pezzè 2003), S. 440-441.<sup>206</sup> Ziele des Software-Engineerings bestehen darin, mit Hilfe eines geeigneten Vorgehens, den Entwurf für ein Softwareprodukt in einer Anwendungsdomäne handhabbar zu machen. Für eine Charakterisierung des Software-Engineerings als Forschungsfeld schlagen Denaro und Pezzè die in Abbildung 87 auf Seite 218 dargestellten Dimensionen „Produktentwicklung“, „Prozessunterstützung“ und „Anwendungsdomäne“ vor.<sup>207</sup>

Bei der *Produktentwicklung*<sup>208</sup> müssen unter Berücksichtigung der Anforderungen unterschiedlicher Stakeholder Systemaspekte, wie z. B. Beziehungen zwischen einzelnen Komponenten des Produkts, in einer Entwicklungsarchitektur spezifiziert werden. Darüber hinaus sollte die Entwicklung in Phasen (Anforderungsanalyse, Spezifikation usw.) und Aktivitäten (Analyse, Abstraktion, Modellierung, Dokumentation und Verfeinerung) eingeteilt werden.

Bei der Planung der Entwicklungsphasen sind geeignete Sprachen, Werkzeuge, Methodologien und Methoden auszuwählen, die den Stakeholdern eine ausreichende Unterstützung zur Umsetzung unterschiedlicher Aspekte des Systems während der Aktivitäten in den jeweiligen Phasen bieten. Die Auswahlentscheidungen werden in der Dimension *Prozessunterstützung* (Process Support) zusammengefasst. In den Phasen der Entwicklung kommen dabei Sprachen zum Einsatz, die sich sowohl hinsichtlich ihres Typs (operational, deskriptiv, ausführbar usw.) als auch ihres Stils (textuell, visuell, diagrammatisch, hybrid usw.) unterscheiden.

---

<sup>206</sup> Neben dem Reverse-Engineering, kommen Petri-Netze auch in anderen Teilbereichen des Software-Engineerings, z. B. beim Entwurf von Benutzerschnittstellen, bei der Analyse sicherheitskritischer Systeme (Cho u.a. 1996), bei der Modellierung von verteilten Systemen (Bass u.a. 1994) und im Software-Process-Management (Min u.a. 1997) zum Einsatz.

<sup>207</sup> Die Abbildung wurde um Literaturverweise ergänzt.

<sup>208</sup> Hier: Produkt = zu erstellende Software.

In der dritten Dimension wird eine Einteilung des Software-Engineerings entlang der *Anwendungsdomäne* vorgenommen. Die Domäne hat dabei Einfluss auf Anforderungen an die zu entwickelnde Software, die sich wiederum auf den Entwicklungsprozess auswirken. Die Entwicklung interaktiver, reaktiver, verteilter und mobiler Systeme macht die Anwendung unterschiedlicher Sprachen, Techniken, Werkzeuge, Aktivitäten, Spezialisten und Phasen erforderlich.

### ***Schrittweiser Architektorentwurf mit Petri-Netzen***

Petri-Netze bieten sich insbesondere für eine schrittweise Entwicklung von Architekturen und Systemspezifikationen an, da sie Techniken für die Verfeinerung, Partitionierung und Erweiterung von Modellen bereitstellen (Reisig 1986a); (Girault und Valk 2003), S. 105-11.

(Reisig 1986a) fasst sechs Prinzipien zusammen, die Petri-Netze als Entwicklungsmethode zur Spezifikation von Systemen im Software Engineering prädestinieren<sup>209</sup>:

*Prinzip zweier Elementtypen:* Obwohl unterschiedliche Petri-Netz-Typen zur Darstellung verschiedener Abstraktions- und Entwicklungsstufen im Software Engineering verwendet werden, liegt – sieht man von der Datenstruktur der zu verarbeitenden Elemente/Marken ab – allen Petri-Netz-Typen die gleiche Struktur in Form aktiver und passiver Elemente zugrunde. Mittels Petri-Netzen können *Systemkomponenten* beschrieben werden (Reisig und Rozenberg 1998), S. 5-6. Passive Komponenten können als Speicher, Schnittstellen<sup>210</sup> oder Kommunikationskanäle zwischen aktiven Komponenten betrachtet werden. „Angebotene“ und „benötigte Schnittstellen“ einer Komponente lassen sich durch die Kantenrichtung ermitteln.

*Prinzip der Verfeinerung und Vergrößerung:* Elemente eines Netzes (Stellen und Transitionen) können durch Einführung weiterer Petri-Netze verfeinert werden. Eine Transition, die eine Systemkomponente darstellt, kann durch ein transitionsberandetes Netz bzw. eine Stelle durch ein stellenberandetes Teilnetz verfeinert werden. „Eine Verfeinerung eines Netzes bedeutet eine Konkretisierung, eine Detaillierung der Modellierung, und zwar (im Gegensatz

---

<sup>209</sup> Begrifflichkeit angepasst nach (Baumgarten 1996).

<sup>210</sup> „Eine Schnittstelle definiert einen Vertrag, der für den Fall des ‚vertragsgemäßen‘ Aufrufs die ebenfalls ‚vertragsgerechte‘ Ausführung einer bestimmten Aktion garantiert.“ (Reussner und Hasselbring 2006), S. 44.

zur Einbettung [...] bezüglich der inneren Mechanismen von Zuständen oder Ereignissen.“ (Baumgarten 1996), S. 62. Umgekehrt kann bei Bedarf ein Netz, z. B. zur übersichtlicheren Gestaltung, durch Streichung nicht relevanter Komponenten, vereinfacht werden.

*Prinzip der Einbettung und Restriktion:* Ein bestehendes Netz wird durch Hinzufügen von Kanten und Knoten erweitert. "Eine Einbettung bedeutet für das Netz als Modell eines realen Systems die Ergänzung um weitere Aspekte und Systemteile, eine Vervollständigung, eine Einbeziehung von Teilen der Umgebung." (Baumgarten 1996), S. 63-64.<sup>211</sup>

*Prinzip der schrittweisen Einführung von Systemdynamik:* Ausgehend von der Trennung passiver und aktiver *Komponenten* in Kanal/Instanzen-Netzen können Netze weiterentwickelt werden, indem dynamisches Verhalten durch Hinzufügen von nicht unterscheidbaren und schließlich unterscheidbaren Marken in Netzsysteme integriert wird. (Übergang von Low-Level- zu High-Level-Netzen)

*Prinzip der Lokalität:* Das Verhalten von Netzkomponenten wirkt sich nur unmittelbar auf die direkte Umgebung aus. So bewirkt das Schalten einer Transition lediglich eine Änderung der Markierung im direkten Vor- und Nachbereich, nicht jedoch in "weiter entfernten" Netzteilen.

*Prinzip des nichtdeterministischen Verhaltens:* Alternativen können im Systemdesign beschrieben werden; das Entscheidungsverhalten wird nicht vorgegeben.

Nach Valk in (Girault und Valk 2003), S. 9-28 zeichnen sich Petri-Netze durch ein weiteres „Prinzip“ aus, das in Zusammenhang mit (6) zu betrachten ist:

*Prinzip der Nebenläufigkeit:* Transitionen, die eine unterschiedliche Umgebung aufweisen, können nebenläufig auftreten.

### ***Eignung von Petri-Netzen als Entwurfsmethode und Kritik***

In Abbildung 88 wird die Eignung von Petri-Netzen als Entwicklungsmethode entlang der in Abbildung 87 von (Denaro und Pezzè 2003) beschriebenen Teilaspekte und Phasen des ingenieurmäßigen Entwurfs von Softwareprodukten untersucht. Dabei ist intuitiv durch die Anzahl der Literaturstellen ersichtlich, dass Petri-Netze gerade in den Phasen der Systemanalyse (s. unter *Development*

---

<sup>211</sup> Zwecks Vollständigkeit sollten außerdem „Faltung“ und „Entfaltung“ genannt werden. Vgl. z. B. (Baumgarten 1996), S. 66-69.

*Phases*) besondere Stärken aufweisen. Neben der Simulation durch *Markenspiel* und *Erreichbarkeitsanalysen* fällt hierunter z. B. die algebraische Analyse von Netzen (S-Invarianten, T-Invarianten etc.) Zudem sind Petri-Netze vor allem zur Darstellung von *Systemaspekten* (System Aspects) geeignet. Dynamische Strukturen eines Systems, im Sinne der weiter oben beschriebenen Prozessmodelle, lassen sich durch die Einführung von Netzen abbilden sowie Funktionen bzw. Komponenten durch Transitionen modellieren.

Obwohl grafische Artefakte für Petri-Netze durch die Anwender leicht erfasst werden können, setzt die Modellierung vor allem von High-Level-Petri-Netzen ein mathematisch-formales Grundverständnis (Logik, Mengenlehre, Graphentheorie und Algebra) voraus. Während anzunehmen ist, dass die in Abbildung 88 genannten Stakeholder der Gruppen Analysten, Architekten, Programmierer und Test-Designer mit formalen Methoden vertraut sind, müssen Endanwender durch geeignete Anleitung mit Petri-Netzen als Sprache zur Dokumentation vertraut gemacht werden.<sup>212</sup>

Um die Gesamtsicht eines Systems im Sinne einer Architektur darzustellen, werden neben der Darstellung dynamischer Aspekte weitere Sprachen benötigt, die die Abbildung statischer Strukturen unterstützen. Obwohl Petri-Netze die Manipulation von Datenobjekten beschreiben, werden in der Regel keine grafischen Modelle bereitgestellt, um statische Datenstrukturen abzubilden. Eine Ausnahme bilden bspw. XML-Netze. Im Falle relationaler Datenstrukturen können z. B. das Entity-Relationship-Modell nach (Chen 1977) bzw. dessen Erweiterungen zur Modellierung verwendet werden.<sup>213</sup>

---

<sup>212</sup> (Reisig 1986a) vertritt eine gegenteilige Auffassung.

<sup>213</sup> Streng genommen muss das ER-Modell weiter in ein Relationenmodell / in Relationen überführt werden.



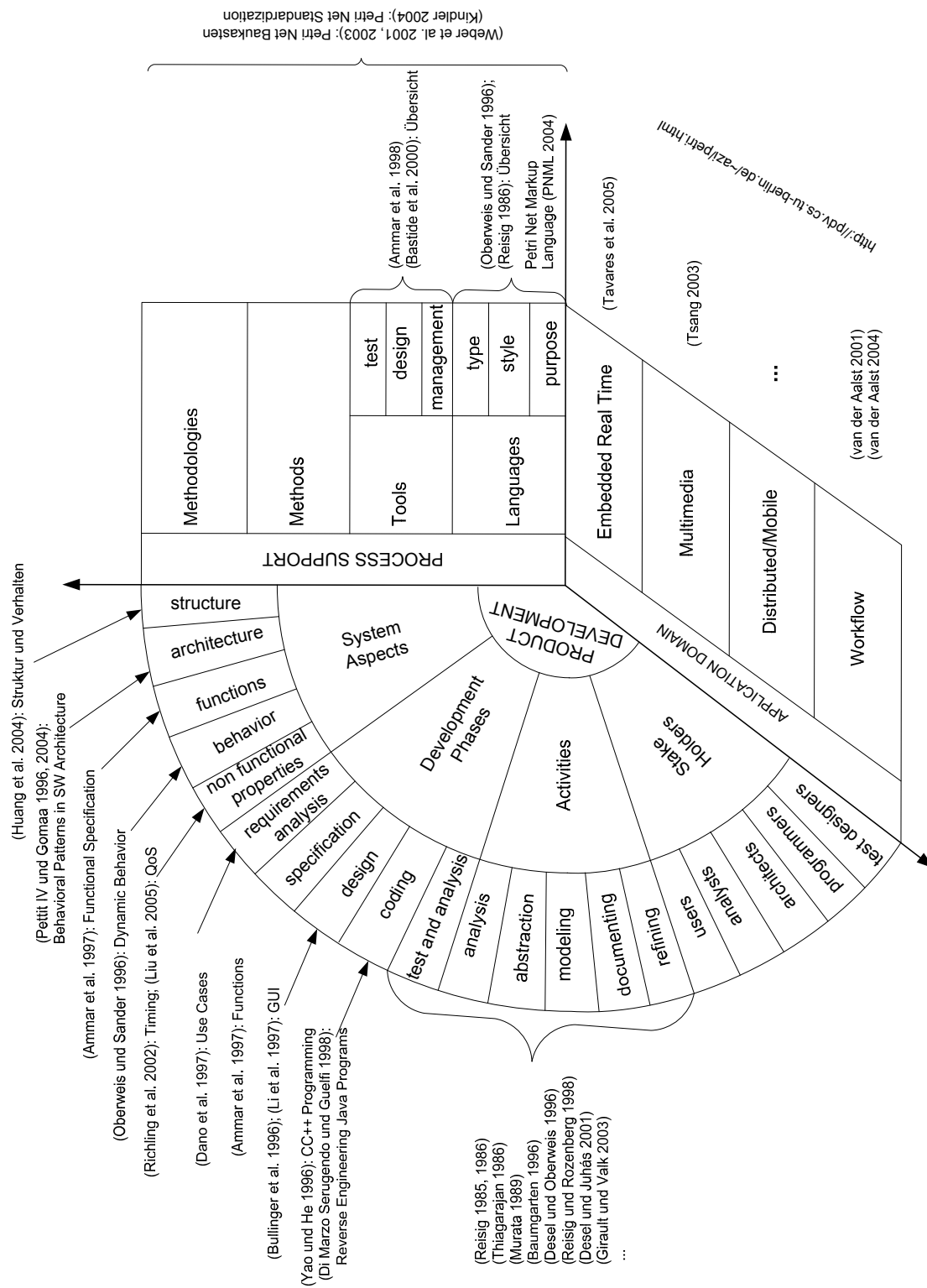
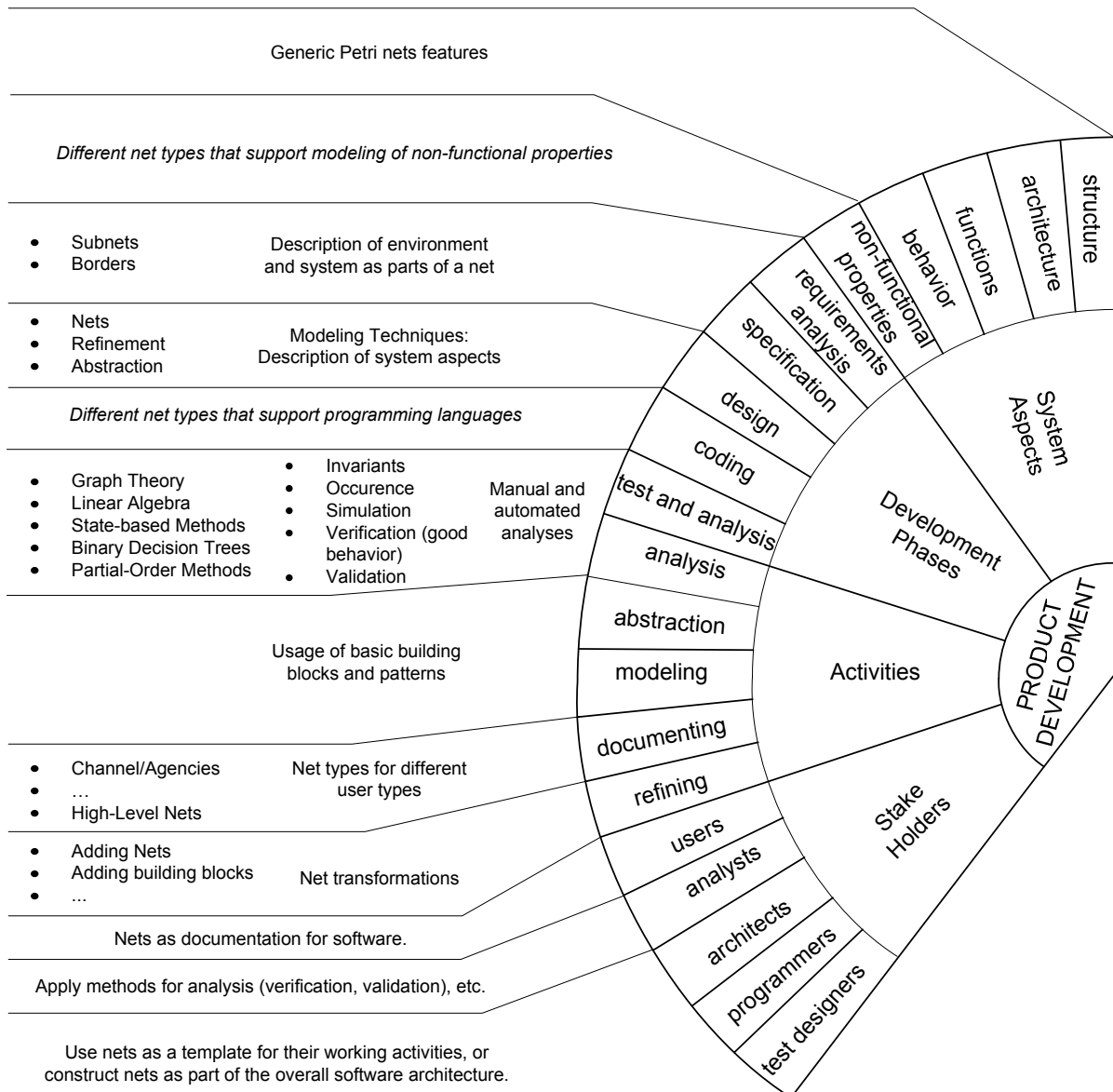


Abbildung 87: Dimensionen des Software-Engineerings und ausgewählte Literatur<sup>214</sup>



**Abbildung 88: Entwicklung von Software-Systemen mit Petri-Netzen – Einordnung von Petri-Netz-Konzepten**<sup>215</sup>

### A.3.5 CPN ML

CPN ML ist eine Programmiersprache, die an der Universität Aarhus entwickelt wurde und bei der Erstellung von CP-Netzen mit Hilfe des Werkzeugs CPN Tool zur Annotation von CP-Netzen verwendet wird. CPN ML erweitert Standard ML. Standard ML ist eine (überwiegend) funktionale Programmiersprache, die neben einer formalen Fundierung zusätzlich durch das Typsystem und durch die

<sup>214</sup> Quelle: modifiziert nach (Denaro und Pezzè 2003), S. 442.

<sup>215</sup> Quelle: modifiziert nach (Denaro und Pezzè 2003), S. 442.

Einführung von Modulen eine flexible und erweiterbare Syntax bereitstellt (Hudak 1989; Goldberg 1996). Die Verwendung einer funktionalen Programmiersprache gestattet dem Anwender, beliebig komplexe Funktionen für Netz-Inschriften, z. B. rekursive Funktionen zur Verarbeitung von Listen, zu definieren. Erweiterungen von Standard ML in CPN ML zielen darauf ab, die Anwenderfreundlichkeit und die Anwendbarkeit zu erhöhen. Neben der Erweiterung um Colour-Sets lässt CPN ML die Verwendung *getypter Variablen* zu. *Referenzvariablen* sind nur in Codeblöcken erlaubt.<sup>216,217,218</sup>

### **Standard ML**

Standard ML geht auf die Programmiersprache ML zurück, die zur Steuerung eines Systems namens LCF, das das Reasoning über rekursive Funktionen erlaubt, verwendet wurde (Milner 1984). Standard ML ist eine *funktionale Programmiersprache* mit Erweiterungen um Referenzen, Funktionen für Input und Output, ein Modulsystem und Ausnahme-Behandlungsmechanismen. Standard ML zeichnet sich auch dadurch aus, dass Datentypen existieren, die für die Darstellung von Relationen im Datenbanksinne geeignet sind und dass der Quellcode portabel ist, da Entwicklungs- und Laufzeitumgebungen für eine Vielzahl unterschiedlicher Betriebssysteme verfügbar sind. Für die professionelle Entwicklung von Programmen ist vor allem die Erweiterung des funktionalen Programmierparadigmas um ein eigenes Typsystem wichtig, das über folgende Eigenschaften verfügt:

- Standard ML ist „statically typed“, d.h. jedem Ausdruck in Standard ML wird ein Typ zugewiesen, der beschreibt, in welchem Bereich ein Rückgabewert nach Auswertung des Ausdrucks liegen darf. Die Überprüfung der Kompatibilität von Datentypen findet zur Compilezeit statt. Dieses Vorgehen soll gewährleisten, dass Fehler, die durch die Auswahl inkompatibler Datentypen entstehen können, bereits in frühen Entwicklungsphasen vermieden werden (Gilmore 1997).
- Neben einfachen Datentypen für die Abbildung von Integer-Zahlen, Realzahlen und Zeichenketten, können auch Listen und Records (indizierte

---

<sup>216</sup> <http://www.daimi.au.dk/designCPN/sml/why.html>

<sup>217</sup> <http://www.daimi.au.dk/designCPN/sml/choice.html>

<sup>218</sup> <http://www.daimi.au.dk/designCPN/sml/cpnml.html>

Listen) verwendet sowie eigene konkrete und abstrakte Datentypen durch den Anwender definiert werden.

- Zur Ermittlung des Typs eines Ausdrucks werden Typreferenzen statt der expliziten Definition des Wertebereichs verwendet (Hudak 1989).
- Referenzen, als Zeiger auf Speicherzellen, werden mit der Funktion `ref` definiert.
- Funktionen und Datenstrukturen können *polymorph* sein. So kann eine Infix-Funktion für die Addition (z. B.  $5 + 5$ ) neben der Addition von zwei Integer-Zahlen auch die Addition von zwei Realzahlen (z. B.  $1.34 + 2.78$ ) zulassen. Funktionen können Argumente eines beliebigen Typs aufnehmen, bspw. auch weitere Funktionen.
- In Standard ML können Funktionen als Daten behandelt werden. Sie können als Argumente an Funktionen übergeben werden und selbst Rückgabewert einer Funktion sein (Hughes 1989).

Neben den obigen Eigenschaften, die in der so genannten *Core-Language* verankert sind, umfasst Standard ML ein Modul-System.<sup>219</sup>

- Eine `structure` kann selbst Werte, Typen und Ausnahmen der Core-Language enthalten.
- Eine `signature` definiert einen Typ für eine `structure`.
- Ein `functor` definiert eine Funktion über eine `structure`.

Für weiterreichende Informationen zur Programmierung mit Standard ML sei u. a. auf (Milner u.a. 1997; Gansner und Reppy 2004; Harper 2005) verwiesen.

---

<sup>219</sup> Zu den Modulen in Standard ML gehören `structures`, `signatures` und `functors`, die in etwa Werten, Typen und Funktionen entsprechen (Tofte 1993).

## A.4 Abrechnungssysteme – Begriffsdefinitionen

Begriff	Definition
Accounting	<ol style="list-style-type: none"> <li>1. Aggregiert im Collecting gesammelte Ressourcennutzungsdaten und Charging-Informationen. Sie speichert die Daten in einer (zentralen) Accounting-Datenbank, die in der Regel in User- und Session-Records organisiert ist und für Charging, Billing, Auditing und die Bereitstellung von Kunden- sowie Steuerungsinformationen für das Management (Trendanalysen, Kostenverteilung, Auditing) und für das Billing verwendet werden kann. Das Accounting-Management setzt voraus, dass die Ressourcennutzung gemessen, gerated und Objekten zugewiesen werden kann.</li> <li>2. Die Aktivität der Aggregation von Informationen über die durchgeführten Operationen eines Systems.</li> <li>3. Accounting bestimmt die Art der Sammlung von Informationen zu der Dienstenutzung eines Kunden, die sich im Ressourcenverbrauch oder dem Konsum niederschlägt. Accounting definiert daher eine Funktion, die die Nutzung in technische Messgrößen überführt. Die zu sammelnden Informationen werden durch eine Menge von Parametern, die in Accounting-Records enthalten sind, bestimmt. Dieser Record hängt ab von (1) der Netzwerkinfrastruktur, die den Dienst unterstützt, (2) dem Protokoll. Der Inhalt eines Accounting-Records ist technischer Art. Dieser Record bildet die Basis für das Charging und Billing.</li> <li>4. Die Accounting-Ebene konsolidiert die gesammelten Informationen aus der Collecting-Ebene, entweder innerhalb der Domäne eines einzigen Providers oder von Domänen anderer Provider. In der Accounting-Ebene werden zudem Netzwerk-Accounting-Data-Sets oder – Records generiert, die an die Charging-Ebene zwecks Preisbestimmung weiter geleitet werden.</li> </ol>
Accounting-Datenbank	→Accounting
Accounting-Attribute	Messgrößen, die für Accountingzwecke zur Verfügung stehen.
Accounting-Datensammlung	Alle Aktivitäten, die benötigt werden, um Verbrauchsdaten zu messen, sammeln (collect) und an das Accounting weiterzuleiten.

Begriff	Definition
Accounting-Protokoll	Ein Protokoll, das den Austausch von Daten für Accountingzwecke unterstützt.
Accounting-Server	Ein Server der Ressourcen- und Dienste-Verbrauchsdaten von den Collectors empfängt, diese in Session Records übersetzt und diese Daten in einer Accounting-Datenbank sammelt. Der Accounting-Server steuert außerdem die Weiterleitung von Session Records an beteiligte Akteure.
Administrative Domäne	Gruppe von Hosts, Routern und Netzwerken, die durch eine einzelne Organisation betrieben werden. Eine administrative Domäne verfügt in der Regel über einen Accounting-Server und eine Accounting-Datenbank.
Accounting-Management	Eine von fünf Kategorien des Netzwerk-Managements nach ISO. Accounting-Managementsysteme sind dafür verantwortlich, Netzwerkdaten zu sammeln, die mit der Ressourcennutzung einhergehen.
Archival-Accounting	Dient der Sammlung von Accounting-Daten, um fehlende Einträge im Falle eines Datenverlustes zu rekonstruieren und Daten über einen vorgeschriebenen Zeitraum zu sammeln. Rechtliche Anforderungen oder Anforderungen aus der Buchhaltung schreiben die Rahmenbedingungen der Datensicherung, z.B. Geheimhaltungspflichten und Aufbewahrungsfristen, vor.
Auditing	<ol style="list-style-type: none"> <li data-bbox="596 1243 1294 1547">1. Reviews und Untersuchung von System Records sowie Aktivitäten, um die Genauigkeit der System Controls zu prüfen, die Compliance mit Policies und Vorgangsbeschreibungen zu überwachen, Sicherheitsverletzungen festzustellen und Änderungen in der Kontrolle, den Policies und in den Vorgehensweisen vorzuschlagen und einzuleiten.</li> <li data-bbox="596 1559 1294 1718">2. Aktivität zur Überprüfung der Korrektheit, z.B. einer Rechnung durch einen Service-Provider oder die Überprüfung, dass der Verbrauch einer Usage-Policy entspricht.</li> </ol>

Begriff	Definition
Billing	<ol style="list-style-type: none"> <li>1. Aktivität zur Erstellung einer Rechnung. Die Spezifikation einer Rechnung kann erforderlich sein. Notwendige Eingangsdaten für die Erstellung sind die Payment-Methode, z.B. Barzahlung, Zahlung per Scheck, Micropayment etc. Die eigentliche Zahlung, d.h. der Transfer von Währungseinheiten, ist nicht Teil des Billing-Prozesses. Hinweis: Billing behandelt nicht die Ermittlung des Rechnungsbetrages, den ein Konsument zu zahlen hat, sondern die Spezifikation einer Rechnung und die Auswahl einer Zahlungsmethode.</li> <li>2. Billing beschreibt den Prozess der Aufbereitung gesammelter Charging-Informationen zu einem Kunden in Form einer Rechnung. Es beinhaltet den Prozess der Auflistung aller Charging-Informationen für einen Kunden, die in den Charging-Records, die über eine bestimmte Zeit gesammelt wurden, enthalten ist. Die Rechnung kann außerdem Auskunft über die Zahlungsmethode und -modalitäten ermöglichen.</li> </ol>
Billing-Server	→ Billing-System
Billing-Street	Die Billing-Street umfasst alle Aktivitäten, die vom Ressourcenverbrauch bis zur Bezahlung reichen: Accounting-Datensammlung (Metering, Collecting), Accounting, Charging, (Charge-Distribution) und Billing.
Billing-System	Ein Informationssystem, das die Rechnungsvorbereitung (Bill-Preparation) und die Präsentation der Rechnung (Bill-Presentation) auf Basis der Accounting- und Charging-Informationen unterstützt.
Kapazitätsanalyse	Analyse der Effektivität des Ressourcenverbrauchs, der Bestimmung von Engpässen usw.

Begriff	Definition
Charging	<ol style="list-style-type: none"> <li>1. Ermittlung eines Geldbetrages der für Benutzer für eine Sitzung anfällt, basierend auf Pricing- und Accounting-Records.</li> <li>2. Charging ermittelt als Prozess die Kosten einer Ressource durch Verwendung eines bestehenden Accounting-Records, der für eine spezifische Ressourcennutzung steht. Accounting kann also als Funktion betrachtet werden, die technische Größen in Geldeinheiten überführt.</li> <li>3. Eine Aktivität zur Ermittlung von Session-Charges ausgehend von dienstespezifischen Charging- und Pricing-Schemata, die in der Charging-Policy definiert sind.</li> </ol>
Charge-Distribution	Benutzerzuordnung zu Belastungsbeträgen (Charges) und Aufteilung der Charges über die zugewiesenen Benutzer.
Collecting	Prozess des Zugriffs (auf die Daten) unterschiedlicher Metering-Entities auf Verbrauchsdaten und deren Inbezugsetzung zu Ressourcenverbäuchen durch Übernahme in Data-Records. Der Zugriff auf Metering-Daten findet in regelmäßigen Zeitintervallen statt. Es handelt sich nicht um eine kontinuierliche Aktivität. Die Collecting-Ebene leitet Informationen an das Accounting weiter.
Collector (Synonym: Meter-Reader)	Ein Instrument, um selektiv gemessene Daten zu sammeln, diese temporär zu speichern und an die Accounting-Datenbank weiterzuleiten.
Content-basiertes Billing	Billing, bei dem Content oder Metadaten, die diesen Content beschreiben, zur Bestimmung eines Preises beitragen.
Cost-Recovery (Kostendeckung)	Der Prozess der Erzielung von Erlösen, um die Kosten der Erstellung, des Deployments und des Betriebs eines Produktes oder Dienstes zu decken.
Dynamic-Pricing	Beim dynamischen Pricing, variiert der Preis, der für eine Ressourcennutzung oder die Inanspruchnahme von Diensten zu zahlen ist, z.B. abhängig von der Nachfrage oder durch Auktionierung.
Electronic-Bill-Presentation und -Payment (EBPP)	Elektronische Vorlage der Kundenrechnungen. Enthält Mechanismen, die dem Kunden die Begleichung der Rechnung ermöglichen.



Begriff	Definition
Interims-Accounting	Ein Interim-Accounting-Paket liefert einen Snapshot der Verbräuche während einer Sitzung. Wichtig dann, wenn ein Gerät neu gestartet werden muss oder Netzwerkprobleme aufgetreten sind, die den Empfang oder die Generierung eines Session-Records verhindern.
Intra-Domain-Accounting	Sammlung und Speicherung von Informationen über den Ressourcenverbrauch innerhalb einer Administrativen Domäne zur Verwendung in dieser Domäne. Im Intra-Domain-Accounting, überschreiten Accounting-Packets und Session-Records typischerweise nicht die „administrativen Grenzen“.
IP-Billing-System	Billing-System, das zum Billing paketbasierter Dienste (inkl. IP-Telefonie, Datenpakete, Voice, Video, Spiele etc.) eingesetzt wird.
Kostenallokation	Aktivität der Zuweisung von Kosten zu Entities. Kostenallokation und Rating sind unterschiedliche Prozesse. Bei der Kostenallokation liegt das Ziel üblicherweise darin, bekannte Kosten zwischen unterschiedlichen Entities aufzuteilen. Beim Rating liegt der Fokus darauf, den geschuldeten Betrag zu ermitteln.
Meter	Ein Gerät, um Attribute im Netz auszuwerten bzw. zu zählen (z.B. Verbindungszeiten oder Anzahl der Pakete/übertragener Datenmengen in Kilobytes). Meter klassifizieren die gemessenen Daten nach Accountable-Entities (zurechenbare Entitäten) unter Verwendung weiterer Attribute, z.B. Quellen- und Zieladresse.
Meter-Reader	→ Collector
Metering	<ol style="list-style-type: none"> <li>1. Prozess der Sammlung von Verbrauchsdaten und der Speicherung dieser Daten durch Software- und / oder Hardware-Komponenten.</li> <li>2. Alle Aktivitäten, die dazu dienen, die Ressourcennutzung zum Zwecke des Accounting und der Kontrolle der Nutzungsdaten zu monitoren. (Usage-Metering)</li> </ol>
Metering-Hierarchie	Hierarchie von Metern und Collectors, i.d.R. innerhalb einer administrativen Domäne. Wird zur Generierung von Accounting-Daten verwendet.
Micropayment	Eine "Low-Value"-Transaktion, die den Einsatz von Kartenzahlung nicht rechtfertigt, da die Verwaltungskosten dann i. d. R. den Abrechnungsbetrag übersteigen.
Pay-Before, Prepaid, Debit-Payment	Der Nutzer zahlt vorab einen Betrag und kann Ressourcen solange nutzen, bis der Betrag verbraucht ist.

Begriff	Definition
Pay-Later, Postpaid, Credit-Payment	Traditionelle Bezahlmethode in der Telekommunikationsindustrie. Abrechnung z.B. im Nachhinein am Ende eines Monats.
Pay-Now, Real-Time-Payment	Der Nutzer muss die Nutzung einer Ressource sofort (innerhalb eines Zeitfensters) bezahlen. Erfordert Real-Time-Accounting und -Charging sowie Micropayment.
Payment	Zahlung, Wertübertragung. Meist mit der Übertragung von Beträgen zwischen Konten (Kreditkarte / Giro) verbunden.
Payment-Instrument / Zahlungsinstrument	Ein Instrument, das benötigt wird, um eine Zahlung zu initiieren (Kontokarte, Kreditkarte etc.).
Payment-Method / Zahlungsmethode	Methode, die angewendet wird, um den Zahlungsprozess zu vereinfachen. Umfasst Zahlungsinstrumente und Payment-Systems.
Payment-Protocol / Zahlungsprotokoll	Beschreibt eine Serie von Transaktionen, an deren Ende eine Zahlung als abgeschlossen gilt. In der Regel wird die Zahlung durch ein Token eines unabhängigen Dritten gesichert.
Payment-System	Ein System oder Netzwerk, das zur Verarbeitung von Zahlungen Verwendung findet, z.B. Clearingstellen der Kreditinstitute.
Pricing	<ol style="list-style-type: none"> <li>1. Bestimmung eines Preises für Ressourcennutzung, ein Produkt, den Abruf von Informationen oder Funktionen eines Dienstes. Kann zum Rating in Beziehung gesetzt werden, ist aber nicht unbedingt erforderlich. Das Pricing wird durch Wettbewerb, Geschäftsstrategien und –modelle sowie Erlösmodelle beeinflusst.</li> <li>2. Pricing ist der Prozess der Preissetzung für einen Dienst, ein Produkt oder Content. Dieser Prozess ist ein integraler Bestandteil des und damit kritisch für das Business. Pricing steht in enger Beziehung zum Marketing. Preise können ausgehend von Kosten/Gewinn-Verhältnissen oder von der Marktsituation des Unternehmens bestimmt werden.</li> </ol>
Pricing-Process	Alle Regeln und Prozeduren, die zur Festlegung und Implementierung von Preisen dienen. Dazu gehören: Modelle, Entscheidungsregeln, Organisationsstrukturen, Verantwortungsbereiche, Anreize, Timing, IT-Unterstützung, Vorgehensmodelle (Analyse, Entscheidung, Implementierung und Monitoring), objektive Daten und subjektive Erfahrungen. (vgl. (Simon u.a. 2003))

Begriff	Definition
Pricing-Table	Eine Datenstruktur mit statischen Informationen über den/ die abzurechnenden Preise(e), die für die Ressourcennutzung anfallen. Oft aufgeteilt nach der Art der Ressourcennutzung (Resource-Type), nach Kunden (Customer-Type) oder nach Tageszeiten.
Rating	<ol style="list-style-type: none"> <li>1. Aktivität zur Bestimmung der Kosten der Ressourcennutzung für ein Unternehmen.</li> <li>2. Bestimmung eines Preises zu einer Ressourcennutzung..</li> </ol>
Real-Time-Accounting	Real-Time-Accounting verlangt die Verarbeitung von Nutzungsdaten in definierten Zeitfenstern, so dass ein Real-Time-Charging durchgeführt werden kann.
Real-Time-Charging	Berechnung des Geldbetrages, der innerhalb eines definierten Zeitfensters geschuldet wird. Setzt Real-Time-Accounting voraus und ist selbst Voraussetzung für Real-Time-Payment. In Kombination mit Real-Time-Payment, dient die Einführung von Zeitrestriktionen zur Minimierung des finanziellen Risikos.
Session / Sitzung	<ol style="list-style-type: none"> <li>1. Eine logische Verbindung zwischen einem Nutzer und einem Service Provider, die zur Kommunikation aufgebaut wird.</li> <li>2. Eine Menge von in Beziehung zueinander stehenden Kommunikationstransaktionen zwischen zwei oder mehreren Netzwerk-Devices.</li> </ol>
Static-Pricing	Im Falle des Static-Pricing wird der Preis für die Ressourcennutzung oder die Dienstenutzung vorab festgelegt und für eine bestimmte Zeit fixiert.
Trendanalyse	Analyse der Änderung, z.B. der Wachstumsrate einer Ressourcennutzung über eine Zeitspanne.

Die Definitionen sind, soweit nicht anders vermerkt, modifiziert übernommen aus folgenden Quellen: (Blake 1998), (Hille u.a. 2000); (Hartanto und Carle 1999)

## A.5: Industrielle Abrechnungssysteme

Die folgenden Tabellen geben einen Überblick über die Funktionalität industrieller Abrechnungssysteme und deren Anbieter. Neben Nischenanbietern wie die Firma Soprano werden auch Produzenten von Abrechnungssystemen aufgeführt, die die komplette Wertschöpfungskette in unterschiedlichen Dienstleistungsbereichen, vor allem der Telekommunikation, abdecken.

<b>Firma</b>	<b>Amdocs</b>
URL	<a href="http://www.amdocs.com">http://www.amdocs.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Finanzdienstleistungen</li> <li>• Satelliten-Dienste</li> <li>• Service-Industrien</li> </ul>
Produkte	<p>OSS mit folgenden Funktionen / Komponenten:</p> <ul style="list-style-type: none"> <li>• Billing-System: Charging, Collecting, Dokumentenentwurf, Accounts Receivable, Fraud-Management</li> <li>• Service-Mediation</li> <li>• Risk-Management: Zahlungsrisiko, Revenue-Assurance</li> <li>• Fulfillment: Service-Aktivierung, Inventory-Management</li> <li>• Order-Management</li> <li>• Self-Care-System</li> </ul>

<b>Firma</b>	<b>Convergys</b>
URL	<a href="http://www.convergys.com">http://www.convergys.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Fernsehen (Pay-TV)</li> </ul>
Produkte	<p>OSS mit folgenden Funktionen / Komponenten:</p> <ul style="list-style-type: none"> <li>• Billing-System: Pricing und Invoicing, Unterstützung unterschiedlicher „Geschäftsmodelle“</li> <li>• Real-Time-Authorisierung, Rating und Pricing von Transaktionen</li> <li>• Revenue-Management-System</li> <li>• Self-Care-System zur Verwaltung von Konten über das Internet</li> <li>• Katalog-Management</li> <li>• Unterschiedliche Preis- und Produktkataloge für Branded-Services, Konfiguration neuer Produkte</li> <li>• Unterstützung unterschiedlicher Netzwerktypen</li> </ul>

<b>Firma</b>	<b>Danet Group</b>
URL	<a href="http://www.danet.com">http://www.danet.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Finanzdienstleistungen</li> <li>• Automobilindustrie (Hersteller, Zulieferer, Dienstleister, Händler, Werkstätten)</li> <li>• Transport und Logistik (Logistik-Konzerne, Inhouse-Logistiker, Kontrakt-Logistiker)</li> </ul>
Produkte	<p>Die Firma danet vertreibt Produkte in den Bereichen</p> <ul style="list-style-type: none"> <li>• Inter-Carrier-Billing: Lösung für Invoicing, Settlement und Reconciliation von CDRs</li> <li>• Partner-Billing und Settlement: Übernahme von Kunden-Abrechnungen für Partner</li> <li>• Content-Solutions: Plattformen zur Verbreitung von Content-Angeboten über unterschiedliche Vertriebskanäle verbreiten inkl. notwendiger Abrechnungsstruktur</li> <li>• Abrechnung transaktionsbasierter Services, Mobile-Payment, verbrauchsabhängiges Billing und Payment</li> <li>• Analysefunktionen: Analyse des Anwenderverhaltens, Cross-Selling-Konzepte und Lieferanten-Rating, Analyse der Akzeptanz von Marketing-Aktionen und Internet-Auftritt</li> <li>• Kredit-Management: Bonitätsprüfung und Mahnwesen</li> <li>• Revenue-Assurance</li> <li>• Data-Collection und -Rating: Management von komplexen Datenquellen und Entwicklung von Kennzahlen</li> <li>• System-Matching: Testautomationssysteme, Performance-Analyse, Content- und Datenstrom-Filterung, Least-Cost-Routing</li> </ul>

<b>Firma</b>	<b>High Deal</b>
URL	<a href="http://www.highdeal.com">http://www.highdeal.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Abrechnung von Online- und Mobile-Application-Services</li> </ul>
Produkte	<p>Produkte in den Bereichen</p> <ul style="list-style-type: none"> <li>• Performance-Rating und Settlement</li> <li>• Einführung von Pricing-Strategien in dynamischen Märkten</li> <li>• Software zum Pricing und Rating sehr großer Transaktionszahlen, konvergentes Billing und Settlement</li> </ul>

<b>Firma</b>	<b>Infodirections</b>
URL	<a href="http://www.infodirections.com">http://www.infodirections.com</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Satellitendienste</li> </ul>
Produkte	<p>Produkte in den Bereichen</p> <ul style="list-style-type: none"> <li>• Konvergente Operations- und Billing-Software für Kabelnetzanbieter, die bei der Erzeugung und dem Verkauf von Produkten, bei der Interaktion mit Partnern und Kunden sowie beim Billing und beim Forderungseinzug helfen Als ASP-Lösung oder als eigenständiges Softwareprodukt</li> <li>• OSS-Systeme für Mobilfunk- und Satellitendiensteanbieter</li> <li>• Point-of-Sale-Systeme für Retailing im Telekommunikations-Bereich. Real-time Datenaustausch mit Vielzahl von Filialgeschäften bei zentraler Verwaltung des Systems</li> <li>• Online-Billing: EBPP-Lösung im Internet, die sich in OSS und andere Systeme der Firma integrieren lässt</li> <li>• Software zur Verwaltung unterschiedlicher "Channel Organizations"</li> </ul>

<b>Firma</b>	<b>Intec</b>
URL	<a href="http://www.intec-telecom-systems.com/">http://www.intec-telecom-systems.com/</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Internet-Service-Provider</li> </ul>
Produkte	<p>Produkte u. a. in den Bereichen</p> <ul style="list-style-type: none"> <li>• OSS-/BSS-Systeme für konvergentes Billing, Mediation, Aktivierung und Real-time-Charging, Retail-Billing, Mediation, Routing, Interconnection-Billing, Billing für Carrier-Access, Dienstaktivierung und Content-Partner-Management</li> <li>• Services-Management: Multi-Service-Mediation, IP-Charging, Service-Aktivierung, Konvergentes Charging</li> <li>• Business-Management: Überwachung von Vereinbarungen und Settlement für die Dienstenutzung im Bereich Multi-Media-Downloads (Musik, Klingeltöne, Clips etc.), Ticketing- und Voucher-Services, Micro-Transactions und Zahlungen im M-/E-Commerce, Mehrwert-Sprachdienste (Push to Talk, Videokonferenzen), Mobiles Fernsehen, Online-Infotainment (News, Wetter, Navigation), Online-Gaming, Voice-over-IP, Premium-MMS-Content, Content-Sponsorship, Werbe-Dienstleistungen, Netzwerküberwachung, Content-Aggregation und -Brokering</li> </ul>

<b>Firma</b>	<b>LavaStorm</b>
URL	<a href="http://www.lavastormtech.com">http://www.lavastormtech.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Online-Dienste</li> </ul>
Produkte	<p>Produkte in den Bereichen</p> <ul style="list-style-type: none"> <li>• Business-Assurance: Aufspüren und Recovery versteckter Erlöse, Kosten und „Kapitalmöglichkeiten“ bei TK-Providern durch Analyse der Systemlandschaft. Dabei Aufbau „analytischer Brücken“ zwischen Self-Care-Systemen, Sprach- und Datennetzen sowie Billing-Systemen im Einzel- und Großhandel</li> <li>• Datensammlung über OSS- und BSS-Systeme eines Providers entlang vordefinierter Business-Rules. Bereitstellung der Daten</li> <li>• Ausnahmeverfolgung bei Abrechnungsdaten</li> <li>• Definition von Workflows in einer Business-Rules-Architektur</li> </ul>

<b>Firma</b>	<b>Lucent Technologies</b>
URL	<a href="http://www.lucent.com">http://www.lucent.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Satellitendienste</li> <li>• Internet-Service-Provider</li> <li>• Energie- und Versorgungsunternehmen</li> </ul>
Produkte	<p>Produkte in den Bereichen</p> <ul style="list-style-type: none"> <li>• Charging, Rating, Discounting von Kundenverbräuchen</li> <li>• EBPP durch Definition von Financial Workflows</li> <li>• Customer-Self-Care über Web-Applikationen</li> <li>• Einführungsunterstützung für Billing und Customer-Self-Care-Anwendungen für Next-Generation-IP-Netzwerke und Mobilfunkdienste</li> </ul>

<b>Firma</b>	<b>Metasolv</b>
URL	<a href="http://www.metasolv.com">http://www.metasolv.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> </ul>
Produkte	<p>Produkte, die den „order-to-activate service provisioning process“ automatisieren sollen.</p> <ul style="list-style-type: none"> <li>• OSS</li> <li>• Order-Management: Rückverfolgung aller Auslieferungsschritte, die notwendig sind, um Kommunikationsdienste zu erbringen</li> <li>• Inventory-Management: Management aller Ressourcen, die zur Erbringung eines Kommunikationsdienstes erforderlich sind, z. B. physische, logische, technische Ressourcen oder dazu eingesetzte Dienste</li> <li>• Dienstaktivierung für ATM-Dienste, Frame-Relay-Verbindungen, IP-VPNs, optische Verbindungen, Subscriber-Dienste, Sprach-, Daten- und Breitband-Dienste</li> <li>• Network-Mediation: konvergentes Billing und Mediation für traditionelle Dienste (Sprachdienste etc.) sowie IP-basierte Dienste in mobilen und kabelgebundenen Systemen</li> </ul>

<b>Firma</b>	<b>Nazca Software</b>
URL	<a href="http://www.isp-billing.de">http://www.isp-billing.de</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• IT-Branche</li> </ul>
Produkte	<p>OSS für die Bereiche</p> <ul style="list-style-type: none"> <li>• Konvergentes Billing für die Dienstarten Application, Content und Telefonie</li> <li>• Abrechnungssoftware</li> </ul>

<b>Firma</b>	<b>Netcracker</b>
URL	<a href="http://www.netcracker.com">http://www.netcracker.com</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Service-Provider</li> <li>• Öffentliche Verwaltung</li> <li>• Kabelnetzbetreiber</li> </ul>
Produkte	<p>OSS mit Funktionen zum</p> <ul style="list-style-type: none"> <li>• Service- und Ressourcen-Inventory-Management</li> <li>• Service-Provisioning und -Aktivierung</li> <li>• Kundenanalyse</li> <li>• Telekommunikationskosten-Management</li> <li>• Asset-Management</li> </ul>

<b>Firma</b>	<b>Nimbus Systems</b>
URL	<a href="http://www.nimbussys.com">http://www.nimbussys.com</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Finanzdienstleistungen</li> <li>• Versorgungsbetriebe</li> </ul>
Produkte	<p>Produkte, in den Bereichen</p> <ul style="list-style-type: none"> <li>• Billing, Rating und Settlement: konvergentes Rating und Billing für beliebige Daten (Einzelhandel, Sprachdienste etc.)</li> <li>• Settlement: Unterstützung von Revenue-Sharing-Contracts</li> <li>• Mobile-Payment-Lösungen</li> <li>• Business-Intelligence</li> </ul>



<b>Firma</b>	<b>Ntels</b>
URL	<a href="http://www.ntels.com">http://www.ntels.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Fernsehen (Kabelfernsehen, Direct-to-Home-Fernsehen)</li> <li>• Local-Carrier im Telekommunikationssektor</li> <li>• Internet-Provider und VoIP-Provider</li> <li>• Telekommunikation (Sprache, Daten und Video)</li> </ul>
Produkte	<p>Produkte, in den Bereichen</p> <ul style="list-style-type: none"> <li>• Customer-Care und IP-Billing: Billing nach Zeit, Art und Größe des Datenabrufs, nach Price-Policy, Discount etc. basierend auf der Nutzung von Sprach-, Daten- und IP-Diensten. Bereitstellung der Billing-Daten dem Kunden gegenüber. Unterstützung des Settlements durch externe Dienstleister</li> <li>• IP-Billing-Mediation: Lösung zur Extraktion, Konvertierung, Filterung und Klassifikation von Daten, so dass die Informationen wie Art der Daten, Abrufzeit etc. von unterschiedlichen Netzwerk-Devices verfügbar gemacht werden</li> <li>• Service-Provisioning und -Aktivierung</li> <li>• Kundenanalyse</li> <li>• Telekommunikationskosten-Management</li> <li>• Asset-Management</li> </ul>

<b>Firma</b>	<b>OmniOSS</b>
URL	<a href="http://www.omnioss.com">http://www.omnioss.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Internet-Service-Provider (Festnetz, Wireless)</li> </ul>
Produkte	<p>BSS/OSS für IP-basierte Dienste, kabelgebundenes und kabelloses Internet</p> <ul style="list-style-type: none"> <li>• Integrierte Systeme für Customer-Care und Billing mit Funktionen für Billing/CRM, Diensteeerbringung und -aktivierung, Produktkataloge, Beschwerde-Management, Online-Kundenservice, Partner-Management, Kommissionierung, Settlement etc.</li> </ul>

<b>Firma</b>	<b>Oracle Communications und Portal Software</b>
URL	<a href="http://www.oracle.com">http://www.oracle.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Handel</li> <li>• Telekommunikation</li> <li>• Automobil-, Luftfahrt-, Rüstungsindustrie</li> </ul>
Produkte	<p>Software der Firma Oracle „Oracle Advanced Pricing“ umfasst die Funktionen (Stand Mai 2004)</p> <ul style="list-style-type: none"> <li>• Verwaltung von Listenpreisen für Produkte, inkl. Block-Pricing</li> <li>• Verbrauchsabhängiges Pricing</li> <li>• Web-Shops</li> <li>• Discounting, Couponing, Upgrades etc.</li> </ul> <p>Software der früheren Firma Portal Software umfasst die Funktionen:</p> <ul style="list-style-type: none"> <li>• Pricing-Management: Definition unterschiedlicher Produkte und Dienste</li> <li>• Customer-Management: Möglichkeit Postpaid- und Prepaid-Dienste in einem einzigen Kundenkonto zu verwalten Verwaltung von Balances für unterschiedliche Dienste mit getrennt definierbaren Thresholds</li> <li>• Partner-Management: Anbindung neuer Service-Partner inkl. Vereinbarung von Revenue-Sharing-Vereinbarungen</li> <li>• Autorisierung: Unterstützung der Autorisierung für unterschiedliche Diensttypen (Wireless, Sprache, Daten, Content)</li> <li>• Rating, Discounting, Promotions, Billing</li> <li>• Revenue-Assurance</li> <li>• Anbindung an ERP-Systeme</li> </ul>

<b>Firma</b>	<b>Orange Objects</b>
URL	<a href="http://www.orangeobjects.de/">http://www.orangeobjects.de/</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation, insb. Local-Carrier</li> </ul>
Produkte	<p>Die Produkte decken folgende Funktionen ab:</p> <ul style="list-style-type: none"> <li>• Rating von Voice-CDRs ohne Beschränkung der Zahl der Tarifmodelle</li> <li>• Konvergentes Billing (Sprachdienste, IP,...) für Endkunden und Inter-Connection-Partner</li> <li>• Rabattierung und Provisionierung</li> <li>• Vertragsverwaltung und Call-Center-Funktionalität</li> <li>• Datenex-/import (Stamm- und Vertragsdatenübernahme von SAP/R3 etc.)</li> </ul>

<b>Firma</b>	<b>Orga Systems</b>
URL	<a href="http://www.orga-systems.com">http://www.orga-systems.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Internet-Service-Provider</li> </ul>
Produkte	<p>Produkte spezialisiert auf Billing, u.a. in den Bereichen:</p> <ul style="list-style-type: none"> <li>• Konvergentes Prepaid- und Postpaid-Billing, IP/Data-Billing</li> <li>• Mediation</li> <li>• Bonus- und Promotion</li> <li>• Voucher und Recharge</li> <li>• Customer-Self-Care</li> </ul>

<b>Firma</b>	<b>Par3</b>
URL	<a href="http://www.par3.com/">http://www.par3.com/</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Reiseindustrie</li> <li>• Handel</li> <li>• Healthcare</li> </ul>
Produkte	<p>Spezialisiert auf Kommunikationslösungen für die Rechnungsstellung gegenüber Kunden.</p> <ul style="list-style-type: none"> <li>• Aggregation von Daten aus den Produktdatenbanken eines Providers (Billing, CRM, Telefontransaktionssysteme etc.)</li> <li>• Bereitstellung dieser Daten für eine „One-to-one Communication“ von Voice-CDRs ohne Beschränkung der Tarifmodelle</li> <li>• Konvergentes Billing (Sprachdienste, IP,...) für Endkunden und Inter-Connection-Partner</li> <li>• Rabattierung und Provisionierung</li> <li>• Vertragsverwaltung und Call-Center-Funktionalität</li> <li>• Datenex-/import über offene Schnittstellen (Stamm- und Vertragsdatenübernahme von SAP/R3 etc.)</li> </ul>

<b>Firma</b>	<b>Phoneware</b>
URL	<a href="http://www.phoneware.com.au/">http://www.phoneware.com.au/</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Internet-Service-Provider</li> </ul>
Produkte	<p>Business-Management-Software zur Kostenüberwachung und zum Troubleshooting in Sprach- und Datennetzen. Funktionen:</p> <ul style="list-style-type: none"> <li>• Konvergentes Billing (Sprache, Voice Mail, ISP, Carrier, Gerätemiete etc.)</li> <li>• Kostenverrechnung</li> <li>• Netzwerkmanagement-Software für VoIP- und Daten-Netzwerke</li> <li>• Monitoring (nicht zulässige Telefonnummern, übermäßige Gesprächsdauern etc.)</li> </ul>

<b>Firma</b>	<b>Select Software Solutions</b>
URL	<a href="http://www.selectsoftware.com.au/products.htm">http://www.selectsoftware.com.au/products.htm</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> </ul>
Produkte	<p>Funktionen:</p> <ul style="list-style-type: none"> <li>• Customer-Care und Billing-Systeme für alle gängigen australischen Carrier</li> <li>• Billing: Unterstützung für Telefon-Tarife, Entgelte für Leitungsmiete und Discounts</li> <li>• Customer-Care: Request-Tracking-System zur Behandlung von Kundenanfragen</li> <li>• Unterstützung von Telekommunikationsdiensten für Festnetz, Mobilfunk, VoIP, Datenkommunikation (Breitband, DSL), Video-over-IP</li> <li>• Sales und Provisioning</li> </ul>

<b>Firma</b>	<b>Sigma Systems</b>
URL	<a href="http://www.sigma-systems.com">http://www.sigma-systems.com</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> </ul>
Produkte	<p>Billing-System und OSS für Telekommunikationsdienst-Provider.</p> <ul style="list-style-type: none"> <li>• Grundlage ist eine Service-Management-Plattform mit folgenden Funktionen: Order-Management, Service-Provisioning, Dienst-Aktivierung/Autorisierung, Ressourcen-Management, Self-Care und Diagnose für komplexe Dienste in einem Multi-Vendor-Service-Delivery-Netzwerk</li> <li>• Auf der Service-Management Plattform setzen Anwendungen auf, die OSS-Funktionen implementieren, z.B. zur Generierung neuer Dienste, zum Katalogmanagement von Diensten, für den Order-Eingang, zum Management der Netzwerk-Topologie und zur Dienstdiagnose</li> <li>• Auf Basis der Applikationen werden vorgefertigte Pakete für spezifische Domänen (kabelgebundene Dienste, Wireless, DSL etc. angeboten</li> </ul>

<b>Firma</b>	<b>Soprano</b>
URL	<a href="http://www.soprano.com.au">http://www.soprano.com.au</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Mobil-Commerce-Lösungen für <ul style="list-style-type: none"> <li>○ Finanzindustrie</li> <li>○ TV-Industrie</li> <li>○ Government</li> </ul> </li> </ul>
Produkte	<p>Spezialisiert auf die Erstellung von Plattformen für den Bereich M-Commerce</p> <ul style="list-style-type: none"> <li>• Deployment von Mobile-Commerce-Diensten, z.B. Ticketing, Parking, Banking, Wiederaufladen von Prepaid-Karten, Bezahlung von Rechnungen und Zahlungen zwischen Personen (Person-to-Person-Payment)</li> <li>• Der Plattform-Betreiber kann über die Software als Händler agieren und Kunden / Händlern M-Commerce-Dienste anbieten, z. B. Verarbeitung von Autorisierungsanfragen, Ausstellung und Überprüfung von One-Time-Passwörtern, Content-Management, Information-on-Demand, Auslieferung auf Basis von Events oder scheduled, Rating und Settlement</li> </ul>

<b>Firma</b>	<b>Subex Azure</b>
URL	<a href="http://www.subexazure.com">http://www.subexazure.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Nischenanbieter für Software zur Erlösmaximierung im TK-Bereich</li> </ul>
Produkte	<ul style="list-style-type: none"> <li>• Fraud-Management-System: regelbasierte Auslösung von Alarmen und Aufdeckung von Fraud durch Pattern-Matching</li> <li>• Revenue-Assurance-System: Kostenoptimierung, „Assuring earned revenue“ und „Revenue enhancement“, Identifizierung von Revenue-Leakages über die gesamte Wertschöpfungskette Order-to-Cash-Überwachung</li> <li>• Interparty-Management-System für die Bereiche Retail, Interconnect, Großhandel, IP- und Sattelitenlösungen</li> <li>• Interconnect-Billing: Verwaltung von Geschäftsbeziehungen und Abbildung von Interconnection-Abrechnungsmodalitäten</li> </ul>

<b>Firma</b>	<b>Telcordia</b>
URL	<a href="http://www.telcordia.com">http://www.telcordia.com</a>
Anwendungsdomäne / Kunden	<ul style="list-style-type: none"> <li>• Telekommunikation</li> <li>• Öffentliche Verwaltung</li> </ul>
Produkte	<p>OSS Software umfasst u.a.</p> <ul style="list-style-type: none"> <li>• Customer-Care und Billing: Charging, Mediation, Exception-Handling, Performance-Monitoring,</li> <li>• Service-Management: Aktivierung, Bereitstellung, Netzwerkmanagement (under- und over-provisioned areas)</li> <li>• Resource- und Inventory-Management</li> </ul>

<b>Firma</b>	<b>Viziqor</b>
URL	<a href="http://www.viziqor.com">http://www.viziqor.com</a>
Anwendungs- domäne / Kunden	<ul style="list-style-type: none"><li>• Telekommunikation</li></ul>
Produkte	<ul style="list-style-type: none"><li>• Billing- und OSS-System zur Erstellung von Plattformen für den Bereich M-Commerce</li><li>• Konvergentes Billing, inkl. Mediation, Provisioning, Rating und Self-Care für Kunden</li><li>• Netzwerk-Management</li><li>• Inventory-Management</li><li>• Revenue-Assurance</li></ul>

## Literaturverzeichnis

- Abdelnur, A. und Hepper, S. (2003): JSR 168: Java™ Portlet Specification - Version 1.0, Final Release, <http://www.jcp.org/en/jsr/detail?id=168> (06.07.2005).
- Abrazhevich, D. (2001). Electronic Payment - Classification and Characteristics of Electronic Payment Systems, in: Proceedings of Electronic Commerce and Web Technologies: 2<sup>nd</sup> International Conference, EC-Web, LNCS 2115, München, Springer-Verlag, S. 81.
- Adam, N. R. und Yesha, Y. (1996). Electronic Commerce: Current Research Issues and Applications - Workshop, New York, Springer-Verlag.
- Adl. (2004a): SCORM 2004 Overview, [http://www.adlnet.org/screens/shares/dsp\\_displayfile.cfm?fileid=992](http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=992) (17.12.2005).
- Adl. (2004b): SCORM Content Aggregation Model - Version 1.3, [http://www.adlnet.org/screens/shares/dsp\\_displayfile.cfm?fileid=993](http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=993) (17.12.2005).
- Adl. (2004c): SCORM XML Controlling Document - SCORM CAM Version 1.3 Content Packaging Extensions XML XSD Version, [http://www.adlnet.org/screens/shares/dsp\\_displayfile.cfm?fileid=994](http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=994) (17.12.2005).
- Albers, S. (2001). Strategien zum Markterfolg, in: Albers, S., Clement, M., Peters, K. und Skiera, B. (Hrsg.), Marketing mit interaktiven Medien, 3. Auflage, Frankfurt, Verlag Frankfurter Allgemeine.
- Albers, S., Clement, M., Peters, K. und Skiera, B. (2001). Marketing mit interaktiven Medien, 3. Auflage, Frankfurt, Verlag Frankfurter Allgemeine.
- Albers, S. und Herrmann, A. (2002). Handbuch Produktmanagement - Strategieentwicklung - Produktplanung - Organisation - Kontrolle, 2., überarbeitete und erweiterte Aufl. September 2002, Wiesbaden, Gabler-Verlag.
- Alder, G. (2002). Design and Implementation of the JGraph Swing Component.

- Allen, I. E. und Seaman, J. (2003). Sizing the Opportunity - The Quality and Extent of Online Education in the United States, 2002 and 2003, The Sloan Consortium, Needham, MA, USA.
- Aydinli, K. (2002). Mobile Banking, Köln, Bank-Verlag.
- Bacsich, P., Ash, C., Heginbotham, S. und Kandare, P. (2001). The Costs of Networked Learning - Phase Two, Sheffield Hallam University, UK.
- Bakos, Y. und Brynjolfsson, E. (1999): Bundling Information Goods: Pricing, Profits, and Efficiency, in: Management Science, Vol. 45, No. 12, S. 1613-1618.
- Bar, F. (2001). The Construction of Marketplace Architecture, in: Tracking a Transformation: E-Commerce and the Terms of Competition in Industries, Washington DC, Brookings Institution Press, S. 27-49.
- Basoglu, O. (2006). Prototypische Implementierung von dynamischen Preismodellen mittels Prädikate-Transitionen-Netzen, *Diplomarbeit*, Johann Wolfgang Goethe-Universität, Frankfurt, 80 S.
- Bass, J. M., Browne, A. R., Hajji, M. S., Marriot, D. G., Croll, P. R. und Fleming, P. J. (1994): Automating the Development of Distributed Control Software, in: IEEE Concurrency, Vol. 2, No. 4, S. 9-19.
- Baumgarten, B. (1996). Petri-Netze - Grundlagen und Anwendungen, 2. Aufl., Heidelberg et al., Spektrum Akad. Verlag.
- Bechmann, T. und Huhn, G. (1999): Electronic Commerce in der Assekuranz, Tagungsauswertung des Institute for International Research (IIR), in: IVW Management-Information HSG Trendmonitor, Vol. 4, 1. August 1999, S. 12-22.
- Becker, J. und Schütte, R. (2004). Handelsinformationssysteme - Domänenorientierte Einführung in die Wirtschaftsinformatik, 2. Aufl., Redline Wirtschaft Verlag.
- Bellas, F. (2004): Standards for Second-Generation Portals, in: IEEE Internet Computing, Vol. 8, No. 2, S. 54-60.
- Benninger, S. und Grandjot, H.-H. (2001). Supply Chain - Revolution durch E-Commerce, Hamburg, Deutscher Verkehrs-Verlag.
- Bentlage, U. (2002). eLearning - Markets, Business Models, Perspectives, Gütersloh, Bertelsmann Stiftung.



- Bergen, M., Ritson, M., Dutta, S., Levy, D. und Zbaracki, M. (2003): Shattering the Myth of Costless Price Changes, in: European Management Journal, Vol. 21, No. 6, S. 663-669.
- Bernecker, M. (2005). Bildungsmarketing, 2. Aufl., johanna verlag.
- Beutner, M. und Twardy, M. (2003). Lernkonstellationen als didaktische Umsetzungsvariante – Überlegungen zu einem fächerübergreifenden praxisorientierten Konzept unter Berücksichtigung von Tele- und Präsenzlernphasen, in: Dehnbostel, P. (Hrsg.), Perspektiven moderner Berufsbildung: E-Learning - didaktische Innovationen - modellhafte Entwicklungen, Bielefeld, Bertelsmann-Verlag.
- Blake, S. (1998): An Architecture for Differentiated Services, RFC2475 (IETF), <http://www.ietf.org/rfc/rfc2475.txt>.
- Blois, K., Gijbrechts, E. und Campo, K. (2000). Pricing, in: Oxford Textbook of Marketing, Oxford University Press.
- Bodendorf, F. (1999). Wirtschaftsinformatik im Dienstleistungsbereich, Berlin et al., Springer-Verlag.
- Bodendorf, F. und Robra-Biisantz, S. (2003). Elektronische Dienstleistungen in der Finanzwirtschaft, München, Oldenbourg-Verlag.
- Bohley, P. (1989). Statistik: einführendes Lehrbuch für Wirtschafts- und Sozialwissenschaftler, München, Oldenbourg-Verlag.
- Böhm, R., Fuchs, E. und Pacher, G. (1996). System-Entwicklung in der Wirtschaftsinformatik, 4. Aufl., Zürich, vdf Hochschulverlag.
- Borden, N. (1964): The Concept of the Marketing Mix, in: Journal of Advertising Research, Vol. 4, S. 2-7.
- Brandtweiner, R. (2000). Differenzierung und elektronischer Vertrieb digitaler Informationsgüter, 1. Aufl., Düsseldorf, Symposium Publishing.
- Braude, E. J. (2001). Software Engineering: An object-oriented Perspective, New York et al., John Wiley & Sons, Inc.
- Brockhoff, K. (1999). Produktpolitik, Stuttgart, Lucius & Lucius-Verlag.
- Bullinger, H.-J. und Fähnrich, K.-P. (1997). Betriebliche Informationssysteme: Grundlagen und Werkzeuge der methodischen Softwareentwicklung, Berlin et al., Springer-Verlag.
- Bushan, B., Gringel, T., Ryan, C., Leray, E., De Leastar, E. und Cloney, J. (2002). Federated Accounting Management System Architecture for Multimedia

- Service Usage Management, in: Almeroth, K. C. und Hasan, M. (Hrsg.), Proceedings of MMNS 2002; LNCS 2496, Berlin et al., Springer-Verlag, S. 12-24.
- Cesarini, M., Monga, M. und Tedesco, R. (2004). Carrying on the e-learning process with a workflow management engine, in: Proceedings of the 2004 ACM Symposium on Applied computing, Nicosia, Cyprus, ACM Press.
- Chen, P. P. S. (1977). The Entity-Relationship Model: Toward a Unified View of Data, Cambridge, Mass., USA, M.I.T. Center for Information Systems Research.
- Cho, S. M., Hong, H. S. und Cha, S. D. (1996). Safety Analysis Using Coloured Petri Nets, in: Proceedings of the 3<sup>rd</sup> Asia-Pacific Software Engineering Conference (APSEC'96), IEEE Press, S. 176-183.
- Clarke, S. (2004): Standards for Second-Generation Portals, in: IEEE Internet Computing, Vol. March/April 2004, S. 54-60.
- Clement, M., Litfin, T. und Peters, K. (2001). Netzeffekte und Kritische Masse, in: Albers, S., Clement, M., Peters, K. und Skiera, B. (Hrsg.), Marketing mit interaktiven Medien, 3. Auflage, Frankfurt, Verlag Frankfurter Allgemeine, S. 101-115.
- Collier, G. und Robson, R. (2002, 2002): E-Learning Interoperability Standards, SUN Whitepaper, [http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning\\_Interoperability\\_Standards\\_wp.pdf](http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/eLearning_Interoperability_Standards_wp.pdf) (2004-06-10).
- Conrady, R. (2002). eMarketing - Das Internet als Kommunikations- und Distributionskanal, in: Wannenwetsch, H. H. und Nicolai, S. (Hrsg.), E-Supply-Chain-Management. Grundlagen, Strategien, Praxisanwendungen, Wiesbaden, Gabler-Verlag, S. 39-69.
- Crawford, C. M. und Di Benedetto, C. A. (2000). New Product Management, 6<sup>th</sup> ed., McGraw-Hill.
- Cushnie, J. (2000). Evolution of Charging and Billing Models for GSM and Future Mobile Internet Services, in: Crowcroft, J., Roberts, J. und Smirnoy, M. (Hrsg.), Proceedings QofIS 2000, LNCS 1922, Heidelberg et al., Springer-Verlag, S. 312-323.
- Daripa, A. und Kapur, S. (2001): Pricing on the Internet, in: Oxford Review of Economic Policy, Vol. 17, No. 2, S. 202-216.

- Denaro, G. und Pezzè, M. (2003). Petri Nets and Software Engineering, in: Desel, J. und Rozenberg, G. (Hrsg.), Advanced Course on Petri Nets 2003; LNCS 3098, Eichstätt, Springer-Verlag, S. 439-466.
- Desel, J. und Juhás, G. (2001). What is a Petri net? in: Ehrig, H., Juhás, G., Padberg, J. und Rozenberg, G. (Hrsg.), Unifying Petri Nets, LNCS 2128, Berlin, Heidelberg, Springer-Verlag, S. 1-25.
- Desel, J. und Oberweis, A. (1996): Petri-Netze in der Angewandten Informatik: Einführung, Grundlagen und Perspektiven, in: Wirtschaftsinformatik, Vol. 38, No. 4, S. 359-367.
- Desel, J. und Reisig, W. (1998). Place/Transition Petri Nets, in: Reisig, W. und Rozenberg, G. (Hrsg.), Lectures on Petri Nets I: Basic Models - Advances in Petri Nets, LNCS 1471, Berlin et al., Springer-Verlag, S. 122-173.
- Díaz, O. und Paz, I. (2005). Turning Web Applications into Portlets: Raising the Issues, in: Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05), IEEE.
- Diller, H. (2000). Preispolitik, Stuttgart et al., Verlag W. Kohlhammer.
- Din1032-1 (2004). PAS 1032-1 - Aus- und Weiterbildung unter besonderer Berücksichtigung von e-Learning - Teil 1: Referenzmodell für Qualitätsmanagement - Planung, Entwicklung, Durchführung und Evaluation von Bildungsprozessen und Bildungsangeboten, Deutsches Institut für Normung e.V., Berlin.
- Dogrusoz, U., Feng, Q., Madden, B., Doorley, M. und Frick, A. (2002): Graph Visualization Toolkits, in: IEEE Computer Graphics and Applications, No. January/February 2002, S. 30-37.
- Dohmen, D. und Simons, S. (2003). Geschäftsmodelle, in: Dohmen, D. und Michel, L. P. (Hrsg.), Marktpotenziale und Geschäftsmodelle für eLearning-Angebote deutscher Hochschulen, Bielefeld, Bertelsmann-Verlag, S. 145-206.
- Dorloff, F.-D., Leukel, J. und Schmitz, V. (2002): Produktmodelle in elektronischen Katalogen, in: WISU, Vol. 12, No. 02, S. 1557-1591.
- Dti (2005). Study of the e-learning suppliers' "market" in Europe - Final Report, Danish Technological Institute.

- Durmann, D. (2006). Analyse des wirtschaftlichen Potenzials von E-Learning-Angeboten an einer deutschen Hochschule, *Diplomarbeit*, Johann Wolfgang Goethe-Universität, Frankfurt, 88 S.
- Ehrenberg, D. (2001): Internetbasierte Lehrangebote - ein Potenzial für die IT-Aus- und Weiterbildung der Hochschulen, in: HMD - Praxis der Wirtschaftsinformatik, Vol. 218.
- Ernest&Young (2000). A Study to Develop a Costing Methodology for the Australian Higher Education Sector - Final Report, Ernest & Young.
- Euler, D. und Seufert, S. (2005). Fallstudie Universität Stuttgart, e-teaching.org.
- Faßnacht, M. (1996). Preisdifferenzierung bei Dienstleistungen: Implementationsformen und Determinanten, *Dissertation*, Universität Mainz, Wiesbaden, 239 S.
- Faßnacht, M. (2003). Preisdifferenzierung, in: Diller, H. und Hermann, A. (Hrsg.), *Handbuch Preispolitik: Strategien - Planung - Organisation - Umsetzung*, Wiesbaden, Gabler-Verlag, S. 483-502.
- Feess, E. (1994). Mikroökonomie. Eine Einführung in die neoklassische und klassisch-neoricardianische Preis- und Verteilungstheorie, 4. Aufl., Marburg, Metropolis-Verlag.
- Feess, E. (1997). Mikroökonomie - Eine spieltheoretisch- und anwendungsorientierte Einführung, Marburg, Metropolis-Verlag.
- Ferstl, O. K. und Sinz, E. J. (1993). Der Modellierungsansatz des Semantischen Objektmodells (SOM), *Bamberger Beiträge zur Wirtschaftsinformatik*, Nr. 18, Univ., Bamberg.
- Ferstl, O. K. und Sinz, E. J. (1998). *Grundlagen der Wirtschaftsinformatik*, 3. Aufl., München, Wien.
- Foegen, M. (2003): Architektur und Architekturmanagement - Modellierung von Architekturen und Architekturmanagement in der Softwareorganisation, in: HMD - Praxis der Wirtschaftsinformatik, Vol. 232, S. 57-65.
- Foldvary, F. E. (2000). *Das Lexikon der freien Marktwirtschaft*, Düsseldorf, Verlag Wirtschaft und Finanzen.
- Frank, U. und Van Laak, B. L. (2003). Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen, *Arbeitsberichte des Instituts für Wirtschaftsinformatik*, Nr. 34, Universität Koblenz.

- Franz, E., Jerichow, A. und Wicke, G. (1998). A Payment Scheme for Mixes Providing Anonymity, in: Lamersdorf, W. und Merz, M. (Hrsg.), Proceedings of TREC '98, LNCS 1402, Berlin et al., Springer-Verlag, S. 94-108.
- Friedrich, B. C. (2003). Internet-Ökonomie - ökonomische Konsequenzen der Informations- und Kommunikationstechnologien (IuK): Eine industrieökonomische Fallstudie, Dresdner Beiträge zur Volkswirtschaftslehre, Technische Universität Dresden.
- Gansner, E. R. und Reppy, J. H., (Hrsg.). (2004). The Standard ML Basis Library. Cambridge, U.K., Cambridge University Press.
- Geigant, F., Sobotka, D. und Westphal, H. M. (1993). Lexikon der Volkswirtschaft, 4. Aufl., Landsberg/Lech.
- Genrich, H.-J. und Lautenbach, K. (1981): System Modelling with High-Level Petri Nets, in: Theoretical Computer Science, Vol. 13, S. 109-136.
- Gerlach, E. (2003). Cost-Benefit-Analysen für nachhaltigen Erfolg im E-Learning-Bereich -- Eine Checkliste zur Unterstützung von Entscheidungen bei E-Learning-Projekten in Schulen und Hochschulen in Österreich, deutsche medienakademie köln.
- Gerogiannis, V. C., Kameas, A. D. und Pintelas, P. E. (1997): Comparative Study and Categorization of High-Level Petri Nets, in: The Journal of Systems and Software, Vol. 43, S. 133-160.
- Gilmore, S. (1997): Programming in Standard ML '97: A Tutorial Introduction, <http://www.dcs.ed.ac.uk/home/stg> (16.07.2006).
- Girault, C. und Valk, R. (2003). Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications, Berlin et al., Springer-Verlag.
- Goldberg, B. (1996): Functional Programming Languages, in: ACM Computing Surveys, Vol. 28, No. 1, S. 249-251.
- Greiffenberg, S. (2003). Methoden als Theorien der Wirtschaftsinformatik, in: Uhr, W., Esswein, W. und Schoop, E. (Hrsg.), Wirtschaftsinformatik 2003, Band II: Medien Märkte und Mobilität, Dresden, Physica-Verlag, S. 947-967.
- Gruber, T. (1993): A Translational Approach to Portable Ontologies, in: Knowledge Acquisition, Vol. 5, No. 2, S. 199-220.

- Gruene, M., Keferstein, K., Lenz, K., Oberweis, A., Von Mevius, M. und Vossen, G. (2004). Individualization of E-Learning Processes by Workflow-Based Learning-Object Management, in: Abramowicz, W. (Hrsg.), Proceedings of 7th International Conference on Business Information Systems, Bd. 1, Poznan, Polen, Akademia Ekonomiczna w Poznaniu, S. 214-226.
- Gruene, M., Lenz, K. und Oberweis, A. (2005). Pricing of Learning Objects in a Workflow-based E-Learning Scenario, in: Proceedings HICSS 2005, Honolulu, Hawaii, USA.
- Gruene, M. und Oberweis, A. (2005). Model-Based Pricing of E-Learning, in: Karmakar, N. und Isaiás, P. (Hrsg.), Proceedings e-commerce 2005 - IADIS International Conference, Porto, Portugal, IADIS Press, S. 127-135.
- Grüne, M., Keferstein, K., Lenz, K., Oberweis, A., Von Mevius, M. und Vossen, G. (2003). Workflow-gestütztes Lernobjekt-Management, in: Uhr, W., Esswein, W. und Schoop, E. (Hrsg.), Wirtschaftsinformatik 2003, Band I: Medien - Märkte - Mobilität, Dresden, Physica-Verlag, S. 801-818.
- Grüne, M., Nikolopoulos, A. und Holten, R. (2006). Der Einsatz von E-Learning an einer Massenuniversität, in: Proceedings Multikonferenz Wirtschaftsinformatik 2006; Teilkonferenz: E(lectronic)-Learning-Technologiebasiertes Lehren und Lernen, Berlin et al., Springer-Verlag.
- Halmans, G. und Pohl, K. (2002). Modellierung der Variabilität einer Software-Produktfamilie, in: Glinz, M. und Müller-Luschnat, G. (Hrsg.), Proceedings Modellierung 2002 – Modellierung in der Praxis – Modellierung für die Praxis, Lecture Notes in Informatics, Tutzing, Gesellschaft für Informatik, S. 63-74.
- Hanekop, H., Tasch, A. und Wittke, V. (2001): "New Economy" und Dienstleistungsqualität: Verschiebung der Produzenten- und Konsumentenrolle bei digitalen Dienstleistungen, in: SOFI-Mitteilungen, Vol. 29/2001.
- Harper, R. (2005). Programming in Standard ML, Carnegie Mellon University.
- Hartanto, F. und Carle, G. (1999). Policy-based Billing Architecture for Internet Differentiated Services, in: Proceedings of IFIP Fifth International Conference on Broadband Communications (BC'99), Hong Kong.

- Hartung, J. und Elpelt, B. (1999). *Multivariate Statistik - Lehr- und Handbuch der angewandten Statistik*, München, Oldenbourg Wissenschaftsverlag.
- Hasebrook, J. und Otte, M. (2002). *eLearning in the Age of eCommerce - The Third Wave*, Bern, Huber-Verlag.
- Hass, B. H. (2002). *Management neuer Medienunternehmen: Ökonomische Grundlagen und Innovative Geschäftsmodelle, Dissertation*, München, 193 S.
- Hasselbring, W. (2006): *Software-Architektur*, in: *Informatik Spektrum*, Vol. 29, No. 1, S. 48-52.
- Heffner, M. und Mühlfeld, K. (2001). *Application Service Providing - eine Analyse auf Basis des Geschäftstypenansatzes*, Forschungsbericht, Nr. 30, Universität Münster, Münster.
- Herman, I., Melancon, G. und Marshall, M. S. (2000): *Graph Visualization and Navigation in Information Visualization: A Survey*, in: *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, S. 24-43.
- Hermann, E. und Kessler, D. (2005). *XML Signatures in an Enterprise Service Bus Environment*, in: Dittmann, J., Katzenbeisser, S. und Uhl, A. (Hrsg.), *Proceedings Communications and Multimedia Security: 9th IFP TC-6 TC-11 International Conference, CMS 2005, LNCS 3677*, Salzburg, Springer-Verlag, S. 339.
- Hesser, W. (2006): *Verwertung von Inhalten auf der E-Lernplattform*, [http://www.hsu-hh.de/ilias/index\\_AoLkVqIQ5RDnZ4nm.html](http://www.hsu-hh.de/ilias/index_AoLkVqIQ5RDnZ4nm.html) (08.02.2006).
- Hille, S., Jonkers, H., Stap, R. und Bakker, A. (2002). *A Flexible Architecture for Inter-Domain Accounting, Billing and Payment - Deliverable D1.1*, Telematica Instituut, T.-E., Atos Origin, GigaABP, Telematic Instituut, Enschede, Niederlande.
- Hille, S., Jonkers, H., Tokmakoff, A. und Wibbels, M. (2000). *Taxonomy of accounting, billing and payment concepts*, Jonkers, H., GigaABP, Telematica Instituut, Delft et al.
- Holten, R. (1999). *Entwicklung von Führungsinformationssystemen - ein methodenorientierter Ansatz, Dissertation*, Universität Münster, Münster, 306 S.

- Holten, R. (2001). Konstruktion domänenspezifischer Modellierungstechniken für die Modellierung von Fachkonzepten, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 78, Universität Münster.
- Holten, R. (2002). Metainformationssysteme - Backbone der Anwendungssystemkopplung, Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 89, Universität Münster.
- Holten, R. (2003a): Integration von Informationssystemen, in: Wirtschaftsinformatik, Vol. 45, S. 41-52.
- Holten, R. (2003b). Integration von Informationssystemen. Theorie und Anwendung im Supply Chain Management, *Habilitationsschrift*, Westfälische Wilhelms-Universität Münster, 261 S.
- Hoppe, G. und Breitner, M. H. (2003). Business Models for E-Learning, Discussion Paper der wirtschaftswissenschaftlichen Fakultät, Nr. 287, Universität Hannover, Hannover.
- Hudak, P. (1989): Conception, Evolution, and Application of Functional Programming Languages, in: ACM Computing Surveys, Vol. 21, No. 3, S. 359-411.
- Hughes, J. (1989): Why Functional Programming Matters, in: The Computer Journal, Vol. 32, No. 2, S. 98-107.
- Hui, K. L. und Chau, P. Y. K. (2002): Classifying Digital Products, in: Communications of the ACM, Vol. 45, No. 6, S. 73-79.
- Hummert, U. (1989). Algebraische Theorie von High-Level-Netzen, *Dissertation*, TU Berlin, o. S.
- Inoue, K., Nakajima, H. und Yoshikawa, N. (2001). Pricing Strategies in the E-Business Age, NRI Papers, Nr. 23, Nomura Research Institute.
- Iso/Iec19796-1 (2005). ISO/IEC 19796-1 - Information Technology - Learning, Education and Training - Quality Management, Assurance and Metrics - Part 1: General Approach (Standard), PAS 1032-2:2004, ISO, Genf, Schweiz.
- Jensen, K. (1981): Coloured Petri Nets and the Invariant Method, in: Theoretical Computer Science, Vol. 14, S. 317-336.
- Jensen, K. (1986). Coloured Petri Nets, in: Brauer, W., Reisig, W. und Rozenberg, G. (Hrsg.), Advances in Petri Nets, Part I, LNCS 254, Berlin, Springer-Verlag, S. 248-299.



- Jensen, K. (1994). An Introduction to the Theoretical Aspects of Coloured Petri Nets, in: de Bakker, J. W., de Roever, W.-P. und Rozenberg, G. (Hrsg.), A Decade of Concurrency, LNCS 803, Springer-Verlag, S. 230-272.
- Jensen, K. (1996a). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use - Analysis Methods, zweite Auflage, 1996, Berlin et al., Springer-Verlag.
- Jensen, K. (1996b). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use - Basic Concepts, zweite Auflage, 1996, Berlin et al., Springer-Verlag.
- Jensen, K. (1997): A Brief Introduction to Coloured Petri Nets, <http://www.daimi.au.dk/CPnets/intro/BriefIntroduction.pdf> (12.07.2006).
- Jensen, K. (1998). An Introduction to the Practical use of Coloured Petri Nets, in: Reisig, W. und Rozenberg, G. (Hrsg.), Lectures on Petri Nets II: Applications, LNCS 1492, Berlin et al., Springer-Verlag, S. 237-292.
- Jensen, K. und Rozenberg, G. (1991). High-Level Petri Nets - Theory and Applications, Berlin, Springer-Verlag.
- Jügel, M., Kindler, E. und Weber, M. (2000). The Petri Net Markup Language, in: Philippi, S. (Hrsg.), 7. Workshop Algorithmen und Werkzeuge für Petrinetze, AWPN 2000, Universität Koblenz-Landau, Koblenz, S. 47-52.
- Kalakota, R. und Whinston, A. B. (1996). Frontiers of Electronic Commerce, Reading, MA, USA, Addison-Wesley.
- Kamlah, W. und Lorenzen, P. (1996). Logische Propädeutik. Vorschule des vernünftigen Redens, 3. Aufl., Stuttgart, Weimar.
- Karagiannis, D. und Kühn, H. (2002). Metamodeling Platforms, in: Bauknecht, K., Min Tjoa, A. und Quirchmayer, G. (Hrsg.), Proceedings 3<sup>rd</sup> International Conference on EC-Web 2002 - Dexa 2002, LNCS 2455, Aix-en-Provence, Frankreich, Springer-Verlag, S. 182.
- Kaspar, C. und Hagenhoff, S. (2003). Differenzierungsstrategien für digitale Medienprodukte, Schumann, M., Arbeitspapiere der Abteilung Wirtschaftsinformatik II, Nr. 7, Universität Göttingen.
- Katz, R. H. (1990): Toward a Unified Framework for Version Modeling in Engineering Databases, in: ACM Computing Surveys, Vol. 22, No. 4, S. 375-408.

- Keferstein, K., Von Mevius, M. und Oberweis, A. (2005). Management of E-Learning Processes with High-Level Petri Nets, in: Uskov, V. (Hrsg.), Proceedings 8<sup>th</sup> IASTED International Conference on Computers and Advanced Technology in Education (CATE 2005), Oranjestad, Aruba, IASTED.
- Kelly, K. (1998). New Rules for the New Economy - 10 Radical Strategies for a Connected World, New York et al.
- Kemper, P. (2000). Some Notes about Petri Net Interchange Formats, in: Philippi, S. (Hrsg.), 7. Workshop Algorithmen und Werkzeuge für Petrinetze, AWPN 2000, Universität Koblenz-Landau, Koblenz, S. 53-56.
- Kerres, M. (2001). Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung, 2., vollst. überarb. Aufl., München et al., Oldenbourg.
- Kirby, D. und Dally, C. (2000). E-Insurance: Insurance and the Internet Revolution, London, Euromoney Books.
- Knolmayer, G. F. und Montandon, C. (2003). Eignung multimedialer Lernobjekte zur Erreichung der in Blooms Taxonomie unterschiedenen Lernziele, in: Uhr, W., Esswein, W. und Schoop, E. (Hrsg.), Wirtschaftsinformatik 2003, Band I: Medien - Märkte - Mobilität, Dresden, Physica-Verlag, S. 819-838.
- Koch, P. (1998). Versicherungswirtschaft - Ein einführender Überblick, 5., Neubearb. Aufl., Karlsruhe, VVW-Verlag.
- Kom (2002). Vorschlag für einen Beschluss des Europäischen Parlaments und des Rates über ein Mehrjahreprogramm (2004-2006) für die wirksame Integration von Informations- und Kommunikationstechnologien (IKT) in die Systeme der allgemeinen und beruflichen Bildung in Europa (Programm "eLearning"), Nr. 751.
- Korschenewski, E. (2005). Formale, modellbasierte Beschreibung des Pricings digitaler Produkte, *Diplomarbeit*, Universität Karlsruhe (TH), 78 S.
- Kotler, P., Armstrong, G. und Saunders, J. (1999). Grundlagen des Marketing, 2., überarbeitete Auflage, Prentice Hall.
- Kowalk, W. (1996). System, Modell, Programm: vom GOTO zur objektorientierten Programmierung, Heidelberg et al., Spektrum Akademischer Verlag.

- Kristensen, L. M., Jörgensen, J. B. und Jensen, K. (2003). Application of Coloured Petri Nets in System Development, in: Desel, J., Reisig, W. und Rozenberg, G. (Hrsg.), Proceedings ACPN 2003, LNCS 3098, S. 626-685.
- Külzer, W., Krause, T. und Buller, D. (2004). Innovative Preis- und Verrechnungsmodelle für IT-Leistungen, in: Zarnekow, R., Brenner, W. und Grohmann, H. H. (Hrsg.), Informationsmanagement - Konzepte und Strategien für die Praxis, 1. Auflage, Heidelberg, dpunkt, S. 169-176.
- Kumar, A. (2005). *Distributed System Development Using Web Service and Enterprise Java Beans*, in: Proceedings IEEE International Conference on Web Services (ICWS'05), IEEE Press, S. xxiii.
- Kumar, V. und Reinartz, W. J. (2006). Customer Relationship Management: a Databased Approach, Hoboken, Wiley.
- Lampe, F. (1999). Marketing und E-Commerce, 1. Auflage, Wiesbaden.
- Langhammer, R. (2004). Prozessmodellierung für Market Data Administration im Private Banking, *Diplomarbeit*, Johann Wolfgang Goethe-Universität, Frankfurt, 86 S.
- Larsson, M. und Svensson, B.-M. (2004). Classifying Digital Products in an Online Newspaper Context, in: Proceedings IRIS 27, Falkenberg, Sweden.
- Lee, M. H. (1999): On Models, Modelling and the Distinctive Nature of Model-based Reasoning, in: AI Communications, Vol. 12, S. 127-137.
- Lehmann, F. R. (1998): Aktuelles Schlagwort - Normsprache, in: Informatik Spektrum, Vol. 21, S. 366-367.
- Lehner, F. (2003). Mobile und drahtlose Informationssysteme: Technologien, Anwendungen, Märkte, Berlin et al., Springer-Verlag.
- Leithner, B. und Back, A. (2004). Beiträge der Balanced Scorecard für ein nachhaltiges E-Learning im Unternehmen, Arbeitsberichte des Learning Center der Universität St. Gallen, Nr. 4/2004, St. Gallen.
- Lin, J., Ho, C., Sadiq, W. und Orłowska, M. E. (2001). *On Workflow Enabled e-Learning Services*, in: Proceedings IEEE International Conference on Advanced Learning Technologies (ICALT'01), Madison, WI, USA, IEEE Press, S. 349-352.
- Liu, X., El Saddik, A. und Georganas, N. (2003). An Implementable Architecture of E-Learnings System, in: Proceedings of the Canadian Conference on

- Electrical and Computer Engineering (CCECE 2003), Montréal, Quebec, Canada.
- Loebbecke, C. (1999). Electronic Trading in On-line Delivered Content, in: Proceedings of the 32nd Hawaii International Conference on System Sciences, IEEE Press.
- Lorenzen, P. (1973). Semantisch normierte Orthosprachen, in: Kambartel, F. und Mittelstraß, J. (Hrsg.), Zum Normativen Fundament der Wissenschaft, Frankfurt, S. 231-349.
- Loy, M. (2002). Java Swing: [covers Java 2 SDK 1.4], 2<sup>nd</sup> ed., Beijing et al., O'Reilly.
- Ludewig, J. (2002). Modelle im Software Engineering - eine Einführung und Kritik. Proceedings Modellierung 2002. Tutzing, Gesellschaft für Informatik, S. 7-22.
- Ludwig, H., Keller, A., Dan, A., King, R. und Franck, R. (2003): A Service Level Agreement Language for Dynamic Electronic Services, in: Electronic Commerce Research, Vol. 3, S. 43-59.
- Lutz, M. und Pelka, B. (2003). Marktumfeld und Marktentwicklung, in: Dohmen, D. und Michel, L. P. (Hrsg.), Marktpotenziale und Geschäftsmodelle für eLearning-Angebote deutscher Hochschulen, Bielefeld, Bertelsmann, S. 145-206.
- Mack, T. (2002). Schadensversicherungsmathematik, 2. Auflage, Karlsruhe, Verlag Versicherungswirtschaft.
- Mcknight, L. W. und Bailey, J. P. (1997): Internet Economics - When Constituencies Collide in Cyberspace, in: IEEE Internet Computing, Vol. Nov./Dez. 1997, S. 31-37.
- Mclaughlin, B. (2002). Java & XML, 2., deutsche Ausgabe, Beijing u.a., O'Reilly.
- Mending, J., Neumann, G., Pinterits, A. und Bernd, S. (2005). Revenue Models for E-Learning at Universities, in: Proceedings 7. Internationale Tagung Wirtschaftsinformatik 2005, Bamberg.
- Meyer, U. (1999): Kfz-Haftpflichtversicherung in Europa: Vergleichende Untersuchung der ökonomisch-statistischen Situation, <http://web.uni-bamberg.de/sowi/economics/meyer/forschung/kfz/studie-deutsch.pdf> (21.06.2006).

- Mili, H., Mili, A., Yacoub, S. und Addy, E. (2002). Reuse-based Software Engineering: Techniques, Organization, and Measurement, John Wiley & Sons, Inc.
- Mills, H. D. (1980): The Management of Software Engineering - Part I: Principles of Software Engineering, in: IBM Systems Journal, Vol. 19, No. 40.
- Mills, H. D. (1988): Stepwise Refinement and Verification in Box-Structured Systems, in: IEEE Computer, Vol. June 1988, S. 23-36.
- Milner, R. (1984). A proposal for Standard ML. Proceedings of the 1984 ACM Symposium on LISP and functional programming. Austin, Texas, United States, ACM Press, S. 184-197.
- Milner, R., Tofte, M., Harper, R. und Macqueen, D. (1997). The definition of Standard ML (revised), Cambridge, Mass., MIT Press.
- Min, S.-Y., Han, I.-J., Park, W.-J. und Bae, D.-H. (1997). An Approach to Software Process Management Based on Formal Process Modeling and Analysis, in: Proceedings 4th Asia-Pacific Software Engineering and International Computer Science Conference (APSEC '97 / ICSC '97), IEEE Press, S. 292-301.
- Monroe, K. B. (2003). Pricing - Making Profitable Decisions, 3<sup>rd</sup> ed., Boston et al., McGraw-Hill Irwin.
- Moog, H. (1998). Rechnerische Fundierung der Preisfindung: eine Konzeption für vorhandene Gemeinschaftspositionen, *Dissertation*, Universität Trier, Gabler Edition Wirtschaft, 301 S.
- Mortensen, K. H. (1997). Coloured Petri Nets - a Pragmatic Formal Method for Designing and Analysing Distributed Systems, Technical Report No. DAIMI PB-522, CPN Group, University of Aarhus.
- Murata, T. (1989). Petri Nets: Properties, Analysis, and Applications, in: Proceedings of the IEEE, Bd. 77, S. 541-580.
- Nagle, T. T. und Holden, R. K. (1995). The Strategy and Tactics of Pricing: A Guide to Profitable Decision Making, Englewood Cliffs, NJ, Prentice-Hall.
- Nist (1993). Integration Definition for Function Modeling (IDEF0), Department of Commerce, Standards and Technology, Computer Systems Laboratory, <http://www.itl.nist.gov/fipspubs/idef02.doc>.

- Nord, R. L. und Tomayko, J. E. (2006): Software Architecture-Centric Methods and Agile Development, in: IEEE Software, Vol. March/April 2006, S. 47-53.
- Oasis. (2003): Web Services for Remote Portlets Specification, OASIS Standard, <http://www.oasis-open.org/committees/wsrp> (05.10.2005).
- Obbink, H. und Pohl, K. (2005). *Software Product Lines*, in: Proceedings 9<sup>th</sup> International Conference SPLC 2005, LNCS 3714, Rennes, Frankreich, Springer-Verlag.
- Oberweis, A., Pankratius, V. und Stucky, W. (2005). Product Lines in E-Learning, Forschungsberichte des Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Nr. 501, Universität Karlsruhe (TH).
- Oberweis, A. und Sander, P. (1996): Information System Behavior Specification by High-Level Petri Nets, in: ACM Transactions on Information Systems, Vol. 14, No. 4, S. 380-420.
- Olbrich, R. und Grünblatt, M. (2001). Definition Electronic Commerce, in: Bruhn, M. und Homburg, C. (Hrsg.), *Gabler Marketing Lexikon*, 1. Auflage, Wiesbaden, Gabler-Verlag, S. 179.
- Olderog, T. und Skiera, B. (2000): The Benefits of Bundling Strategies, in: *Schmalenbach Business Review*, Vol. 1, S. 137-160.
- Ortner, E. (2005). *Sprachbasierte Informatik - Wie man mit Wörtern die Cyber-Welt bewegt*, Leipzig, Eagle-Verlag.
- Peterson, R. A., Balasubramanian, S. und Bronnenberg, B. J. (1997): Exploring the Implications of the Internet for Consumer Marketing, in: *Journal of the Academy of Marketing Science*, Vol. 25, No. 4, S. 329-346.
- Petri, C. A. (1962). *Kommunikation mit Automaten*, *Dissertation*, Technische Fakultät für Mathematik und Physik der TH Darmstadt, 128 S.
- Picot, A. und Heger, D. K. (2003a). Braucht das Internet eine neue Wettbewerbspolitik? in: Oberender, P. (Hrsg.), *Wettbewerb in der Internetökonomie*, Berlin.
- Picot, A. und Heger, D. K. (2003b). Braucht das Internet eine neue Wettbewerbspolitik? in: Oberender, P. (Hrsg.), *Wettbewerb in der Internetökonomie*, *Schriften des Vereins für Socialpolitik*, N.F. Bd. 292, Berlin, Duncker und Humblot, S. 9-38.

- Pigou, A. C. (1932). *The Economics of Welfare*, 4<sup>th</sup> ed., London, MacMillan and Co.
- Pohl, A. und Kluge, B. (2000): Pricing im Internet: Gewinnoptimale Preisgestaltung ist kein Zufall, <http://www.eai-competence-center.de/marketing.nsf/C2BE4AEF80288853C1256A1F00430266/%24File/pricing%2520im%2520internet.pdf> (05.10.2005).
- Premium (2003). Vorhabensbeschreibung - Erlöstypen und Preismodelle im Internet, Johann Wolfgang Goethe-Universität, Frankfurt.
- Ramboll (2004). Studies in the Context of the E-Learning Initiative: Virtual Models of European Universities (Lot1) - Final Report to the EU Commission, PLS Ramboll Management, <http://www.elearningeuropa.info>.
- Ratzer, A. V., Wells, L., Lassen, H. M., Laursen, M., Qvortrup, J. F., Stissing, M. S., Westergaard, M., Christensen, S. und Jensen, K. (2003). CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets, in: LNCS 2679, Springer-Verlag, S. 450-462.
- Redeker, G. H. J. (2003). An Educational Taxonomy for Learning Objects, in: Proceedings 3<sup>rd</sup> IEEE International Conference on Advanced Learning Technologies (ICALT'03), IEEE Press, S. 250-251.
- Reichl, P., Hausheer, D. und Stiller, B. (2003): The Cumulus Pricing Model as an adaptive Framework for feasible, efficient, and User-friendly Tariffing of Internet Services, in: *Computer Networks*, Vol. 43, No. 1, S. 3-24.
- Reil, D. und Appelrath, H.-J. (2004). Kostenpflichtiger Content in Lernportalen, in: Engels, G. und Seehusen, S. (Hrsg.), *Proceedings DeLFI 2004*, LNI 52, Paderborn, GI, S. 91-102.
- Reisig, W. (1985). *Systementwurf mit Netzen*, Berlin et al., Springer-Verlag.
- Reisig, W. (1986a). Petri Nets in Software Engineering, in: Brauer, W., Reisig, W. und Rozenberg, G. (Hrsg.), *Petri Nets: Applications and Relationships to Other Models of Concurrency*, LNCS 255, Berlin et al., Springer-Verlag, S. 63-96.
- Reisig, W. (1986b). Place/Transition Systems, in: Brauer, W., Reisig, W. und Rozenberg, G. (Hrsg.), *Petri Nets: Central Models and Their Properties - Advances in Petri Nets 1986, Part I*, Proceedings of an Advances Course, LNCS 254, Bad Honnef, S. 117-142.

- Reisig, W. und Rozenberg, G. (1998). Informal Introduction to Petri Nets, in: Reisig, W. und Rozenberg, G. (Hrsg.), Lectures on Petri Nets I: Basic Models, LNCS 1491, Berlin et al., Springer-Verlag, S. 1-11.
- Reklaitis, V., Baniulis, K. und Okamoto, T. (2003). Shaping e-Learning applications for a service oriented Grid, in: Proceedings 2<sup>nd</sup> International LeGE-WG Workshop on e-Learning and Grid Technologies, Paris, British Computer Society.
- Reussner, R. und Hasselbring, W. (2006). Handbuch der Software-Architektur, Heidelberg, dpunkt.
- Rozenberg, G. und Engelfriet, J. (1998). Elementary Net Systems, in: Reisig, W. und Rozenberg, G. (Hrsg.), Lectures on Petri Nets: Advances in Petri nets; 1. Basic Models, LNCS 1491, Berlin et al., Springer-Verlag, S. 12-121.
- Rudolf, R. (2003). GPRS Basics: die Grundkonzepte des General Packet Radio Service, Wilburgstetten, Schlembach-Fachverl.
- Sandhu, R. S., Coyne, E. J., Feinstein, H. L. und Youman, C. E. (1996): Role-Based Access Control Models, in: IEEE Computer, Vol. 29, No. 2, S. 38-47.
- Sauter, A. M., Bender, H. und König, C. (2002). Blended Learning - Effiziente Integration von E-Learning und Präsenztraining, Neuwied, Kriftel (Taunus), Luchterhand-Verlag.
- Scheer, A.-W. (1997). Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 7., erw. Aufl., Berlin et al., Springer-Verlag.
- Scheffer, U. und Hesse, F. W. (2002). eLearning - Applying the Revolution of Learning Profitability, Stuttgart, Klett-Cotta-Verlag.
- Schewe, C. D. und Smith, R. M. (1983). Marketing: Concepts and Applications, 2<sup>nd</sup> ed., New York et al., McGraw-Hill.
- Schierenbeck, H. (1990). Bank- und Versicherungslexikon, München, Wien, Oldenbourg.
- Schmees, M. (2004). Integrating eCommerce into eLearning, in: Janssen, M., Sol, H. G. und Wagenaar, R. W. (Hrsg.), Proceedings 6th International Conference on Electronic Commerce - ICEC'04, S. 177-186.
- Schmid, L. M. (1965): Grundlagen und Formen der Preisdifferenzierung im Lichte der Marktformlehre und der Verhaltenstheorie, in: Volkswirtschaftliche Schriften, Heft 91.



- Schmitz, V., Leukel, J. und Dorloff, F.-D. (2002). Modeling and Exchange of Product Classification Systems using XML, in: Proceedings 4<sup>th</sup> IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2002), S. 242-245.
- Schütte, R. (1998). Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle, *Dissertation*, Universität Münster, 402 S.
- Seufert, S. (2001). E-Learning Business Models, Strategies, Success Factors and Best Practice Examples, in: De Fillippi, R. W. und Wankel, C. (Hrsg.), *Rethinking Management Education for the 21<sup>st</sup> Century*, Greenwich, Information Age Pub.
- Shapiro, C. und Varian, H. R. (1999). *Information Rules: A Strategic Guide to the Network Economy*, Boston, MA, USA, Harvard Business School Press.
- Simon, H., Kucher und Partners. (2003): Pricing: It's the Process - The Call for a Profit Renaissance, <http://www.simon-kucher.com/> (07.10.2003).
- Skiera, B. und Lambrecht, A. (2000). Erlösmodelle im Internet, in: Albers, S. und Herrmann, A. (Hrsg.), *Handbuch Produktmanagement - Strategieentwicklung - Produktplanung - Organisation - Kontrolle*, Wiesbaden, Gabler-Verlag, S. 813-831.
- Skiera, B. und Spann, M. (2002). Preisdifferenzierung im Internet, in: Schögel, M., Tomczak, T. und Belz, C. (Hrsg.), *Ro@map to E-Business - Wie Unternehmen das Internet erfolgreich nutzen*, St. Gallen.
- Sneed, H. M. (2006). Integrating legacy Software into a Service oriented Architecture, in: Proceedings Conference on Software Maintenance and Reengineering (CSMR'06), S. 3-14.
- Sommerville, I. (2001). *Software Engineering*, 6<sup>th</sup> ed., Edinburgh, U.K., Addison Wesley.
- Spann, M., Skiera, B. und Walz, U. (2005): Erlösquellen und Preismodelle für den Business-to-Consumer-Bereich im Internet, in: *Wirtschaftsinformatik*, Vol. 47, No. 4, S. 285-293.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*, Wien, New York, Springer-Verlag.
- Stegmann, R., Koch, M., Lacher, M., Leckner, T. und Renneberg, V. (2003). *Generating Personalized Recommendation in a Model-Based Product*

- Configurator System, in: Proceedings International Joint Conference on Artificial Intelligence (IJCAI) - Workshop on Configuration, Acapulco, Mexiko.
- Stelzer, D. (2000): Digitale Güter und ihre Bedeutung in der Internet-Ökonomie, in: WISU - Das Wirtschaftsstudium, Vol. 2000, No. 6, S. 835-842.
- Stickel, E., Groffmann, H.-D. und Rau, K.-H. (1997). Gabler-Wirtschaftsinformatik-Lexikon, ungek. Wiedergabe der Orig.-Ausg., Taschenbuchkass. mit 2 Bdn., Wiesbaden, Gabler-Verlag.
- Strahring, S. (1996). Metamodellierung als Instrument des Methodenvergleichs: eine Evaluierung am Beispiel objektorientierter Analysemethoden, *Dissertation*, Shaker-Verlag, Aachen, 400 S.
- Tellis, G. J. (1986): Beyond the Many Faces of Price: An Integration of Pricing Strategies, in: Journal of Marketing, Vol. 50, No. 4 (Oct.), S. 146-160.
- Thimm, C., (Hrsg.). (2005). Netz-Bildung: lehren und lernen mit neuen Medien in Wissenschaft und Wirtschaft. Bonner Beiträge zur Medienwissenschaft. Frankfurt et al., Lang-Verlag.
- Timmers, P. (1998): Business Models for Electronic Markets, in: Electronic Markets, Vol. 8, No. 2, S. 3-8.
- Tofte, M. (1993): Tips for Computer Scientists on Standard ML, <http://www.lfcs.inf.ed.ac.uk/software/ML/docs/tips.ps.gz> (15.07.2006).
- Turowski, K. und Pousttchi, K. (2004). Mobile Commerce - Grundlagen und Techniken, Berlin.
- Varian, H. R. (1998). Markets for Information Goods, in: Proceedings Monetary Policy in a World of Knowledge-Based Growth, Quality Change, and Uncertain Measurement (2000).
- Vetter, M. (1998). Aufbau betrieblicher Informationssysteme mittels pseudoobjektorientierter, konzeptioneller Datenmodellierung, 8., durchges. Aufl., Stuttgart, B.G. Teubner.
- Von Boettcher, J. (2004): Online Course Development: What Does It Cost? Campus Technology Magazine, <http://www.campus-technology.com/article.asp?id=9676>.
- Vossen, G. und Jaeschke, P. (2003). Learning Objects as a Uniform Foundation for E-Learning Platforms, in: Proceedings 7<sup>th</sup> IEEE International

- Conference on Database Engineering and Applications, Hong Kong, China, IEEE Press, S. 278-287.
- Vossen, G. und Westerkamp, P. (2003). E-Learning as a Web Service, in: Proceedings 7<sup>th</sup> IEEE International Conference on Database Engineering and Applications, Hong Kong, China, IEEE Computer Society Press, S. 242-249.
- Vossen, G. und Westerkamp, P. (2005). Mediators for Integrating Content into Service-Based e-Learning Environments, in: Proceedings 5<sup>th</sup> IEEE International Conference on Advanced Learning Technologies (ICALT'05), IEEE Press, S. 580-584.
- Vugdalic, D. (2004). Modellierung von Tarifen für digitale Produkte am Beispiel von Mobilfunkverträgen, *Diplomarbeit*, Johann Wolfgang Goethe-Universität, Frankfurt, 71 S.
- Wache, M. (2003): E-Learning im Bereich Hochschule, <http://www.bpb.de> (06.06.2006).
- Weber, M. und Kindler, E. (2003). The Petri Net Markup Language, in: Ehrig, H., Reisig, W., Rozenberg, G. und Weber, H. (Hrsg.), Petri Net Technology for Communication-Based Systems: Advances in Petri Nets, LNCS 2472, Berlin et al., Springer, S. 124-144.
- Wedekind, H. (1981). Datenbanksysteme I: Eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung, 2. Aufl., Mannheim, BI-Wissenschafts-Verlag.
- Wedekind, H., Görz, G., Kötter, R. und Inhetveen, R. (1998): Modellierung, Simulation, Visualisierung: Zu aktuellen Aufgaben der Informatik, in: Informatik-Spektrum., Vol. 21, No. 5, S. 265-272.
- Weinhardt, C., Holtemann, C. und Neumann, D. (2003): Market-Engineering, in: Wirtschaftsinformatik, Vol. 45, No. 6, S. 635-640.
- Whinston, A. B., Stahl, D. O. und Choi, S.-Y. (1997). The Economics of Electronic Commerce, Indianapolis, IN, USA, MacMillan.
- Wiley, D. (2000). The Instructional Use of Learning Objects, Logan, UA, USA.
- Wilms, F. (2004). Computergestützte Modelle im Lernprozess, in: Lehner, M. und Döring, K. W. (Hrsg.), E-Learning und Didaktik: Perspektiven für die betriebliche Bildung, Düsseldorf, Symposion Verlag.
- Wirtz, B. W. (2000). Electronic Business, Wiesbaden, Gabler-Verlag.

- Zelewski, S. (1996): Eignung von Petrinetzen für die Modellierung komplexer Realsysteme - Beurteilungskriterien, in: Wirtschaftsinformatik, Vol. 1996, No. 38, S. 369-381.
- Zerdick, A., Picot, A., Schrape, K., Artopé, A., Goldhammer, K., Lange, U. T., Vierkant, E., López-Escobar, E. und Silverstone, R. (2000). E-Conomics - Strategies for the Digital Marketplace, Berlin et al., Springer-Verlag.
- Zuser, W., Grechenig, T. und Köhle, M. (2004). Software Engineering mit UML und dem Unified Process, 2., überarb. Aufl., München et al., Pearson.

Ein wesentlicher Beitrag zur Sicherung der Nachhaltigkeit von E-Learning-Angeboten liegt in der Fähigkeit eines E-Learning-Providers, Erlöse zu erzielen und diese durch geeignete Preisdifferenzierungsstrategien zu „maximieren“.

Preisdifferenzierungsstrategien werden in Preismodellen kodifiziert und in Abrechnungssystemen umgesetzt. Marktgängige Abrechnungssysteme für digitale Produkte sind für die Abrechnung digitaler Produkte im E-Learning ungeeignet. Gründe hierfür sind die Abhängigkeit von Hardware und Software, die mangelnde Anwenderunterstützung beim Tarifentwurf und die daraus entstehende, unzureichende Unterstützung bei der Einführung von Preismodellen für eine Vielzahl von Produktvarianten.

Die Arbeit adressiert die genannten Probleme und schlägt als Lösungsansatz die Entwicklung von Abrechnungssystemen vor, die durch den flexiblen Austausch von Modellen konfiguriert werden. In einem ersten Schritt werden dazu eine konzeptuelle Beschreibungssprache für Preismodelle (PriMoL) formal definiert und Begriffe im Pricing aus Sicht der Wirtschaftsinformatik normiert. PriMoL und ein Prototyp unterstützen den grafischen Entwurf von Preismodellen durch alle am Pricing beteiligten Personengruppen. In einem zweiten Schritt wird die Einsatzfähigkeit von PriMoL anhand eines entwickelten E-Learning-Systems und prototypischen Abrechnungssystems aufgezeigt. Neben dem E-Learning wird für Produkte auch das Pricing im Versicherungsbereich und im Mobilfunk skizziert.

édition scientifique  
**VVB LAUFERSWEILER VERLAG**

VVB LAUFERSWEILER VERLAG  
STAUFENBERGRING 15  
D - 3 5 3 9 6 G I E S S E N

Tel: 0641-5599888 Fax: -5599890  
redaktion@doktorverlag.de  
www.doktorverlag.de

ISBN 3-8359-5185-8



9 78 3 835 951853