

---

Entwurf von Geschäftsprozessen  
mit  
Petri-Netzen

---

Zur Erlangung des akademischen Grades eines  
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

an der Fakultät für Wirtschaftswissenschaften  
der Universität Fridericiana zu Karlsruhe

eingereichte

DISSERTATION

von

Dipl.-Wi.-Ing. Thomas Erwin

REFERENT:

Prof. Dr. Jörg Desel

KORREFERENT:

Prof. Dr. Andreas Geyer-Schulz

TAG DER MÜNDLICHEN PRÜFUNG:

22. Februar 2002

Karlsruhe, 2002



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Geschäftsprozeßmodelle als Schnittstelle beim Entwurf von Informationssystemen . . . . .	3
1.2	Aspekte der Qualität von Geschäftsprozeßmodellen . . . . .	6
1.3	Entwurf von nicht vollständig strukturierten Geschäftsprozessen . . . . .	7
1.4	Aufbau und einzelne Beiträge der Arbeit . . . . .	9
<b>2</b>	<b>Entwurf von Geschäftsprozessen: Grundlagen, Qualität und Ansätze</b>	<b>13</b>
2.1	Grundlagen: Begriffe und Typen . . . . .	13
2.1.1	Begriffe . . . . .	14
2.1.2	Typen . . . . .	15
2.2	Qualität von Geschäftsprozeß-Entwürfen . . . . .	18
2.2.1	Qualität (I): Korrektheit und Validierung . . . . .	20
2.2.2	Qualität (II): Konsistenz und Verifikation . . . . .	21
2.2.3	Qualität (III): Effizienz und Leistungsbewertung . . . . .	23
2.2.4	Anforderungen an Formalismen und Methoden für den Entwurf . . . . .	32
2.3	Ansätze für den Entwurf von Geschäftsprozessen . . . . .	35
2.3.1	Allgemeinere Arbeiten . . . . .	35
2.3.2	Modellbasierte Ansätze . . . . .	37
2.3.3	Petrinetzbasierte Ansätze . . . . .	43
2.3.4	Fazit . . . . .	47
<b>3</b>	<b>Der <i>VIPbusiness</i>-Ansatz</b>	<b>49</b>
3.1	Ein 3-stufiger Ansatz . . . . .	49
3.1.1	Schritt 1: Modellierung . . . . .	50
3.1.2	Schritt 2: Simulation . . . . .	53
3.1.3	Schritt 3: Analyse und Optimierung . . . . .	55
3.1.4	Arbeiten mit <i>VIPbusiness</i> . . . . .	56
3.2	Werkzeugunterstützung des <i>VIPbusiness</i> -Ansatzes . . . . .	61
3.2.1	Ein Anforderungsprofil für ein Werkzeug für den Geschäftsprozeß-Entwurf . . . . .	61
3.2.2	Untersuchung kommerzieller Werkzeuge . . . . .	65
3.2.3	Profile der Werkzeuge . . . . .	75
3.2.4	Profil des <i>VIPtools</i> . . . . .	82

<b>4</b>	<b>Modellierung: Grundlagen</b>	<b>85</b>
4.1	Einfache Petrinetze . . . . .	86
4.2	Höhere Petrinetze . . . . .	91
4.3	<i>VIPmodel</i> -Netze . . . . .	95
<b>5</b>	<b>Modellierung mit <i>VIPbusiness</i>-Netzen</b>	<b>99</b>
5.1	Aktivitäts-Transitionen . . . . .	101
5.1.1	Vergrößerungen . . . . .	101
5.1.2	Aktivitäts-Transitionen als Vergrößerungen . . . . .	103
5.2	Routing . . . . .	106
5.2.1	Sequenz . . . . .	108
5.2.2	AND-Konstrukte . . . . .	109
5.2.3	OR-Konstrukte . . . . .	110
5.2.4	Zyklen . . . . .	116
5.2.5	Zusammengesetzte Routing-Konstrukte . . . . .	117
5.2.6	Zusammenfassung . . . . .	119
5.3	Nutzung von Ressourcen . . . . .	119
5.3.1	Modellierung als Erweiterung der Kantenbeschriftung . . . . .	122
5.3.2	Modellierung als interne Stelle . . . . .	123
5.4	Zeit . . . . .	124
5.4.1	Grundsätzliche Modellierungsmöglichkeiten . . . . .	126
5.4.2	Bekannte Ansätze . . . . .	128
5.4.3	Zeitmodellierung für Geschäftsprozesse . . . . .	130
5.5	Kosten . . . . .	133
5.6	Nicht-strukturierte Teile und hybride Spezifikationen . . . . .	136
5.6.1	Startzustand und Zielzustand . . . . .	138
5.6.2	Fragmente . . . . .	140
5.6.3	Business Rules . . . . .	143
5.6.4	Hybride Spezifikationen . . . . .	148
5.7	<i>VIPbusiness</i> -Netze . . . . .	149
<b>6</b>	<b>Simulation</b>	<b>153</b>
6.1	Beschreibung des Verhaltens von Petrinetzen . . . . .	154
6.1.1	Schaltfolgen und sequentielle Semantik . . . . .	154
6.1.2	Halbgeordnete Abläufe und kausale Semantik . . . . .	157
6.2	Das klassische <i>VIP</i> -Simulationskonzept . . . . .	163
6.3	<i>VIPbusiness</i> -Erweiterungen . . . . .	164
6.3.1	Abläufe und Aktivitäts-Transitionen . . . . .	165
6.3.2	Routing-Konstrukte in Abläufen . . . . .	166
6.3.3	Nutzung von Ressourcen in Abläufen . . . . .	167
6.3.4	Abbildung von Zeitgrößen auf Abläufe . . . . .	168
6.3.5	Abbildung von Kostengrößen auf Abläufe . . . . .	170
6.3.6	Abläufe von nicht-strukturierten Segmenten . . . . .	171
6.3.7	Abläufe von hybriden Spezifikationen . . . . .	178

<b>7</b>	<b>Analyse und Optimierung</b>	<b>183</b>
7.1	Methoden zur qualitativen Analyse . . . . .	184
7.1.1	Verifikation mit Hilfe „klassischer“ Eigenschaften von Petrinetzen . . .	184
7.1.2	Restriktive Korrektheit: Überprüfung von Business Rules . . . . .	186
7.2	Methoden zur Zeitanalyse . . . . .	191
7.2.1	Ermittlung der Aktivitätszeit . . . . .	191
7.2.2	Ermittlung der echten Wartezeit . . . . .	195
7.2.3	Ermittlung der Durchlaufzeit . . . . .	201
7.2.4	Ressourcenbezogene Zeitgrößen . . . . .	201
7.2.5	Abgeleitete Zeitgrößen . . . . .	206
7.3	Methoden zur Kostenanalyse . . . . .	209
7.3.1	Ermittlung der Aktivitätskosten . . . . .	210
7.3.2	Ermittlung der echten Wartekosten . . . . .	214
7.3.3	Ressourcenbezogene Kostengrößen . . . . .	220
7.3.4	Ermittlung der Gesamtkosten . . . . .	224
7.3.5	Abgeleitete Kostengrößen . . . . .	225
7.4	Weitere Methoden zur Ressourcenanalyse . . . . .	229
7.4.1	Linearisierungen durch Ressourcen . . . . .	229
7.4.2	Konfiguration von Kontrollparametern . . . . .	237
7.5	Konfigurationen und Szenarien . . . . .	249
7.5.1	Konfigurationen . . . . .	249
7.5.2	Szenarien . . . . .	251
<b>8</b>	<b>Arbeiten mit dem <i>VIPtool</i></b>	<b>255</b>
8.1	Das Werkzeug . . . . .	255
8.1.1	Technische Details und Architektur . . . . .	255
8.1.2	Unterstützung des <i>VIPbusiness</i> -Ansatzes durch das <i>VIPtool</i> . . . . .	256
8.1.3	Verknüpfung mit anderen Werkzeugen . . . . .	259
8.2	Eine kleine Fallstudie . . . . .	261
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>269</b>
<b>A</b>	<b>Liste der anfänglich berücksichtigten Werkzeuge für den Entwurf von Geschäftsprozessen</b>	<b>273</b>
<b>B</b>	<b>Kurzbeschreibung der untersuchten Werkzeuge für den Geschäftsprozeß-Entwurf</b>	<b>279</b>



# Abbildungsverzeichnis

1.1	Ein 5-stufiger Ansatz für den Entwurf von Informationssystemen ([JS98] in Anlehnung an [Mar89]). . . . .	4
1.2	Verschiedene Arten von Geschäftsprozeßmodellen für verschiedene Sichtweisen. . . . .	4
1.3	Verschiedene Arten der Spezifikation für verschiedene Arten von Geschäftsprozessen. . . . .	8
1.4	Aufbau der Arbeit. . . . .	10
2.1	Abgrenzung verschiedener Typen von Geschäftsprozessen nach [Aal98d]. . .	16
2.2	Unterschiedlich strukturierte Geschäftsprozesse und ihre (technische) Unterstützung ([Sch00] nach [SGJ <sup>+</sup> 96]). . . . .	17
2.3	Revolution und Evolution beim Business Process Reengineering [Öst95b]. .	36
2.4	Lebenszyklus eines Geschäftsprozesses [Dei00]. . . . .	37
2.5	<i>SOM</i> -Rahmenkonzept [FS98]. . . . .	39
2.6	<i>ARIS</i> -Rahmenkonzept [Sch95]. . . . .	40
2.7	<i>EPKs</i> als Modell der Steuerungssicht [SJ96]. . . . .	40
2.8	Referenz-Architektur der WfMC [Law97] und ihre Unterstützung durch <i>VIPbusiness</i> . . . . .	42
2.9	Integration der Teilmodelle in INCOME [PRO01]. . . . .	44
2.10	Dimensionen von <i>Workflow Nets</i> [Aal98d]. . . . .	46
3.1	<i>VIPbusiness</i> : ein 3-stufiger Ansatz. . . . .	50
3.2	Das Röhren-Konzept. . . . .	53
3.3	<i>VIPbusiness</i> : Durchführung der drei Schritte. . . . .	57
3.4	<i>VIPbusiness</i> : Iteration. . . . .	59
3.5	Schema für die Darstellung der Werkzeug-Profile. . . . .	76
3.6	Profil des <i>VIPtools</i> . . . . .	83
4.1	Schritt 1 (I): Grundlagen. . . . .	85
4.2	Ein erster Entwurf des Geschäftsprozeßmodells als einfaches Netz. . . . .	87
4.3	Markierung. . . . .	89
4.4	Folgemarkierung. . . . .	89
4.5	Das Geschäftsprozeßmodell als Prädikate/Transitionen-Netz. . . . .	93
4.6	Schalten und Folgemarkierung in Prädikate/Transitionen-Netzen. . . . .	94
4.7	Externe Transition. . . . .	97
5.1	Schritt 1 (II): <i>VIPbusiness</i> -Netze. . . . .	99
5.2	Teilnetz und Vergrößerung. . . . .	103
5.3	Das Geschäftsprozeßmodell aus Abbildung 4.5 mit Aktivitäts-Transitionen. .	107

5.4	Sequenz von Aktivitäten. . . . .	109
5.5	AND-Split und AND-Join. . . . .	110
5.6	Impliziter OR-Split, impliziter OR-Join. . . . .	112
5.7	Expliziter OR-Split, expliziter OR-Join. . . . .	114
5.8	Spezielle Anwendung für den expliziten OR-Join. . . . .	115
5.9	Ein Zyklus. . . . .	117
5.10	Die zusammengesetzten Routing-Konstrukte. . . . .	118
5.11	Alle Routing-Konstrukte. . . . .	119
5.12	Geschäftsprozeßmodell mit Routing-Konstrukten. . . . .	120
5.13	Ressourcen-Nutzung via Kantenbeschriftung. . . . .	123
5.14	Ressourcen-Nutzung via interne Stelle. . . . .	124
5.15	Geschäftsprozeßmodell mit Ressourcen. . . . .	125
5.16	Zeit als Attribut von Stellen. . . . .	126
5.17	Zeit als Attribut von Kanten. . . . .	127
5.18	Zeit als Attribut von Transitionen - atomares Schalten. . . . .	127
5.19	Zeit als Attribut von Transitionen - Schalten in drei Schritten. . . . .	127
5.20	Modellierung von Ereignissen mit <i>Timed Coloured Petri Nets</i> . . . . .	128
5.21	Geschäftsprozeßmodell mit Zeit. . . . .	134
5.22	Geschäftsprozeßmodell mit Kosten. . . . .	137
5.23	Schema für die graphische Darstellung nicht-strukturierter Segmente. . . . .	139
5.24	Der nicht-strukturierte Teil des Geschäftsprozesses aus Abbildung 5.22: Startzustand, Zielzustand, Menge der Netzfragmente. . . . .	142
5.25	Spezielle Transitionen zur Modellierung von Business Rules. . . . .	146
5.26	Geschäftsprozeßmodell als <i>VIPbusiness</i> -Netz mit allen Erweiterungen. . . . .	150
6.1	Schritt 2: Simulation. . . . .	153
6.2	Geschäftsprozeßmodell aus Abbildung 4.5 mit vereinfachten Marken. . . . .	155
6.3	Ein halbgeordneter Ablauf des Geschäftsprozeßmodells aus Abbildung 6.2. . . . .	161
6.4	Begrifflichkeiten für halbgeordnete Abläufe. . . . .	162
6.5	Darstellung von Aktivitäts-Transitionen in Abläufen. . . . .	165
6.6	Darstellung von Routing-Konstrukten in Abläufen. . . . .	167
6.7	Nutzung von Ressourcen in Abläufen. . . . .	168
6.8	Ablauf mit Zeitbewertungen. . . . .	169
6.9	Ablauf mit Kostenbewertungen. . . . .	171
6.10	Nicht-strukturiertes Segment aus Abbildung 5.26 mit markiertem Startzustand. . . . .	172
6.11	Ablauf des nicht-strukturierten Segments aus Abbildung 6.10. . . . .	174
6.12	Ablauf des hybrid strukturierten <i>VIPbusiness</i> -Netzes aus Abbildung 5.26. . . . .	181
7.1	Schritt 3: Analyse und Optimierung. . . . .	183
7.2	Ablauf des Geschäftsprozeßmodells aus Abbildung 5.21 mit Zeitbewertungen. . . . .	193
7.3	Ablauf aus Abbildung 7.2 mit $t_{Go}$ -Werten und Wartezeiten. . . . .	200
7.4	Ablauf aus Abbildung 7.2 mit Ressourcen. . . . .	202
7.5	Ablauf aus Abbildung 7.2 mit Kostengrößen. . . . .	212
7.6	Kritischer Pfad. . . . .	218
7.7	Kostenoptimale Verteilung von Wartezeit für den Ablauf aus Abbildung 7.5. . . . .	221
7.8	Kostenbewerteter Ablauf aus Abbildung 7.5 mit Nutzung von Ressourcen. . . . .	222
7.9	Ausgangsablauf für die Linearisierungen von Ressourcen. . . . .	231



7.10	Ablauf mit Ressourcen und Anfangsmarkierung. . . . .	232
7.11	Linearisierte Version des Ablaufs aus Abbildung 7.10. . . . .	234
7.12	Bearbeitung mehrerer Cases ohne Ressourcen. . . . .	235
7.13	Ablauf aus Abbildung 7.12 mit Ressourcen und Anfangsmarkierung. . . . .	236
7.14	Linearisierte Version des Ablaufs aus Abbildung 7.13 (I). . . . .	238
7.15	Linearisierte Version des Ablaufs aus Abbildung 7.13 (II). . . . .	239
7.16	<i>VIPbusiness</i> -Netz mit Kontrollparametern und Ankunftsrate. . . . .	241
7.17	Abläufe des Geschäftsprozeß-Modells aus Abbildung 7.16. . . . .	245
7.18	Der Grad der Auslastung der Ressourcen-Klasse $r_2$ in Abhängigkeit von $cp_1$ und $cp_2$ . . . . .	247
7.19	Die durchschnittlichen Kosten pro Case in Abhängigkeit der Parameter $cp_1$ und $cp_2$ : die Farbe <i>weiß</i> entspricht dabei maximalen, <i>schwarz</i> minimalen Kosten. . . . .	248
8.1	Architektur des <i>VIPtools</i> . . . . .	256
8.2	<i>Editor</i> -Komponente des <i>VIPtools</i> , Modellierung einer Aktivitäts-Transition. . . . .	257
8.3	Menü zur Konfiguration der Simulations-Parameter. . . . .	258
8.4	Browser für die generierten Abläufe. . . . .	259
8.5	Oberfläche zum Erstellen und Editieren von Konfigurationen. . . . .	260
8.6	Architektur des <i>VIPtool/Woflan</i> -Toolkits [AE01]. . . . .	261
8.7	Ein erster Entwurf des Geschäftsprozesses. . . . .	263
8.8	Ein Ablauf des Geschäftsprozeßmodells aus Abbildung 8.7. . . . .	264
8.9	Linearisierungen durch gemeinsam genutzte Ressourcen. . . . .	265
8.10	Durchlaufzeiten für alle Abläufe und verschiedene Konfigurationen. . . . .	266



# Tabellenverzeichnis

3.1	Grundlegende Eigenschaften des Formalismus . . . . .	67
3.2	Berücksichtigte Dimensionen und ihre Integration. . . . .	68
3.3	Modellierung quantitativer Informationen. . . . .	70
3.4	Routing-Konstrukte und Zyklen. . . . .	70
3.5	Funktionalität der Simulation/Animation. . . . .	71
3.6	Möglichkeiten zur Leistungsbewertung. . . . .	75
6.1	Erreichbarkeitstabelle für das Geschäftsprozeßmodell in Abbildung 6.2. . . .	155
6.2	Mögliche Schaltfolgen für das Geschäftsprozeßmodell in Abbildung 6.2. . . .	156
7.1	Detaillierte Analyse der Zeitgrößen für Aktivitäten. . . . .	208
7.2	Detaillierte Analyse der Kostengrößen für Aktivitäten. . . . .	228
7.3	Detaillierte Analyse echter Wartezeit und echter Wartekosten. . . . .	230
7.4	Durchlaufzeit für Konfigurationen des Ablaufs aus Abbildung 7.2. . . . .	252



# Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) der Universität Karlsruhe (TH) und am Lehrstuhl für Angewandte Informatik der KU Eichstätt.

An dieser Stelle möchte ich mich bedanken.

Zu großem Dank verpflichtet bin ich meinem Doktorvater Prof. Dr. Jörg Desel, ohne dessen Vorschlag, bei ihm als wissenschaftlicher Mitarbeiter anzufangen, diese Arbeit vermutlich nie realisiert worden wäre. Seine Unterstützung bei wissenschaftlicher Arbeit und Publikationen sowie seine herausfordernden Anforderungen an präzisen und korrekten Schreibstil prägen Inhalt und Form dieser Arbeit.

Prof. Dr. Andreas Geyer-Schulz danke ich für die Übernahme des Zweitgutachtens und die in diesem Rahmen entstandenen Anregungen.

Prof. Dr. Wil van der Aalst danke ich für die Möglichkeit zur Zusammenarbeit, die Diskussionen und die dabei entstandenen Ideen.

Darüber hinaus danke ich dem Institutsleiter des AIFB, Herrn Prof. Dr. Wolffried Stucky sowie den Kolleginnen und Kollegen am AIFB und am Eichstätter Lehrstuhl für die inhaltliche, technische, moralische und sonstige Unterstützung. Ein besonderer Dank geht an Thomas Freytag für die Vorarbeiten zum *VIPtool* sowie an Martin Jäger für die kompetente Unterstützung bei der Implementierung der *VIPbusiness*-Erweiterungen.

Meinen Eltern danke ich für ihre Unterstützung und ihre Geduld.

Ein großes Dankeschön geht an alle, die mich während dieser Zeit auch außerhalb von Büros ertragen und in Form von Anregungen, Aufmunterungen und Korrekturen zum Gelingen dieser Arbeit beigetragen haben. Ihr wißt, wer Ihr seid.

Danke, Anja.



# Kapitel 1

## Einleitung

In dieser Arbeit wird mit *VIPbusiness* ein werkzeugunterstützter Ansatz für den Entwurf von Geschäftsprozessen vorgestellt. Spätestens seit den Arbeiten von Hammer und Champy [HC95] zu Beginn der 90er Jahre sind Geschäftsprozesse und ihre Gestaltung als entscheidender Faktor für den geschäftlichen Erfolg in das Bewußtsein der Unternehmen gerückt. Der Zwang zur Auseinandersetzung mit Geschäftsprozessen hat zu einem Bedarf an geeigneten Methoden und Werkzeugen zu deren Identifizierung, Analyse und Optimierung geführt. Die Grundlage bilden dabei Modelle von Geschäftsprozessen, die beispielsweise in verschiedenen Phasen der Ansätze des Business Process Reengineering oder des Process Improvement eine entscheidende Rolle spielen. In den frühen Arbeiten zu Geschäftsprozessen, die zum Großteil dem Gebiet der Organisationstheorie zuzurechnen sind, waren diese Modelle noch eher informelle Beschreibungen der betrachteten Geschäftsprozesse, deren Hauptfunktion in der expliziten Darstellung der Geschäftsprozesse bestand. Mit der Verknüpfung von Geschäftsprozessen und Informationssystemen haben sich die Anforderungen an Geschäftsprozeßmodelle jedoch entscheidend geändert.

### 1.1 Geschäftsprozeßmodelle als Schnittstelle beim Entwurf von Informationssystemen

In den letzten Jahren hat die Integration von Informationssystemen und Geschäftsprozessen für Unternehmen stark an Bedeutung gewonnen. Im Bereich der Informationssysteme hat ein Paradigmenwechsel stattgefunden: von der Unterstützung einzelner Aktivitäten hin zur Unterstützung gesamter Geschäftsprozesse. Zahlreiche Projekte, beispielsweise im Bereich der Einführung von *Standardsoftware* (auch: *Enterprise Resource Planning (ERP) Systems*) oder *Workflow Management Systemen (WFMS)* sind das Ergebnis dieser Entwicklung. Zentrale Bedeutung kommt bei diesen Projekten *deskriptiven Modellen* der zu unterstützenden Geschäftsprozesse zu. Als Ergebnis der frühen, anwendungsorientierten Projektphasen sind diese Modelle die Grundlage für den Beginn der Umsetzungs- und Implementierungsaufgaben, die in den späteren, systemorientierten Projektphasen im Vordergrund stehen (siehe Abbildung 1.1).

Diese Schnittstellenfunktion hat ein neues Anforderungsprofil für die beim Geschäftsprozeß-Entwurf verwendeten Methoden und Modelle zur Folge: die zum Teil sehr unterschiedlichen Bedürfnisse der Anwendungs- und der Systemseite müssen durch *ein* Modell abgedeckt werden. Üblicherweise, dem jeweiligen eher durch Betriebswirtschaft oder Informatik

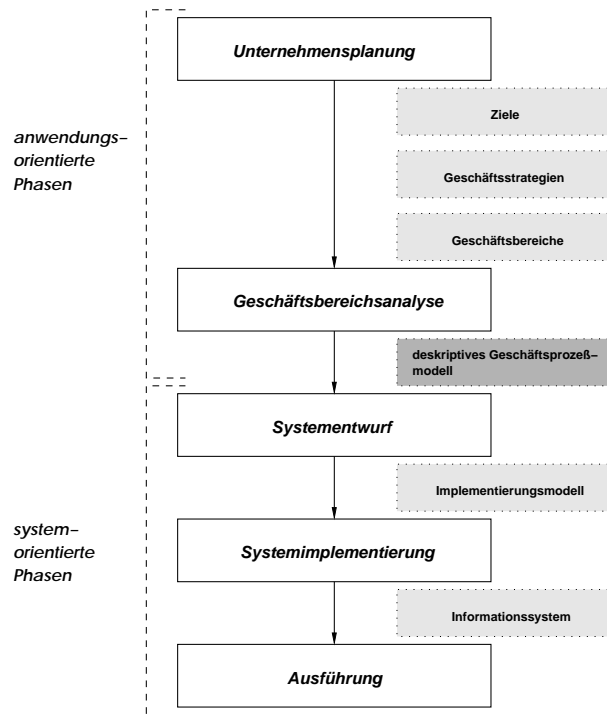


Abbildung 1.1: Ein 5-stufiger Ansatz für den Entwurf von Informationssystemen ([JS98] in Anlehnung an [Mar89]).

geprägten Hintergrund eines Anwenders entsprechend, kommt es hier zu einem Konflikt zwischen der Anwenderfreundlichkeit des verwendeten Modellierungsformalismus und den Möglichkeiten zur Analyse und formalen Interpretation der erstellten Modelle. Im Extremfall trifft dabei der Wunsch nach „Bildern“ auf programmiersprachliche Notationen. Ein Modellierungsformalismus, der an dieser Stelle verwendet werden soll, muß beide Seiten in geeignetem Maß zufriedenstellen können (vgl. Abbildung 1.2). Idealerweise eignet er sich dazu, die spezifischen Modelle der jeweiligen Seite ineinander zu überführen. Er stellt also nicht notwendigerweise das einzige im Verlauf des Entwicklungsprozesses verwendete Modell dar.

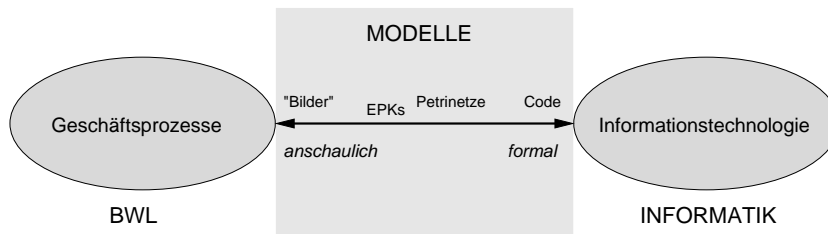


Abbildung 1.2: Verschiedene Arten von Geschäftsprozeßmodellen für verschiedene Sichtweisen.



Eine weitere wichtige Anforderung an verwendete Methoden und Formalismen ist die Unterstützung unterschiedlicher Grade der Detailliertheit eines Geschäftsprozeßmodells. Während es der Anwenderseite in erster Linie um ein Verstehen und damit um eine möglichst kompakte Darstellung der Logik eines Geschäftsprozesses geht, wird die Systemseite eine möglichst detaillierte Darstellung des Geschäftsprozesses verlangen, die als Ausgangspunkt für Weiterentwicklungen in Richtung Implementierungs-Grundlage verwendet werden kann. Hier ist ein geeigneter Kompromiß zu finden. Es ist wichtig festzuhalten, daß Geschäftsprozeßmodelle im Kontext des Entwurfs von Informationssystemen einen Detaillierungsgrad erreichen müssen, der eine Unterstützung durch ein Werkzeug zur Modellierung und Analyse erfordert. Papier und Bleistift reichen hier nicht mehr aus. Von großer Bedeutung ist die Möglichkeit, das potentielle Verhalten des Geschäftsprozeßmodells richtig beurteilen zu können. Dabei sind Möglichkeiten zur Simulation und zur Visualisierung des Verhaltens wichtige Hilfsmittel.

Seit den ersten Veröffentlichungen von Hammer und Champy [Ham90, HC93] sind zahlreiche Beiträge zum Thema Geschäftsprozesse erschienen. Bezüglich des Entwurfs von Geschäftsprozessen beschränken sich die meisten betriebswirtschaftlich geprägten Ansätze, vor allem aus dem Bereich des *Business Process (Re-)Engineering (BPR)* (vgl. beispielsweise [Dav93, JMPW93, MK96, MB93]) auf *qualitative* Aspekte (beispielsweise organisationale Aspekte oder Medienbrüche), *quantitative* Aspekte wie Durchlaufzeit, Kosten oder Ressourcenauslastung werden nicht berücksichtigt. Die bei der Spezifikation verwendeten Modellierungsformalismen sind in der Regel sehr einfach gehalten. Sie eignen sich damit sehr gut zur konzeptuellen Arbeit in frühen Phasen des Entwurfs. Bedauerlicherweise verfügen sie nur selten über eine formale Semantik, die jedoch die Grundlage für Möglichkeiten der Simulation und insbesondere der quantitativen Analyse darstellt. Viele Ansätze aus diesem Bereich beschränken sich sogar auf die Angabe einer mehr oder weniger einsichtigen Menge von *Regeln* für den Entwurf „guter“ Geschäftsprozesse [HC93, Dav93, PH96], ohne darauf einzugehen, wie oder womit solche Entwürfe realisiert werden könnten.

Bei vielen Ansätzen aus dem Bereich des *Workflow Management* (vgl. beispielsweise [JB96, KM95, Kou95, Law97]) stehen Fragestellungen technischer Art, beispielsweise Aspekte verteilter Systeme oder Integrität der Daten, im Vordergrund. Das Ziel beim Entwurf ist üblicherweise eine Spezifikation, die mehr oder weniger direkt für die Steuerungskomponenten entsprechender Systeme genutzt werden kann, beispielsweise für die *Engine* eines *Workflow Management Systems (WfMS)*. Da diese Spezifikationen entsprechend detailliert sein müssen und in manchen Fällen graphische Notationen bereits mit programmiersprachlichen Ergänzungen versehen werden, eignen sie sich nicht besonders für *frühe* Phasen des Entwurfs, in denen konzeptuelle Fragestellungen im Vordergrund stehen.

### **Ziel und Beitrag der Arbeit**

In der vorliegenden Arbeit wird mit *VIPbusiness*-Netzen eine Modellierungssprache für Geschäftsprozesse eingeführt, die Möglichkeiten der graphischen Modellierung mit einer formalen Semantik verbindet und somit den Anforderungen sowohl der anwendungsorientierten als auch der systemorientierten Seite genügt. Bei *VIPbusiness*-Netzen handelt es sich um eine neu entwickelte Klasse höherer Petrinetze. Die Eignung von Petrinetzen

oder verwandten Formalismen wie *Ereignisgesteuerten Prozeßketten (EPKs)* [Sch95] als Modellierungsfomalismus für Geschäftsprozesse ist ausführlich untersucht (vgl. beispielsweise [Aal98a, DO96, Uth01, Zel96]) und mit Hilfe einer Reihe auf Petrinetzen basierender Ansätze (vgl. beispielsweise [Aal98d, GK96, Obe96, Sch00, Wei00]<sup>1</sup>) erprobt worden.

Mit dem *VIPtool* steht ein Werkzeug für die Unterstützung des Entwurfs zur Verfügung. Die in dieser Arbeit vorgestellte Version des *VIPtools* basiert dabei auf einem im Rahmen des *VIP-Projekts*<sup>2</sup> erstellten Software-Prototyp. Durch Verwendung geeigneter Simulationskonzepte (die ebenfalls auf den *VIP*-Vorarbeiten aufbauen) ist es möglich, das Verhalten der Geschäftsprozeßmodelle in kompakter Form graphisch darzustellen. Dadurch wird das Verständnis des eigenen Entwurfs erheblich erleichtert.

## 1.2 Aspekte der Qualität von Geschäftsprozeßmodellen

Aufgrund ihrer Schnittstellenfunktion ist die Qualität der Geschäftsprozeßmodelle ein entscheidender Faktor für den Erfolg beim Entwurf von Informationssystemen. Um eine „Manifestierung“ schlecht entworfener Geschäftsprozesse durch ihre Verwendung als Implementierungs-Grundlage zu vermeiden, wird die Erstellung qualitativ hochwertiger Entwürfe der betroffenen Geschäftsprozesse, beispielsweise im Rahmen von *Business Process (Re-)Engineering (BPR)*-Maßnahmen, zu einem wichtigen (Teil-)Ziel früher Phasen entsprechender Projekte.

In der vorliegenden Arbeit wird ein werkzeugunterstützter Ansatz entwickelt, der die Erstellung qualitativ hochwertiger Geschäftsprozeß-Entwürfe in frühen Phasen entsprechender Projekte ermöglicht.

Der Begriff der *Qualität* eines Geschäftsprozeß-Entwurfs kann dabei durch die folgenden Aspekte konkretisiert werden<sup>3</sup>:

- Der Geschäftsprozeß-Entwurf ist *korrekt* in Bezug auf die Anwendung.  
Bei einem *korrekten* Entwurf entspricht das Verhalten des Geschäftsprozeßmodells dem Verhalten des intendierten Geschäftsprozesses. Notwendig ist in diesem Zusammenhang die *Validierung* und dabei insbesondere die *Visualisierung* des Geschäftsprozeß-Entwurfs und seines Verhaltens als Ausgangspunkt für eine Diskussion mit Experten der Anwendungs-Domäne. Dieser Punkt ist von entscheidender Bedeutung für die anwendungsorientierte Sicht auf den Geschäftsprozeß.
- Der Geschäftsprozeß-Entwurf ist *konsistent* in Bezug auf (logische) Anforderungen.  
Konsistenz bezieht sich auf die logische Struktur eines Geschäftsprozeß-Entwurfs. Die Ausführung des entworfenen Geschäftsprozesses sollte beispielsweise nicht in einem Deadlock enden, es sollte in jedem Fall zu einer Terminierung kommen usw. Die Konsistenz eines Geschäftsprozeß-Entwurfs kann mittels Verfahren aus dem Bereich der

---

<sup>1</sup>Diese Ansätze werden in Abschnitt 2.3.3 ausführlich diskutiert.

<sup>2</sup>Das Projekt *Verifikation von Informationssystemen durch Auswertung halbgeordneter Petrinetz-Abläufe* ist ein von der DFG gefördertes Kooperationsprojekt des Instituts AIFB der Universität Karlsruhe und des Instituts für Wirtschaftsinformatik der J.W.Goethe-Universität Frankfurt am Main. Inhaltliche Details finden sich in den Abschnitten 6.2 und 8.1.

<sup>3</sup>Eine detaillierte Diskussion dieser Aspekte sowie der dazugehörigen Qualitätsmerkmale findet sich in Abschnitt 2.2.

*Verifikation* überprüft werden. Die Konsistenz eines Geschäftsprozeß-Entwurfs spielt vor allem für die systemorientierte Sicht eine große Rolle, da ein nicht-konsistenter Geschäftsprozeß-Entwurf als Implementierungs-Grundlage zu schwerwiegenden Problemen zur Laufzeit des erstellten Systems führen kann.

- Der Geschäftsprozeß-Entwurf ist *effizient* in Bezug auf festgelegte Leistungsparameter.

Diese Anforderung bezieht sich auf Parameter zur Leistungsbewertung, beispielsweise aus den Bereichen *Zeit* und *Kosten*. Die Effizienz eines Geschäftsprozesses läßt sich durch (möglicherweise iterative) Anwendung entsprechender quantitativer Analyse-Methoden überprüfen. Dieser Aspekt ist unter anderem im Hinblick auf die Gewährleistung von *Service Levels*, beispielsweise in Form einer Garantie maximaler Bearbeitungs- oder Lieferzeiten für einen Kunden, von Bedeutung. Entscheidend ist an dieser Stelle, daß die Effizienz eines Geschäftsprozeß-Entwurfs schon in frühen Projektphasen, in jedem Fall vor Beginn der Implementierung, überprüft werden kann. Nur so läßt sich vermeiden, daß zur Laufzeit Performance-Probleme durch schlechte Geschäftsprozeß-Entwürfe entstehen.

Es ist offensichtlich, daß die aufgeführten Aspekte und die damit verbundenen Qualitätsmerkmale ganz unterschiedliche Anforderungen an die beim Entwurf der Geschäftsprozesse verwendeten Methoden und Formalismen stellen<sup>4</sup>. Zwar existieren für jeden der aufgeführten Aspekte Ansätze, die den spezifischen Anforderungen gerecht werden, üblicherweise werden die Aspekte jedoch getrennt betrachtet (vgl. dazu auch die Ausführungen in Abschnitt 2.3). Dies betrifft sowohl die verwendeten Modelle als auch die jeweils zur Verfügung stehenden Verfahren zur Überprüfung der entsprechenden Qualitätsmerkmale. So existiert beispielsweise eine Vielzahl an Werkzeugen zur graphischen Modellierung von Geschäftsprozessen. Diese bieten in der Regel jedoch nur wenig Möglichkeiten zur quantitativen Analyse und so gut wie keine Möglichkeiten zur Überprüfung der Konsistenz der Modelle. Umgekehrt fehlt vielen Verfahren aus dem Bereich der Verifikation ein graphischer Modellierungs-Formalismus, der die Modellierung von relevanten Aspekten von Geschäftsprozessen wie Zeit und Kosten erlauben würde.

## **Ziel und Beitrag der Arbeit**

In dieser Arbeit wird mit dem *VIPbusiness*-Ansatz eine Methode vorgestellt, die neben einem geeigneten Modellierungs-Formalismus eine systematische Analyse der erstellten Geschäftsprozeßmodelle bezüglich der Aspekte Korrektheit, Konsistenz und Effizienz ermöglicht. Eine besondere Rolle spielen dabei Möglichkeiten zur Simulation der Geschäftsprozesse. Das Simulations-Konzept sowie Verfahren zur Analyse werden ebenfalls durch das *VIPtool* unterstützt.

## **1.3 Entwurf von nicht vollständig strukturierten Geschäftsprozessen**

Üblicherweise wird bei der Erstellung eines Geschäftsprozeßmodells versucht, den gesamten Geschäftsprozeß zu erfassen, indem die relevanten Aktivitäten für die Bearbeitung

---

<sup>4</sup>Welche Anforderungen mit welchen Analysemöglichkeiten verbunden sind, wird in Abschnitt 2.2.4 untersucht.

eines Cases<sup>5</sup> identifiziert und in eine feste Reihenfolgebeziehung gebracht werden. Falls es verschiedene Möglichkeiten gibt, Aktivitäten auszuführen, werden einige (in manchen Fällen auch alle) Alternativen in das Geschäftsprozeßmodell integriert. Dasselbe wird für bekannte Ausnahmesituationen (auch: *known exceptions*) versucht. Das Ergebnis dieser Vorgehensweise ist in vielen Fällen ein Geschäftsprozeßmodell, das zwar recht kompliziert ist, jedoch immer noch nicht alle Möglichkeiten der Ausführung von Aktivitäten beinhaltet. Vielmehr stellt es eine mehr oder weniger idealisierte Version des Geschäftsprozesses dar. Je weniger strukturiert ein Geschäftsprozeß ist, desto größer wird der Unterschied zwischen der Anzahl möglicher Variationen zur Bearbeitung realer Fälle und der Anzahl der Möglichkeiten, die im Modell des Geschäftsprozesses vorgesehen wurden. Der Hauptgrund für diese Diskrepanz ist die Schwierigkeit, einen nicht vollständig strukturierten Geschäftsprozeß durch eine Spezifikation des gesamten möglichen Verhaltens zu beschreiben. Für diese Art von Geschäftsprozessen führt der oben beschriebene *konstruktive* Entwurfsansatz (obwohl für strukturierte Geschäftsprozesse völlig geeignet) zu „aufgeblasenen“ Modellen, die schon für Geschäftsprozesse mit wenigen Aktivitäten schwer zu handhaben sind. Obwohl es sinnvoll erscheint, unterschiedliche Möglichkeiten der Spezifikation für unterschiedliche Arten von Geschäftsprozessen zu verwenden (vgl. Abbildung 1.3), haben sich insbesondere im Bereich des werkzeugunterstützten Entwurfs von Geschäftsprozessen ausschließlich konstruktive Ansätze etabliert. Der Entwurf von nicht vollständig strukturierten Geschäftsprozessen ist mit diesen Werkzeugen jedoch nur auf sehr umständliche Weise oder gar nicht möglich. Da nur die wenigsten Geschäftsprozesse vollständig strukturiert sind (vgl. dazu die Ausführungen in Abschnitt 2.1.2), ergeben sich daraus große Nachteile beim Entwurf vieler Geschäftsprozesse. Zwar befinden sich nicht vollständig strukturierte Geschäftsprozesse im Blickfeld des *Computer Supported Cooperative Work (CSCW)* und der dazugehörigen *Groupware*-Systeme und -Anwendungen, dabei steht jedoch nicht mehr der *Prozeß*, sondern die zur Bewältigung einer gemeinsam (kooperativ) zu erbringenden Aufgabe benötigte oder produzierte *Information* im Vordergrund [KM95, Kou95].

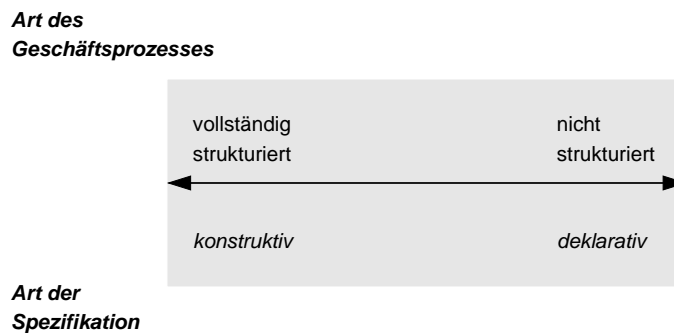


Abbildung 1.3: Verschiedene Arten der Spezifikation für verschiedene Arten von Geschäftsprozessen.

<sup>5</sup>Der Begriff *Case* wird in Abschnitt 2.1.1 erläutert. Er wird in dieser Arbeit synonym zu den deutschen Begriffen *Geschäfts(vor)fall* oder *Fall* verwendet.

## Ziel und Beitrag der Arbeit

In dieser Arbeit wird zur Modellierung der nicht-strukturierten Teile von Geschäftsprozessen ein *deklarativer* Ansatz vorgeschlagen, bei dem die Aktivitäten eines nicht-strukturierten Teilprozesses als Menge von *Fragmenten* spezifiziert werden. Das mögliche Verhalten der nicht-strukturierten Teile wird durch Regeln, sogenannte *Business Rules*, eingeschränkt. Im Unterschied zum rein konstruktiven Vorgehen wird beim deklarativen Ansatz also nur das *nicht* erlaubte Verhalten des Modells explizit spezifiziert.

Der *VIPbusiness*-Ansatz unterstützt damit den Entwurf von nicht vollständig strukturierten Geschäftsprozessen. Bei der Erstellung entsprechender Geschäftsprozeßmodelle können konstruktive und deskriptive Modellierungselemente zu *hybriden* Spezifikationen kombiniert werden. Die so erstellten Geschäftsprozesse sind weiterhin ausführbar, sämtliche vorgestellten Verfahren zur Analyse können ohne Einschränkung angewendet werden.

## 1.4 Aufbau und einzelne Beiträge der Arbeit

In diesem Abschnitt werden der weitere Aufbau sowie die jeweiligen Beiträge der Arbeit in den verschiedenen Kapiteln umrissen. Durch die eher ausführliche Schilderung der Beiträge sowie direkte Verweise auf Abschnitte der Arbeit, in denen Neuerungen diskutiert werden, soll insbesondere Lesern mit Vorkenntnissen das gezielte Erfassen der in dieser Arbeit enthaltenen neuen Konzepte ermöglicht werden. Aus diesem Grund werden an dieser Stelle eine ganze Reihe von Begriffen verwendet, deren genaue Bedeutung sich Lesern ohne Vorkenntnisse erst bei der detaillierten Einführung in den jeweiligen Kapiteln erschließen dürfte. Abbildung 1.4 zeigt den Aufbau der Arbeit in einer Übersicht.

In Kapitel 2 werden zunächst einige grundlegende Begriffe in Bezug auf Geschäftsprozesse eingeführt. Darauf aufbauend wird in Abschnitt 2.2 erörtert, welche Merkmale von Geschäftsprozessen zur Beurteilung der Qualität eines Geschäftsprozeß-Entwurfs bezüglich der Aspekte Korrektheit, Konsistenz und Effizienz geeignet sind. Von großer Bedeutung sind die sich daraus ergebenden Anforderungen an den verwendeten Modellierungsformalismus. In Abschnitt 2.3 werden bekannte Ansätze für den Entwurf von Geschäftsprozessen vorgestellt und hinsichtlich der Anforderungen untersucht.

Die dabei identifizierten Defizite führen zur Einführung des 3-stufigen *VIPbusiness*-Ansatzes in Kapitel 3. Dabei werden die drei Hauptschritte *Modellierung*, *Simulation* und *Analyse und Optimierung* in ein Rahmenkonzept gebettet, das die Erstellung eines Geschäftsprozeß-Entwurfs sowie die systematische Verbesserung eines ersten Entwurfs zu einem qualitativ hochwertigen Entwurf unterstützt. Als besondere Vorteile des Ansatzes sind neben der Integration aller relevanten Qualitäts-Aspekte vor allem die Effizienz und die Vergleichbarkeit der Analyse-Ergebnisse bei Durchführung von Simulationen zu nennen. Details zu den Vorteilen des Ansatzes sowie die Arbeit mit *VIPbusiness* finden sich in Abschnitt 3.1.4. Der zweite Teil des Kapitels ist der Erarbeitung eines Anforderungsprofils für die Werkzeugunterstützung des *VIPbusiness*-Ansatzes gewidmet. Dieses Profil wird genutzt, um bekannte Werkzeuge zur Geschäftsprozeßmodellierung zu untersuchen. Den Abschluß des Kapitels bildet das Profil des *VIPtools*.

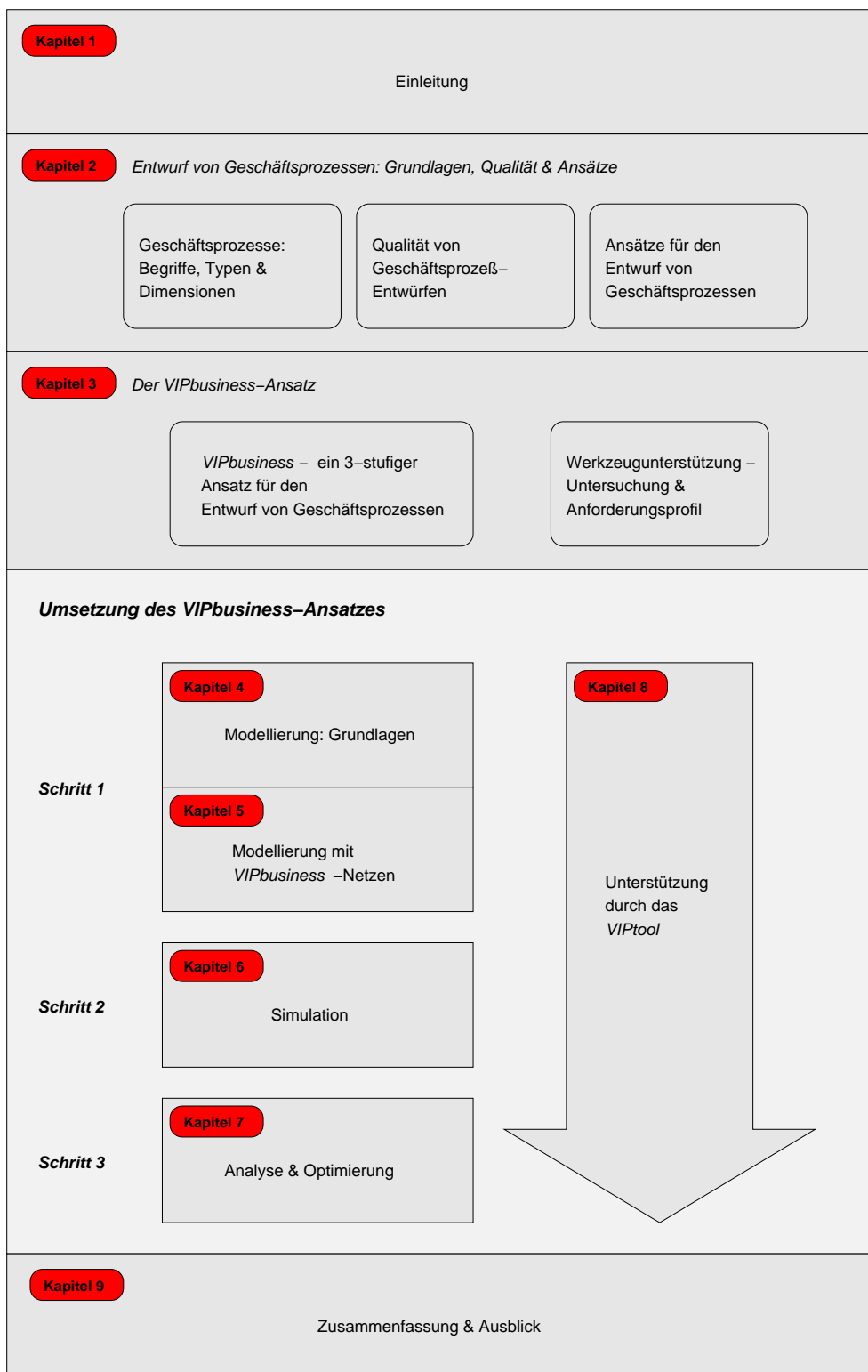


Abbildung 1.4: Aufbau der Arbeit.

Ausgehend von der Vorstellung des *VIPbusiness*-Ansatzes werden in den folgenden Kapiteln alle zur Realisierung der drei Stufen des Ansatzes benötigten Konzepte eingeführt.

In den Kapiteln 4 und 5 wird erläutert, wie mit Hilfe des *VIPbusiness*-Ansatzes Modelle von Geschäftsprozessen erstellt werden können. Dazu wird mit *VIPbusiness*-Netzen eine neue Klasse höherer Petrinetze vorgestellt. Diese Netzklasse zeichnet sich durch eine Reihe spezieller Eigenschaften aus, die im Detail in Abschnitt 3.1.1 erläutert werden. Von besonderer Bedeutung sind in diesem Zusammenhang die Einführung eines ausschließlich zustandsbasierten Zeit- und Kostenkonzepts sowie die Möglichkeit zur deklarativen Spezifikation von nicht-strukturierten Segmenten von Geschäftsprozessen durch Netzfragmente und Business Rules.

Da alle im Rahmen des *VIPbusiness*-Ansatzes entwickelten Modellierungs-Konzepte auf „klassische“ Petrinetz-Konzepte zurückgeführt werden können, bleibt die formale Semantik und damit die Ausführbarkeit der so erstellten Geschäftsprozeßmodelle erhalten. Wie das Verhalten von Geschäftsprozeßmodellen durch Simulation untersucht werden kann, wird in Kapitel 6 beschrieben. Durch Verwendung von halbgeordneten Abläufen und Kausalnetzen kann das Verhalten der Geschäftsprozeßmodelle auf kompakte Weise anschaulich dargestellt werden. Welche Besonderheiten das *VIPbusiness*-Simulations-Konzept sonst noch birgt, wird in Abschnitt 3.1.2 erläutert.

In Kapitel 7 wird gezeigt, wie sich die Geschäftsprozeßmodelle im Rahmen des *VIPbusiness*-Ansatzes bezüglich der eingeführten Qualitätsmerkmale untersuchen lassen. Welche Verfahren dazu entwickelt wurden, wird im Detail in Abschnitt 3.1.3 beschrieben. Dabei decken die Verfahren alle drei Qualitäts-Aspekte Korrektheit, Konsistenz, Effizienz und die damit verbundenen Analyse-Bereiche Validierung, Verifikation und Leistungsbewertung ab.

Der *VIPbusiness*-Ansatz wird durch das *VIPtool* unterstützt. In Kapitel 8 wird neben dem grundsätzlichen Aufbau des *VIPtools* und seiner Verknüpfung mit anderen Werkzeugen gezeigt, wie durch Verwendung des *VIPtools* qualitativ hochwertige Entwürfe von Geschäftsprozessen erstellt werden können.

Eine Zusammenfassung der Ergebnisse sowie ein Ausblick auf mögliche Ansatzpunkte für weitere Arbeiten in Kapitel 9 beenden die Arbeit.





## Kapitel 2

# Entwurf von Geschäftsprozessen: Grundlagen, Qualität und Ansätze

Ein Blick in die Literatur zu Geschäftsprozessen zeigt, daß die Terminologie in diesem Bereich alles andere als einheitlich verwendet beziehungsweise interpretiert wird [VB96]. In Abschnitt 2.1 werden daher zunächst einige grundlegende *Begriffe* festgelegt, auf die im weiteren Verlauf der Arbeit zurückgegriffen werden kann. Darüber hinaus wird in diesem Abschnitt untersucht, welche *Typen* von Geschäftsprozessen unterschieden werden können.

Mit den Merkmalen zur Analyse von Geschäftsprozessen verhält es sich ähnlich wie mit den grundlegenden Begriffen. In den verschiedenen Veröffentlichungen zu diesem Thema findet sich eine große Anzahl unterschiedlicher Merkmale, die zum Teil auch noch uneinheitlich interpretiert werden (beispielsweise der Begriff *Durchlaufzeit*). In Abschnitt 2.2 wird daher versucht, aus der Vielzahl von Veröffentlichungen solche Merkmale herauszufiltern, die für die Beurteilung der Qualität eines Geschäftsprozeß-Entwurfs im Sinne des eingangs dieser Arbeit bereits erläuterten Qualitätsbegriffs sinnvoll erscheinen. Die Möglichkeit zur Analyse dieser Merkmale stellt Anforderungen an die verwendeten Formalismen und Methoden. Welche Merkmale mit welchen Anforderungen verbunden sind, wird in Abschnitt 2.2.4 untersucht.

Ausgehend von dem aus diesen Betrachtungen resultierenden Anforderungsprofil werden in Abschnitt 2.3 bekannte Ansätze für den Entwurf von Geschäftsprozessen vorgestellt und hinsichtlich der Erfüllung dieses Profils untersucht. Ein Schwerpunkt wird dabei auf Ansätze gelegt, die auf Modellen von Geschäftsprozessen basieren.

### 2.1 Grundlagen: Begriffe und Typen

Bevor im weiteren Verlauf dieser Arbeit der Entwurf von Geschäftsprozessen behandelt werden kann, sind in diesem Abschnitt noch einige grundlegende Punkte zu klären. Dazu werden in Abschnitt 2.1.1 zunächst grundlegende Begriffe, beispielsweise der des *Geschäftsprozesses* selbst, in Anlehnung an in der Literatur verbreitete Auffassungen festgelegt.

Je nach Unternehmen und betrachtetem Unternehmensbereich können sich Geschäftsprozesse stark unterscheiden. Welche Merkmale zur Abgrenzung verschiedener *Typen* von Geschäftsprozessen verwendet werden und welche Typen sich aufgrund dieser Abgrenzung ergeben, wird in Abschnitt 2.1.2 beschrieben.

### 2.1.1 Begriffe

Seit der Feststellung in [VB96], daß „die Gebiete der Geschäftsprozeßmodellierung (...) derzeit noch zu sehr in der Diskussion [sind], als daß diese Begriffe bereits endgültig und abschließend definiert werden könnten“, sind einige Jahre vergangen, einheitliche Begriffe für das Gebiet der Geschäftsprozesse existieren jedoch immer noch nicht. Zwar sind beispielsweise im Zuge der Standardisierungs-Versuche im Bereich des *Workflow Management* durch die *Workflow Management Coalition (WfMC)* [WFM96, Law97] eine Reihe von „Standards“ veröffentlicht worden, von diesen hat sich jedoch keiner durchgesetzt. Aus diesem Grund muß zunächst ein begrifflicher Rahmen abgesteckt werden, der im weiteren Verlauf dieser Arbeit zugrunde gelegt werden kann. Aus der Vielzahl existierender Abgrenzungen für die relevanten Begriffe sei dabei lediglich auf bekannte grundlegende Arbeiten auf diesem Gebiet verwiesen<sup>1</sup>, beispielsweise Hammer und Champy [HC95], Davenport [Dav93], Ferstl und Sinz [FS93], Scheer [Sch95] oder Österle [Öst95b]. In dieser Arbeit wird ein Geschäftsprozeß-Begriff verwendet, der im wesentlichen auf den Ausführungen in [Obe96] und [Aal98d] beruht:

Ein *Geschäftsprozeß* besteht aus *Aktivitäten*, die in einem Betrieb nach bestimmten Regeln auf ein bestimmtes Ziel hin ausgeführt werden. Zur Ausführung der Aktivitäten werden *Ressourcen* benötigt. Geschäftsprozesse sind *fallbasiert*<sup>2</sup>, d.h., die Ausführung von Aktivitäten dient der Bearbeitung von *Cases*.

Beispiele für *Geschäftsprozesse* sind etwa die Bearbeitung von Schadensmeldungen in einer Versicherung [Aal98d], die Abwicklung des Einkaufs in einem Industriebetrieb [HC95] oder die Eröffnung eines Bankkontos [Mül95]. Zur Ausführung von Aktivitäten benötigte *Ressourcen* können dabei beispielsweise Maschinen, Materialien oder Mitarbeiter sein. Ein *Case* ist beispielsweise eine konkrete Schadensmeldung, die bearbeitet werden muß.

Statt des Begriffs *Case* werden in der Literatur neben dem ebenfalls englischen Begriff *Business Case* auch die Begriffe *Fall* oder *Geschäfts(vor)fall* benutzt. In dieser Arbeit wird der am weitesten verbreitete Begriff *Case* verwendet. Dies hat den zusätzlichen Vorteil, daß bei der Verwendung des Wortes *Fall* in Wendungen wie „Für den Fall, daß...“ keine Mißverständnisse entstehen.

Synonym zu *Geschäftsprozeß* werden auch die Begriffe *betrieblicher Ablauf*, *Workflow* oder *Workflow-Prozeß*<sup>3</sup> sowie *Prozeß*<sup>4</sup> verwendet (vgl. [Obe96] oder [Uth01]).

Wie in obiger Definition bereits erwähnt, ist die Bearbeitung von *Cases* mit der Ausführung von *Aktivitäten* verknüpft [Aal98d]. Eine *Aktivität* hat Vor- und Nachbedingungen. Vor der Ausführung der Aktivität gelten die Vorbedingungen, nach Beendigung der Ausführung die Nachbedingungen. Aus den jeweils gültigen Vor- und Nachbedingungen aller Aktivitäten eines Geschäftsprozesses läßt sich der Stand der Bearbeitung von Fällen ableiten.

---

<sup>1</sup>Eine detaillierte Diskussion und Abgrenzung bekannter *Geschäftsprozeß*-Begriffe findet sich beispielsweise in [Sch00] oder [Uth01].

<sup>2</sup>vgl. [Aal98d], dort wird der Begriff *case-based* verwendet.

<sup>3</sup>Dies ist vor allem in Arbeiten aus dem Bereich des *Workflow Management* der Fall. [Sch00] differenziert die Begriffe *Workflow* und *Geschäftsprozeß* über die dedizierte Computerunterstützung von Workflows.

<sup>4</sup>Dies trifft vor allem auf die betriebswirtschaftliche Organisationslehre zu (vgl. beispielsweise [PB95] oder [KK92]).

In der Regel setzt sich ein Geschäftsprozeß aus mehreren (oft vielen) Aktivitäten zusammen. Bei der Bearbeitung von Cases müssen diese Aktivitäten koordiniert werden. Ressourcen, die zur Ausführung von Aktivitäten benötigt werden, müssen diesen zugeordnet werden. Um bei diesen Aufgaben die Übersicht zu behalten, ist die Verwendung eines Modells für den Entwurf, die Kontrolle und die Verbesserung eines Geschäftsprozesses zu empfehlen.

In einem *Geschäftsprozeßmodell* wird beschrieben, welche Aktivitäten in welcher Reihenfolge zur Bearbeitung eines Case ausgeführt werden müssen und welche Ressourcen bei der Ausführung benötigt werden.

An einigen Stellen wird in dieser Arbeit der Begriff der *Spezifikation* eines Geschäftsprozesses synonym zu *Geschäftsprozeßmodell* verwendet. Weitere Synonyme für Geschäftsprozeßmodell sind beispielsweise die Begriffe *Geschäftsprozeß-definition*, *Geschäftsprozeßbeschreibung* oder *Geschäftsprozeß-Schema*.

Ausgehend von der Festlegung der wichtigsten Grundbegriffe wird im nächsten Abschnitt untersucht, welche *Typen* von Geschäftsprozessen unterschieden werden können.

### 2.1.2 Typen

Reale Geschäftsprozesse unterscheiden sich in vielfacher Hinsicht, beispielsweise bezüglich der *Häufigkeit* der Durchführung oder des mit der Durchführung verbundenen *Zeit-* und *Kostenverbrauchs*. Zur Identifizierung unterschiedlicher *Typen* von Geschäftsprozessen lassen sich unterschiedliche Merkmale verwenden<sup>5</sup>. Ein solches Unterscheidungsmerkmal kann beispielsweise die Bedeutung eines Geschäftsprozesses für das Kerngeschäft oder die Kunden einer Unternehmung sein. Im Hinblick auf die Modellierung von Geschäftsprozessen mit Petrinetzen interessiert in dieser Arbeit vor allem die *Formalisierbarkeit* eines Geschäftsprozesses. Mit Formalisierbarkeit soll hier die Möglichkeit der Darstellung eines Geschäftsprozesses in einem (formalen) Modell gemeint sein. Je nach untersuchtem Unternehmensbereich können sich die Geschäftsprozesse in ihrer Formalisierbarkeit stark unterscheiden. So ist die Fertigung eines Werkstücks an einem Fließband um einiges einfacher zu modellieren als ein Entscheidungsprozeß der Geschäftsleitung. Bei letzterem stellt sich die grundsätzliche Frage, ob ein solcher Geschäftsprozeß überhaupt in einem Modell abgebildet werden kann und soll.

Die in dieser Arbeit verwendete Kategorisierung von Geschäftsprozessen orientiert sich an [Aal98d] (vgl. Abbildung 2.1). Als Unterscheidungsmerkmale dienen der *Strukturierungsgrad* eines Geschäftsprozesses, sowie die Frage, ob die Durchführung des Geschäftsprozesses im wesentlichen durch dessen Struktur festgelegt ist (*prozeßbetont*) oder tendenziell auf dem Austausch von Informationen beruht (*informationsbetont*). Mit *Strukturierungsgrad* ist dabei der Anteil der vorhersagbaren beziehungsweise sich wiederholenden Bestandteile eines Geschäftsprozesses gemeint (vgl. [GHS95, SM96, Wei00]). Ein strukturierter Geschäftsprozeß verfügt über einen hohen Anteil der genannten Bestandteile.

---

<sup>5</sup>Entsprechende Merkmale finden sich etwa in [Obe96] oder [HC95].

Das Ergebnis dieser Kategorisierung ist eine Unterscheidung zwischen Geschäftsprozessen, die im Rahmen der Geschäftsprozeßmodellierung behandelt werden können (*workflow processes*), und solchen, bei denen dies aufgrund ihrer schweren Formalisierbarkeit nicht möglich oder nicht sinnvoll ist (*collaborative processes*). Die Gruppe der *workflow processes* kann zudem noch in die Bereiche *production workflow*, *administrative workflow* und *ad-hoc workflow* unterteilt werden.

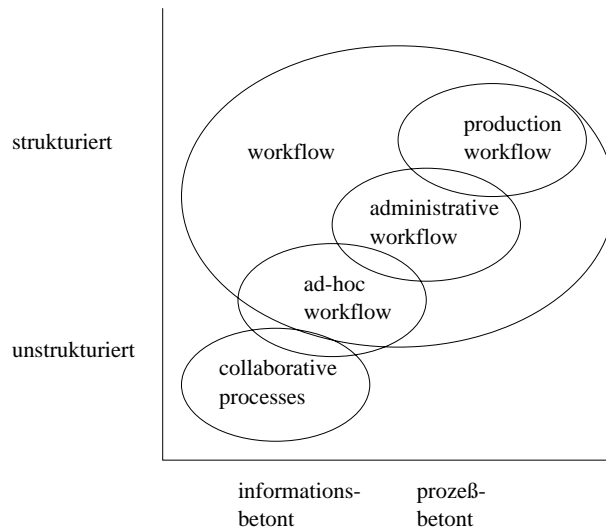


Abbildung 2.1: Abgrenzung verschiedener Typen von Geschäftsprozessen nach [Aal98d].

Geschäftsprozesse, die dem Bereich *production workflow* zuzuordnen sind, zeichnen sich durch ihren hohen Grad an Strukturiertheit und ein geringes Maß an Variation aus. Dabei handelt es sich meist um Abläufe aus den Bereichen Fertigung oder Produktion (zum Beispiel Herstellung eines Fahrzeugs) oder um stark standardisierte Abläufe aus dem Bereich der Büroarbeit, beispielsweise die Annahme eines Kundenauftrags durch einen Sachbearbeiter.

Geschäftsprozesse des *administrative workflow* sind durch ein höheres Maß an Nicht-Determinismus geprägt. Je nach Ausprägung der in der Praxis auftretenden Fälle kann der Geschäftsprozeß unterschiedlich verlaufen, jedoch nur in einem vorher festgelegten Rahmen. Als Ursachen für den Nicht-Determinismus der Ausführung lassen sich die folgenden Punkte ausmachen (vgl. [Sch99] und [DE00]):

- Bei der Bearbeitung von Fällen müssen *Auswahlentscheidungen* getroffen werden. Als Beispiel sei hier die Bearbeitung einer Schadensmeldung durch einen Sachbearbeiter in einer Versicherung genannt, bei der eine detaillierte Prüfung nur stattfindet, wenn der Wert des Schadens hoch genug ist (ansonsten erfolgt die Regulierung direkt, siehe auch [Aal98d]).

- Die Reihenfolge der Ausführung der Aktivitäten ist nicht eindeutig festgelegt, die Ausführung ist also *reihenfolgevariabel*.
- Aktivitäten können in verschiedenen *Modi* ausgeführt werden. Beispielsweise kann die Kontaktaufnahme mit einem Kunden telefonisch oder per Post erfolgen. Diese Art der Ausführung wird im weiteren Verlauf dieser Arbeit als *modusvariabel* bezeichnet. Die (optionale) *Nicht-Ausführung* einer Aktivität stellt dabei einen speziellen Modus dar.

Geschäftsprozesse, die eine oder mehrere der oben geschilderten Eigenschaften aufweisen, werden im weiteren Verlauf dieser Arbeit als *nicht vollständig strukturierte* Geschäftsprozesse bezeichnet. Entscheidendes Merkmal von Geschäftsprozessen diesen Typs ist die *hybride* Zusammensetzung aus strukturierten und nicht strukturierten Teilen [DE00].

Fehlt ein festgelegter Rahmen für die Durchführung, liegt ein Geschäftsprozeß aus dem Bereich *ad-hoc workflow* vor. Für jeden Case muß der Ablauf des Geschäftsprozesses neu festgelegt oder ein bereits existierender Ablauf neu definiert werden. [Obe96] nennt an dieser Stelle *unvorhergesehene Sonderfälle* (auch: *known exceptions*) von Abläufen als Beispiel.

Im Rahmen dieser Arbeit sind vor allem Geschäftsprozesse der Bereiche *production workflow* und *administrative workflow* von Interesse. Zwar können die im weiteren Verlauf eingeführten Konzepte auch zur Unterstützung von Geschäftsprozessen aus dem Bereich *ad-hoc workflow*, beispielsweise zu deren Visualisierung, verwendet werden, die Vorteile der entwickelten Simulations- und Analysemöglichkeiten lassen sich jedoch vor allem bei zumindest teilweise strukturierten und quantitativ bewertbaren Geschäftsprozessen nutzen.

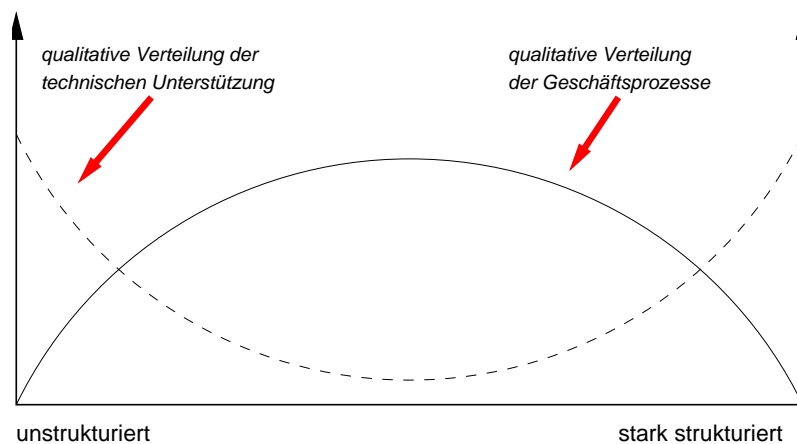


Abbildung 2.2: Unterschiedlich strukturierte Geschäftsprozesse und ihre (technische) Unterstützung ([Sch00] nach [SGJ<sup>+</sup>96]).

Bei der Entwicklung des in dieser Arbeit vorgestellten *VIPbusiness*-Ansatzes wurde ein besonderes Augenmerk auf die Unterstützung des Entwurfs von nicht vollständig strukturierten Geschäftsprozessen gelegt. Obwohl die meisten realen Geschäftsprozesse diesem Typ zuzuordnen sind (vgl. Abbildung 2.2), unterstützen bisher existierende Ansätze für den Entwurf von Geschäftsprozessen vor allem vollständig strukturierte<sup>6</sup> oder unstrukturierte<sup>7</sup> Geschäftsprozesse. Aus diesem Grund werden im weiteren Verlauf dieser Arbeit Konzepte entwickelt, die eine explizite Unterstützung des Entwurfs nicht vollständig strukturierter Geschäftsprozesse durch den *VIPbusiness*-Ansatz ermöglichen (vgl. Abschnitte 3.1, 5.6 und 6.3).

## 2.2 Qualität von Geschäftsprozeß-Entwürfen

Im Hinblick auf das Ziel der vorliegenden Arbeit, einen werkzeugunterstützten Ansatz zur Erstellung qualitativ hochwertiger Modelle von Geschäftsprozessen zu entwickeln, soll an dieser Stelle die Frage diskutiert werden, wie die Qualität eines Geschäftsprozesses beurteilt werden kann. Bedauerlicherweise ist der Begriff der *Qualität* eines Geschäftsprozesses noch schwerer zu fassen als der Begriff des Geschäftsprozesses selbst. Dies hängt mit der ebenfalls fehlenden einheitlichen Terminologie zusammen und wird durch die Vielfalt möglicher relevanter Merkmale noch verstärkt. Das Spektrum der in bekannten Arbeiten betrachteten Merkmale reicht dabei von den bekannten *Entwurfsempfehlungen* für Geschäftsprozesse von Hammer und Champy [HC95] bis hin zu detaillierten, teilweise empirisch ermittelten Empfehlungen, beispielsweise zur Auslastung von Ressourcen. Die Berücksichtigung sämtlicher Qualitätsmerkmale, die in den verschiedenen Arbeiten zum Thema genannt werden, ist allein aufgrund ihrer Anzahl nicht möglich und angesichts der unterschiedlichen Zielsetzungen der jeweiligen Arbeiten auch nicht sinnvoll. In diesem Abschnitt werden daher eine Reihe von Merkmalen ausgewählt, die im Hinblick auf die Beurteilung von Geschäftsprozeß-Entwürfen durch Analyse der zugehörigen Geschäftsprozeßmodelle besonders sinnvoll erscheinen. Dabei sind insbesondere Aspekte der *Operationalisierung* von Merkmalen zu beachten, insbesondere müssen die Merkmale *modellierbar* sein. Dies ist vor allem für *qualitative*<sup>8</sup> Merkmale wie beispielsweise die in [HC95] geforderte *sinnvolle Anordnung der Aktivitäten* leider oft nicht möglich. Weitere Einschränkungen ergeben sich aus dem Umstand, daß die in dieser Arbeit vorgestellten Konzepte in möglichst frühen Phasen des Entwurfs von Geschäftsprozessen zum Einsatz kommen sollen. Beurteilungen eines Entwurfs bezüglich solcher Merkmale wie *Zufriedenheit der Kunden* [HC95] oder *Motivation der Mitarbeiter* [Dav93] lassen sich zu diesem Zeitpunkt nur schwer durchführen.

Auf typische im Kontext der Messung von Kenngrößen auftretende Probleme wie *Dimensionalität* (verschiedene Nutzenfaktoren sollen aggregiert werden), *Bewertung* (unterschiedlich Interessengruppen bewerten Merkmale nach unterschiedlichen Kriterien) oder *Kausalität* (gleiche Merkmalsausprägungen können in unterschiedlichen Kontexten unterschiedliche Konsequenzen haben) soll in dieser Arbeit nicht weiter eingegangen werden. Der interessierte Leser sei an dieser Stelle beispielsweise auf die Ausführungen in [Bec99] verwiesen.

---

<sup>6</sup>Dies trifft vor allem für Ansätze aus dem Umfeld des *Workflow Management* zu [Sch00].

<sup>7</sup>Dieser Typ von Geschäftsprozeß ist Gegenstand der Ansätze aus dem Umfeld des *Computer Supported Cooperative Work (CSCW)* beziehungsweise der *Groupware* [Sch00].

<sup>8</sup>Der Begriff *qualitativ* ist an dieser Stelle als Abgrenzung zu *quantitativ* zu verstehen.

Wie in Kapitel 1 bereits erwähnt, stützt sich der in dieser Arbeit verwendete Qualitätsbegriff für Geschäftsprozesse auf drei zentrale Anforderungen, denen jeder Entwurf eines Geschäftsprozesses gerecht werden sollte:

- Korrektheit in Bezug auf die Anwendung.
- Konsistenz in Bezug auf (logische) Anforderungen.
- Effizienz in Bezug auf festgelegte Leistungsparameter.

Wie in Kapitel 1 schon angedeutet, korrespondiert die Aufteilung des Begriffs der Qualität eines Geschäftsprozesses in die drei Aspekte Korrektheit, Konsistenz und Effizienz mit der Kategorisierung von Analyseverfahren in die Bereiche *Validierung*, *Verifikation* und *Leistungsbewertung*.

Die Trennung der Begriffe *Validierung* und *Verifikation* in der Literatur ist nicht einheitlich. Der Hauptunterschied wird im allgemeinen darin gesehen, daß es sich bei der Verifikation um den formalen Nachweis von Eigenschaften im Sinne eines mathematischen Beweises handelt, während bei der Validierung eine oft nicht weiter erläuterte „Überprüfung“ der Korrektheit des Modells in Bezug auf die Anwendung erfolgt (vgl. [Aal98d] oder [Obe96]). Falls das Modell ausführbar ist, ist neben der Validierung des Modells selbst auch die Validierung des Verhaltens des Modells von Interesse. Dabei wird das (beispielsweise im Rahmen einer Simulation generierte) Verhalten des Modells im Hinblick auf eine Übereinstimmung mit realem oder intendiertem Verhalten überprüft [Des00].

Bei der *Leistungsbewertung*<sup>9</sup> wird anhand des Geschäftsprozeßmodells die Leistungsfähigkeit des Geschäftsprozesses analysiert. Dazu werden in der Regel quantitative Merkmale aus den Bereichen *Zeit* und *Kosten* herangezogen. Um die Leistungsfähigkeit eines Geschäftsprozesses beurteilen zu können, muß das Verhalten im Geschäftsprozeßmodell untersucht werden. Dabei werden Möglichkeiten zur Simulation im Modell benötigt (vgl. [Aal98d], [Rum99] oder [Uth01]).

In den folgenden Abschnitten wird untersucht, welche Merkmale zur Beurteilung der Qualität eines Geschäftsprozeß-Entwurfs bezüglich aller drei beschriebenen Qualitätsaspekte herangezogen werden können. Dazu werden in den Abschnitten 2.2.1 und 2.2.2 zunächst Merkmale für die Bereiche *Korrektheit/Validierung* und *Konsistenz/Verifikation* betrachtet, anschließend in Abschnitt 2.2.3 Merkmale für den Bereich *Effizienz/Leistungsbewertung* mit einem Fokus auf die Teilbereiche *Zeit* und *Kosten*.

Abhängig von den gewünschten Möglichkeiten zur Analyse eines Geschäftsprozeß-Entwurfs sowie den dabei berücksichtigten Merkmalen ergeben sich unterschiedliche Anforderungen an den bei der Modellierung verwendeten Formalismus sowie unterstützende Werkzeuge. Welche Analysemöglichkeiten welche Anforderungen nach sich ziehen, wird in Abschnitt 2.2.4 beschrieben.

---

<sup>9</sup>[Aal98d] verwendet an dieser Stelle den Begriff *performance analysis*.

### 2.2.1 Qualität (I): Korrektheit und Validierung

Wie im vorangegangenen Abschnitt bereits erläutert, soll durch Validierung eines Geschäftsprozeß-Entwurfs überprüft werden, ob das mögliche Verhalten des Geschäftsprozeß-Entwurfs dem Verhalten des intendierten Geschäftsprozesses entspricht. Wichtigstes Hilfsmittel ist dabei die *Visualisierung*, also die graphische Darstellung des Verhaltens. Die Visualisierung des Verhaltens setzt voraus, daß Möglichkeiten zur Simulation des Geschäftsprozeßmodells bestehen. Ein Sonderfall der Visualisierung des Systemverhaltens stellt die *Animation* dar: die Ausführung von Aktivitäten wird während der Simulation des Geschäftsprozeßmodells graphisch dargestellt (beispielsweise durch „Blinken“ der zur Modellierung von Aktivitäten verwendeten Symbole).

Neben der reinen Visualisierung können auch Fragestellungen folgender Art Gegenstand der Validierung eines Geschäftsprozesses sein<sup>10</sup>:

- Sind die Aktivitäten und ihre Reihenfolge sinnvoll im Hinblick auf das Ziel des Geschäftsprozesses?
- Ist der Geschäftsprozeß wirklich an den Bedürfnissen der Kunden ausgerichtet?
- Enthält der Geschäftsprozeß unnötige Überprüfungen und Kontrollen?
- Kommt es bei der Ausführung des Geschäftsprozesses zu *Organisationsbrüchen*?

Ein Organisationsbruch liegt vor, wenn an der Ausführung des Geschäftsprozesses unterschiedliche organisatorische Bereiche des Unternehmens beteiligt sind [HC95].

- Kommt es bei der Ausführung des Geschäftsprozesses zu *Medienbrüchen*?  
Nach [FS96] treten Medienbrüche bei einem Wechsel des für die Kommunikation bei der Durchführung des Geschäftsprozesses benutzten Übertragungsmediums auf (beispielsweise beim Übergang von Papierform zu elektronischer Form).
- Lassen sich Aktivitäten parallel ausführen, um die Durchlaufzeit des Geschäftsprozesses zu verringern?
- Welche Aktivitäten werden am häufigsten durchgeführt?

Diese Fragestellungen lassen sich anhand *qualitativer* Merkmale von Geschäftsprozessen beantworten. Als bekanntes Beispiel seien hier die *Entwurfsempfehlungen für Geschäftsprozesse* aus [HC95] genannt:

- Aktivitäten in natürlicher Reihenfolge anordnen,
- zusammengehörige Aktivitäten nicht unterteilen,
- wenn möglich, mehrere Schritte kombinieren,
- wenn möglich, Aktivitäten parallel ausführen,
- Prozesse möglichst einfach halten,

---

<sup>10</sup> Ausführliche Erläuterungen zu diesen Fragestellungen und den dabei relevanten Merkmalen finden sich beispielsweise in [HC95], [Mat93], [RS96], [KL96] oder [Jae96].



- Überprüfungen und Kontrollen reduzieren,
- generische Prozesse für mehrfache Verwendung erzeugen.

Da beispielsweise eine *natürliche Reihenfolge der Aktivitäten* nur schwer durch ein ausschließlich rechnergestütztes Analyseverfahren beurteilt werden kann, dürfte für die Analyse von Geschäftsprozessen bezüglich dieser Merkmale die Diskussion durch Experten anhand eines geeigneten Modells die beste Methode sein. Möglichkeiten zur Simulation und Visualisierung sind dabei jedoch ein wertvolles Hilfsmittel (vgl. Abschnitt 3.1.4).

Weitere Merkmale, deren Beurteilung in erster Linie von der subjektiven Einschätzung des Betrachters eines Geschäftsprozesse abhängt, sind<sup>11</sup>:

- Qualität der Information,
- Aktualität und Transparenz des Informationssystems,
- Funktionalität benutzter Software,
- Kundenorientierung,
- Zusammenführung der Lenkung für eine bestimmte Leistung („klare Kompetenzen“).

Anders verhält es sich für Merkmale, die mit der Identifizierung und Aufzählung bestimmter Phänomene in Geschäftsprozessen zusammenhängen<sup>12</sup>:

- Grad der Parallelität bei Ausführung der Aktivitäten,
- Mehrfacherfassung von Daten<sup>13</sup>,
- Anzahl der Organisationsbrüche,
- Anzahl der Medienbrüche.

Derartige Merkmale können bei der Modellierung von Geschäftsprozessen berücksichtigt und mittels geeigneter Analyseverfahren untersucht werden.

### 2.2.2 Qualität (II): Konsistenz und Verifikation

Bei der Verifikation eines Geschäftsprozeß-Entwurfs wird dessen Konsistenz bezüglich logischer Anforderungen überprüft. Dies geschieht durch den formalen Nachweis von Eigenschaften des Geschäftsprozeßmodells. Gegenstand der Untersuchung können dabei sowohl Eigenschaften der *Struktur* als auch Eigenschaften des *Verhaltens* des entsprechenden Geschäftsprozeßmodells sein.

---

<sup>11</sup> Ausführliche Erläuterungen dieser Merkmale finden sich beispielsweise in [Mat93], [RS96] oder [FS96].

<sup>12</sup> Details dazu finden sich beispielsweise in [HC95], [FS96], [Jae96], [PRO01] oder [Bre97].

<sup>13</sup> Auch als *Datenredundanz* [Mat93] bezeichnet.

Beispiele für entsprechende Fragestellungen sind<sup>14</sup>:

- Kann es zu einer „Selbstblockade“<sup>15</sup> des Geschäftsprozesses kommen?
- Treten unerwünschte Zustände des Geschäftsprozesses auf?  
Unerwünscht kann beispielsweise der Zustand sein, daß eine Zahlung bereits erfolgt ist, eine notwendige Prüfung jedoch noch nicht.
- Existieren Aktivitäten, die nie ausgeführt werden?
- Werden Reihenfolgebeziehungen zwischen Aktivitäten eingehalten?

In [Aal98d] wird das Kriterium der *Soundness* eines Geschäftsprozeßmodells als „Minimalanforderung“ an die Konsistenz eines Geschäftsprozeß-Entwurfs eingeführt. Die Soundness eines Geschäftsprozeß-Entwurfs setzt die Erfüllung folgender Anforderungen voraus:

- Es existieren ein gültiger Anfangszustand sowie ein gültiger Endzustand.
- Für alle Aktivitäten sind Vor- und Nachbedingungen definiert. Es existieren keine isolierten Aktivitäten.
- Für jeden Case ist die Bearbeitung irgendwann abgeschlossen, der Endzustand also erreicht. Dabei sind ausschließlich die Bedingungen des Endzustands erfüllt.
- Es existieren keine Aktivitäten, die bei der Bearbeitung der Fälle nie ausgeführt werden.

Ein weit verbreitetes Hilfsmittel zur Analyse der Konsistenz eines Geschäftsprozeß-Entwurfs ist die Formulierung und Überprüfung sogenannter *Business Rules*<sup>16</sup> [Her96, HK96]. Eine Business Rule kann beispielsweise die Festlegung sein, daß ein bestimmtes Dokument (zum Beispiel ein Kreditantrag) stets von einer bestimmten Person (beispielsweise einem Vorgesetzten des Sachbearbeiters) überprüft werden muß. Im Prinzip können alle oben genannten Fragestellungen wie die Überprüfung hinsichtlich Selbstblockaden oder nie ausgeführter Aktivitäten als Business Rules aufgefaßt werden. Der entscheidende Punkt beim Entwurf von Geschäftsprozessen ist die Möglichkeit, Business Rules nicht nur allgemein, sondern als gewünschte Eigenschaften des erstellten Geschäftsprozeßmodells formulieren zu können<sup>17</sup>. Im Idealfall kann dann mit Hilfe geeigneter Analysemethoden anhand des Geschäftsprozeßmodells oder des im Rahmen einer Simulation generierten Verhaltens überprüft werden, ob eine Business Rule bei der Bearbeitung von Fällen immer, nie oder nur manchmal verletzt wird.

---

<sup>14</sup>Weitergehende Erläuterungen zu diesen und weiteren Fragestellungen aus dem Bereich *Konsistenz/Verifikation* finden sich beispielsweise [Rol96],[Aal98d], [Jae96], [PRO01] oder [Bre97].

<sup>15</sup>Oft auch als *Deadlock* [Aal98d] bezeichnet.

<sup>16</sup>Manchmal auch als *Geschäftsregeln* bezeichnet. Auch wenn Anglizismen in dieser Arbeit weitestgehend vermieden werden sollen, hat sich hier der englische Begriff in der Literatur durchgesetzt.

<sup>17</sup>Welche Auswirkungen diese zunächst einfach zu erfüllende Anforderung hat, wird in Abschnitt 2.2.4 erläutert

### 2.2.3 Qualität (III): Effizienz und Leistungsbewertung

Die Effizienz eines Geschäftsprozeß-Entwurfs kann durch die Anwendung von Verfahren zur *Leistungsbewertung* überprüft werden. Im Gegensatz zu den in den beiden vorangegangenen Abschnitten geschilderten Verfahren zur Validierung und Verifikation werden *quantitative* Merkmale, vor allem aus den Bereichen *Zeit* und *Kosten* untersucht.

Für die dabei relevanten Merkmale werden in diesem Abschnitt die folgenden Fragestellungen untersucht:

- Welche *Kenngrößen* gibt es?

Da für die einzelnen Kenngrößen wie beispielsweise *Durchlaufzeit* oder *Wartezeit* ähnlich wie für den Begriff *Geschäftsprozeß* in der Literatur durchaus unterschiedliche Interpretationen existieren, erfolgt zunächst eine Festlegung der verwendeten Begriffe.

- Welche *Bezugsgrößen* können verwendet werden?

Den einzelnen Kenngrößen können unterschiedliche Bezugsgrößen zugrunde gelegt werden. So lassen sich Kosten beispielsweise für die Ausführung einer einzelnen Aktivität, für die einmalige Durchführung eines Geschäftsprozesses oder für die Durchführung eines Auftrags ermitteln. Sowohl für Zeit- als auch für Kostengrößen wird betrachtet, welche Bezugsgrößen berücksichtigt werden.

- Welche *Ziele* werden bei der Analyse verfolgt?

Häufiges Ziel bei der Analyse ist die *Ermittlung* von Kenngrößen, die bei der Modellierung des Geschäftsprozesses noch nicht bekannt sind. Ein Beispiel hierfür ist die Ermittlung der Durchlaufzeit. Die Ermittlung läßt sich für unterschiedliche Bezugsgrößen durchführen. So kann beispielsweise die Durchlaufzeit der einmaligen Durchführung eines Geschäftsprozesses oder die Durchlaufzeit eines Auftrags von Interesse sein. Die Ermittlung von Kenngrößen kann als Grundlage zur Beantwortung weiterer Fragestellungen genutzt werden. Von Interesse sind dabei beispielsweise die Ermittlung von Durchschnittsgrößen oder Möglichkeiten der Optimierung.

Abschließend wird kurz auf Möglichkeiten für *weitergehende Fragestellungen* eingegangen, die auf der Analyse von Zeit- und Kostengrößen aufbauen, beispielsweise bezüglich der Auslastung oder der Nichtverfügbarkeit von *Ressourcen*.

### Quantitative Merkmale (I): Zeit

Merkmale zur Zeitbewertung von Geschäftsprozessen lassen sich in verschiedene Kategorien einteilen.

- nicht-deterministische Zeitgrößen

Die Werte von *nicht-deterministischen* Zeitgrößen hängen von zufälligen *Störeinflüssen* auf die Durchführung eines Geschäftsprozesses ab. Solche Einflüsse können beispielsweise der unerwartete Ausfall eines Mitarbeiters oder einer Maschine sein. Der *VIPbusiness*-Ansatz unterstützt die Analyse von Geschäftsprozeß-Entwürfen mit Hilfe zufällig erzeugter Zeitgrößen. Damit lassen sich beispielsweise *Worst Case*- oder *Best Case*-Szenarien bezüglich der Durchlaufzeit für die Bearbeitung eines Case überprüfen. Ein entsprechendes Verfahren wird in Abschnitt 7.5 vorgestellt.

- deterministische Zeitgrößen

In der Gruppe der *deterministischen* Zeitgrößen muß zwischen nicht-variablen und variablen Größen unterschieden werden.

Die Werte der *nicht-variablen* Zeitgrößen eines Geschäftsprozesses zeichnen sich dadurch aus, daß sie von der jeweiligen Ausführung des Geschäftsprozesses unabhängig sind. Ein Beispiel für eine solche Zeitgröße ist etwa die Bearbeitungszeit eines Werkstücks innerhalb einer Aktivität des Geschäftsprozesses. Wird bei der Durchführung des Geschäftsprozesses die entsprechende Aktivität ausgeführt, fällt der (bekannte oder geschätzte) Wert für die Bearbeitung des Werkstücks als Zeitverbrauch an. In der Realität (insbesondere bei nicht voll-automatisierten Aktivitäten) ist die Bearbeitungszeit für je zwei Werkstücke natürlich nie genau gleich. Für die Modellierung wird daher eine *Sollzeit* angesetzt. Die Werte für Sollzeiten ergeben sich beispielsweise aus der Ermittlung von Durchschnittszeiten (bei der Modellierung bereits existierender Abläufe) oder durch Schätzungen.

Sind bei der Ausführung eines Geschäftsprozeßmodells Auswahlentscheidungen zu treffen, kann das Modell auf unterschiedlichen Wegen durchlaufen werden. Für die Ausführung der Aktivitäten auf diesen Wegen werden die oben beschriebenen Sollzeiten als nicht-variable Zeitgrößen angesetzt.

Eine wichtige *variable* Zeitgröße in Geschäftsprozeßmodellen ist die echte Wartezeit, die beispielsweise bei der Synchronisation unabhängig ausführbarer Aktivitäten oder durch Nichtverfügbarkeit von Ressourcen entsteht. Da die Werte für diese Zeitgröße mit unterschiedlichen Ausführungen des Geschäftsprozeßmodells (beispielsweise durch unterschiedlich getroffene Auswahlentscheidungen) variieren, können bei der Modellierung keine Sollwerte verwendet werden.

Ausgehend von obigen Erläuterungen werden im nächsten Abschnitt die Kenngrößen aus dem Bereich *Zeit* festgelegt, die im weiteren Verlauf dieser Arbeit bei der Erstellung und Analyse der Petrinetz-Modelle von Geschäftsprozessen Berücksichtigung finden.

## **Kenngrößen**

Zentrale Größe bei der Zeitbewertung von Geschäftsprozessen ist die *Durchlaufzeit* (vgl. [Aal98d], [PRO01], [DF96], [Zen69] oder [DIN96]). Da sich ein Geschäftsprozeß aus Aktivitäten und Zuständen zwischen Aktivitäten zusammensetzt (vgl. Abschnitt 2.1), wird die Durchlaufzeit für die Ausführung eines Geschäftsprozeßmodells zur Bearbeitung von Cases durch den Zeitverbrauch für die Ausführung der entsprechenden Aktivitäten sowie eventuell anfallende Wartezeit bestimmt.

Die *Durchlaufzeit eines Geschäftsprozesses* bezeichnet damit die Zeitspanne zwischen Beginn und Ende der Bearbeitung eines Case gemäß Geschäftsprozeßmodell. Sie ergibt sich aus:

- Durchlaufzeit der Aktivitäten,
- Wartezeit zwischen der Ausführung von Aktivitäten.

Die *Durchlaufzeit einer Aktivität* hängt von den Teil-Aktivitäten ab, die zur Ausführung einer Aktivität nötig sind. In der Literatur gibt es bezüglich der Zusammensetzung der Durchlaufzeit einer Aktivität aus den Zeitbewertungen der Teil-Aktivitäten eine Reihe verschiedener Ansätze (vgl. [SO95], [Bes86], [Gab94], [FHS83] oder [Zen69]). In Anlehnung an [SO95] wird die Durchlaufzeit einer Aktivität im Rahmen dieser Arbeit wie folgt festgelegt:

Die *Durchlaufzeit einer Aktivität* bezeichnet die Zeitspanne, die zwischen Beginn und Ende der Aktivität verstreicht. Die Durchlaufzeit einer Aktivität wird durch die Summe der Zeitbewertungen der zur Ausführung der Aktivität erforderlichen Teil-Aktivitäten bestimmt:

- Transportzeit

Die *Transportzeit*<sup>18</sup> [Bre97, Cha96, Reu95] bezeichnet den Zeitbedarf für Ortsveränderungen des Arbeitsgegenstandes.

Falls Störungen des Transportvorgangs nicht berücksichtigt werden, handelt es sich bei der Transportzeit um eine deterministische, nicht-variable Zeitgröße, die im Geschäftsprozeßmodell mit einer Sollzeit angesetzt wird. Zur Berücksichtigung möglicher Störungen werden in dieser Arbeit Zufallswerte für diese Sollzeiten verwendet, Transportzeit also auch als nicht-deterministische Zeitgröße behandelt.

- Rüstzeit

Die *Rüstzeit*<sup>19</sup> [Bre97] umfaßt den Zeitverbrauch, der anfällt, um ein Arbeitssystem vorzubereiten und nach Durchführung der Arbeit gegebenenfalls in den ursprünglichen Zustand zurückzusetzen.

Die Rüstzeit wird ebenfalls als nicht-deterministische oder deterministische, nicht-variable Zeitgröße angesehen und mit entsprechenden zufälligen oder festen Sollwerten belegt.

- Bearbeitungszeit

Die *Bearbeitungszeit*<sup>20</sup> [Jae96, Bre97] bezeichnet die Zeitspanne, die für die Bearbeitung des jeweiligen Gegenstandes der Arbeit benötigt wird.

Zur Berücksichtigung von zufälligen Einflüssen auf die Bearbeitungszeit durch Störungen (beispielsweise Stromausfall) sowie Schwankungen der Bearbeitungszeit aufgrund unterschiedlicher Leistungsgrade der Ressourcen (beispielsweise „Mitarbeiter am Montag“) kann die Bearbeitungszeit ebenfalls mit zufälligen Werten belegt werden. Bei Abstraktion von diesen Störgrößen kann die Bearbeitungszeit auch als deterministische, nicht-variable Zeitgröße betrachtet werden.

- fertigungsbedingte Wartezeit.

Die *fertigungsbedingte Wartezeit* bezeichnet die Zeitspanne, die durch planmäßiges Warten auf das Ende eines Arbeitsschritts verstreicht.

Ein Beispiel für fertigungsbedingte Wartezeit ist die Zeit, die ein lackierter Gegenstand zum Trocknen benötigt (vorausgesetzt, das Trocknen ist mit keiner speziellen

---

<sup>18</sup> Auch als *Transferzeit* [Bau97] bezeichnet.

<sup>19</sup> Auch als *Vorbereitungszeit* [Jae96, Bau97]) oder *Umrüstzeit* [Jae96] bezeichnet.

<sup>20</sup> Auch als *Fertigungszeit* [Reu95] oder *Ausführungszeit*[Bau97] bezeichnet.

Arbeit verbunden). Fertigungsbedingte Wartezeit kann ebenfalls als nicht-variable Zeitgröße oder variable Zeitgröße berücksichtigt werden.

Diese Sicht auf die Bestandteile der Durchlaufzeit hat ihren Ursprung in der Betrachtung von Geschäftsprozessen aus dem Bereich der Fertigung [Bec99, FHS83, Zen69]. Für Geschäftsprozesse, bei denen nicht das Erstellen physischer Produkte, sondern das Erbringen von Dienstleistungen im Vordergrund steht, ist diese Sicht oft umständlich beziehungsweise nicht zutreffend. So erweist sich beispielsweise die Transportzeit beim Transport von Informationen im Verhältnis zur Bearbeitungszeit als nicht relevante Größe, Zeiten für das Umrüsten von Maschinen für verschiedene Produkte fallen in dieser Form auch nicht an. Aus diesem Grund wird für Aktivitäten, die zum Erbringen einer Dienstleistung ausgeführt werden (im folgenden als *Dienstleistungs-Aktivitäten* bezeichnet), eine vereinfachte Sicht auf die Bestandteile der Durchlaufzeit eingeführt. Dabei werden Transportzeit und Rüstzeit zu *Vorbereitungszeit* zusammengefaßt, die fertigungsbedingte Wartezeit entfällt.

Die Durchlaufzeit einer Dienstleistungs-Aktivität wird damit durch die Summe der Zeitbewertungen der folgenden zur Ausführung der Aktivität erforderlichen Teil-Aktivitäten bestimmt:

- Vorbereitungszeit,
- Bearbeitungszeit.

Falls für die einzelnen Bestandteile ausnahmslos nicht-variable Zeitgrößen verwendet worden sind, handelt es sich bei der Durchlaufzeit für Aktivitäten ebenfalls um eine nicht-variable Zeitgröße, deren Wert unmittelbar aus den Sollwerten für die Zeitbewertungen der Teil-Aktivitäten abgeleitet werden kann.

Im Idealfall können alle Aktivitäten des Geschäftsprozesses schnellstmöglich ausgeführt werden, ohne daß es zu Verzögerungen kommt. In diesem Fall würde sich die Durchlaufzeit zur Bearbeitung eines Case unmittelbar aus den Sollzeiten für die Durchlaufzeit der Aktivitäten ergeben. In der Realität wird es jedoch trotz aller Bemühungen bei der Gestaltung der Geschäftsprozesse zu folgenden Szenarien kommen:

- Für den Zusammenbau eines Fertigprodukts werden mehrere Komponenten benötigt, deren parallel laufende Fertigung unterschiedlich viel Zeit in Anspruch nimmt. Bevor mit dem Zusammenbau begonnen werden kann, muß auf die Fertigstellung der Komponente mit der längsten Fertigungszeit gewartet werden.
- Zwei Aktivitäten benötigen zu ihrer Ausführung dieselbe Ressource (beispielsweise einen bestimmten Mitarbeiter). Eine der Aktivitäten muß warten, bis die andere Aktivität ausgeführt (und damit die Ressource wieder freigegeben) worden ist.

In beiden Fällen kommt es zu *nicht-fertigungsbedingter Wartezeit*, die im folgenden als *echte Wartezeit* bezeichnet werden soll.

Echte Wartezeit ist nicht mit der Ausführung von Aktivitäten verknüpft. Sie läßt sich vielmehr mit den Vor- und Nachbedingungen der Aktivitäten assoziieren, da sie nach der Ausführung einer Aktivität vor Beginn der nächsten Aktivität entsteht.

Die *echte Wartezeit* bezeichnet die Zeitspanne, die durch nicht-fertigungsbedingtes Warten zwischen der Ausführung zweier Aktivitäten entsteht.

Mit nicht-fertigungsbedingtem Warten ist in der Regel keine Wertschöpfung verbunden, es entstehen lediglich Kosten. Daher ist die Vermeidung echter Wartezeit ein wichtiges Ziel bei der Gestaltung von Geschäftsprozessen.

Da echte Wartezeit einer ihrer Bestandteile ist, handelt es sich bei der Durchlaufzeit eines Geschäftsprozesses ebenfalls um eine variable Zeitgröße, die mit der Ausführung des Geschäftsprozesses variieren kann.

Die Durchlaufzeit stellt auch in dieser Arbeit das wichtigste Merkmal aus dem Bereich *Zeit* dar. Dementsprechend werden in Abschnitt 5.4 Konzepte eingeführt, die es ermöglichen, die zur Analyse der Durchlaufzeit benötigten Zeitgrößen von Geschäftsprozessen mit Petrinetzen zu modellieren.

### **Bezugsgrößen**

Die Bezugsgrößen bei der *Modellierung* von Zeitgrößen sind die Aktivitäten eines Geschäftsprozesses beziehungsweise die Vor- und Nachbedingungen der Aktivitäten. Dabei werden Transport-, Rüst-, Bearbeitungs- und fertigungsbedingte Wartezeit den entsprechenden Teil-Aktivitäten zugeordnet, während die echte Wartezeit für die Vor- und Nachbedingungen der Aktivitäten modelliert wird.

Bei der *Analyse* von Geschäftsprozeß-Entwürfen sind die wichtigsten Bezugsgrößen für die oben geschilderten Zeitgrößen die *Bearbeitung eines Case* und die *Ausführung einer Aktivität*. Wie bereits angedeutet, ist die *Ermittlung der Durchlaufzeit* für diese beiden Bezugsgrößen im Rahmen der Leistungsbewertung von Geschäftsprozessen von großer Bedeutung. Im Prinzip können bei der Analyse jedoch beliebige Bezugsgrößen gewählt werden, genau genommen bedarf es lediglich der Festlegung zweier Bezugspunkte. Solche Bezugspunkte können neben den bereits erwähnten

- Beginn und Ende der Bearbeitung eines Case,
- Beginn und Ende der Ausführung einer Aktivität

auch

- Beginn und Ende des Einsatzes einer Ressource zur Ausführung einer Aktivität,
- Beginn und Ende eines Zyklus im Geschäftsprozeß oder
- Beginn und Ende der Bearbeitung eines Auftrags (zum Beispiel Erzeugung einer bestimmten Anzahl an Fertigprodukten)

sein.

## Ziele

Ebenso wie für Kenngrößen und Bezugsgrößen gibt es auch für die Ziele bei der Analyse von Geschäftsprozessen bezüglich Zeitaspekten viele Möglichkeiten. Voraussetzung für jede weiterführende Analyse ist jedoch zunächst die Ermittlung aller variablen Zeitgrößen, die erst im Rahmen der Simulation des Verhaltens von Geschäftsprozessen mit Werten belegt werden können (zum Beispiel echte Wartezeit). Nach der Ermittlung aller relevanten Zeitgrößen können beispielsweise Durchschnittszeiten gebildet oder Möglichkeiten zur Optimierung der Zeitgrößen untersucht werden.

Mit der Einführung von Zeitgrößen können nun die entsprechenden Kostengrößen betrachtet werden.

## Quantitative Merkmale (II): Kosten

In [Bes86] werden *Kosten* als „bewerteter Güter- und Dienstleistungsverzehr zur Erstellung und Verwertung betrieblicher Leistungen“ eingeführt. Demnach ist der *Kostenbegriff* durch vier Merkmale definiert:

- Es muß ein *mengenmäßiger* Verbrauch (Verzehr) an Gütern und/oder Dienstleistungen vorliegen.
- Dieser Verzehr an Gütern oder Dienstleistungen muß *bewertet* werden.
- Der bewertete Verzehr muß erfolgen, um eine *betriebliche Leistung* zu erzeugen.
- Die betriebliche Leistung muß für die *Verwertung* am Markt bestimmt sein.

Im Rahmen der *Kostenrechnung* wird traditionell zwischen *Kostenartenrechnung*, *Kostenstellenrechnung* und *Kostenträgerrechnung* unterschieden (vgl. beispielsweise [KS97] oder [Fre86]). Dabei werden jeweils unterschiedliche Fragestellungen beantwortet:

- *Welche* Kosten sind angefallen?

Die *Kostenartenrechnung* erfaßt alle in einer Abrechnungsperiode anfallenden Kosten nach Kostenarten [KS97]. Beispiele für Kostenarten sind *Materialkosten* oder *Personalkosten*. Welche Kostenarten in dieser Arbeit berücksichtigt werden, wird in Abschnitt „Kenngrößen“ erläutert.

- *Wo* sind die Kosten angefallen ?

Die *Kostenstellenrechnung* nimmt eine verursachungsgerechte Aufteilung der Kosten auf Kostenstellen vor. Traditionell handelt es sich bei Kostenstellen um organisatorische Einheiten einer Unternehmung (beispielsweise Einkauf oder Produktion). Neuere Ansätze zur Kostenrechnung wie die *Prozeßkostenrechnung* verwenden als Kostenstelle „unteilbare“ Teilprozesse eines Geschäftsprozesses [Sch96]. Auf eine Diskussion der Möglichkeiten der Zuordnung von Kosten zu Teilprozessen wird hier verzichtet. Es sei an dieser Stelle auf entsprechende Literatur zur Kostenrechnung verwiesen (vgl. beispielsweise [BH96] oder [Fre86]). In dieser Arbeit werden Kosten den Aktivitäten sowie den Vor- und Nachbedingungen von Aktivitäten zugeordnet.



- *Wofür* sind die Kosten angefallen?

In der *Kostenträgerrechnung* werden die anfallenden Kosten auf Kostenträger verteilt, beispielsweise die in der Abrechnungsperiode hergestellten Produkte oder die bearbeiteten Aufträge. Daraus lassen sich Bezugsgrößen für die Analyse von Kostengrößen ableiten. In der Prozeßkostenrechnung findet sich im Zusammenhang mit *leistungsmengeninduzierten*<sup>21</sup> Prozessen häufig der Begriff des *Kostentreibers*<sup>22</sup>. Darunter werden Bezugsgrößen verstanden, durch die sich die Abhängigkeit von Kosten und Leistungsvolumen eines Prozesses messen läßt. Ein Beispiel für einen Kostentreiber (eines leistungsmengeninduzierten Teilprozesses in der Fertigung) ist die Anzahl der bearbeiteten Werkstücke. Kostentreiber der Prozeßkostenrechnung können als Bezugsgröße zur Analyse von Kostengrößen verwendet werden.

Von diesen Erläuterungen ausgehend wird in den folgenden Abschnitten untersucht, welche *Kenngrößen*, *Bezugsgrößen* und *Ziele* bei der Analyse von Merkmalen aus dem Bereich *Kosten* betrachtet werden können.

## Kenngrößen

In Anlehnung an die Definition der Durchlaufzeit werden folgende *Kostenarten* unterschieden:

- Transportkosten

Die Transportkosten<sup>23</sup> [Bre97] bezeichnen die Kosten, die mit den Ortsveränderungen des Arbeitsgegenstandes bei der Ausführung einer Aktivität anfallen.

Falls es sich bei der Transportzeit um eine deterministische, nicht-variable Zeitgröße handelt, die im Geschäftsprozeßmodell mit einer Sollzeit angesetzt wird, können für Transportkosten im Modell ebenfalls Sollwerte angesetzt werden.

- Rüstkosten

Die Rüstkosten<sup>24</sup> [Bre97] umfassen alle Kosten, die bei der Ausführung einer Aktivität anfallen, um ein Arbeitssystem vorzubereiten und nach Ausführung der Arbeit gegebenenfalls in den ursprünglichen Zustand zurückzusetzen.

Falls die Rüstzeit als nicht-variable Zeitgröße modelliert wird, können Rüstkosten ebenfalls mit festen Sollwerten belegt werden.

- Bearbeitungskosten

Die *Bearbeitungskosten*<sup>25</sup> [Bre97] bezeichnen die Kosten, die bei der Ausführung einer Aktivität für die Bearbeitung des jeweiligen Gegenstandes der Arbeit anfallen.

Auch hier können für den Fall nicht-variabler Bearbeitungszeit feste Sollwerte angesetzt werden.

---

<sup>21</sup>Bei *leistungsmengeninduzierten* Prozessen variieren die Kosten in direkter Abhängigkeit von der erstellten Leistung [Fre86].

<sup>22</sup>Auch als *cost driver* [Sch96] bezeichnet.

<sup>23</sup>Auch als *Transferkosten* (vgl. [Bau97]) bezeichnet.

<sup>24</sup>Auch als *Vorbereitungskosten* [Bau97] bezeichnet.

<sup>25</sup>Auch als *Ausführungskosten* [Bau97] oder *Fertigungskosten* [Reu95] bezeichnet.

- fertigungsbedingte Wartekosten

Die *fertigungsbedingten Wartekosten* bezeichnen die Kosten, die durch planmäßiges Warten auf das Ende eines Arbeitsschritts bei der Ausführung einer Aktivität entstehen.

Hier können für den Fall nicht-variabler fertigungsbedingter Wartezeit ebenfalls feste Sollwerte verwendet werden.

- echte Wartekosten

Die *echten Wartekosten* entstehen durch nicht-fertigungsbedingtes Warten zwischen der Ausführung zweier Aktivitäten.

Da die echte Wartezeit mit der Ausführung eines Geschäftsprozesses variieren kann, läßt sie sich bei der Modellierung nicht mit Sollwerten belegen. Daher kann bei echten Wartekosten lediglich für den fixen Kostenanteil ein Sollwert angesetzt, sowie ein bestimmter (beispielsweise *linearer*) Zusammenhang zwischen variablem Kostenanteil und echter Wartezeit unterstellt werden. Ein konkreter Wert für die gesamten echten Wartekosten kann jedoch nur mittels Ausführung des Geschäftsprozeßmodells im Rahmen einer Simulation ermittelt werden.

- Vorbereitungskosten

In Analogie zur Vorbereitungszeit werden für Dienstleistungs-Aktivitäten statt Kosten für Transport- und Rüsten vereinfachend *Vorbereitungskosten* betrachtet.

Die echten Wartekosten werden dabei den Vor- und Nachbedingungen der Aktivitäten zugeordnet, alle anderen Kostenarten sind mit der Ausführung von Aktivitäten verbunden und diesen entsprechend zugeordnet.

Dabei kann für jede Kostenart zwischen einem *fixen* und einem *variablen* Anteil unterschieden werden. Der *fixe* Anteil entsteht bei jeder Ausführung einer Aktivität beziehungsweise jedem Auftreten von echter Wartezeit, der Wert des *variablen* Anteils hängt von den entsprechenden Zeitgrößen ab. Damit lassen sich Kostengrößen als Summe dieser beiden Anteile als Funktion der Zeit auffassen.

Der fixe Kostenanteil wird bei der Modellierung mit Sollwerten<sup>26</sup> belegt. Wird zudem der Zusammenhang zwischen variablem Kostenanteil und Zeit festgelegt, so kann für den Fall, daß die Zeit ebenfalls mit Sollwerten angesetzt wurde, der Sollwert für den variablen Kostenanteil (und somit der Sollwert für die gesamte Kostengröße) ermittelt werden.

## Bezugsgrößen

Bezugsgrößen bei der *Modellierung* der Kostengrößen sind wie bei der Modellierung der Zeitgrößen die Aktivitäten eines Geschäftsprozesses sowie die Vor- und Nachbedingungen dieser Aktivitäten. Da die Kostengrößen Funktionen der Zeit sind, lassen sich die Bezugsgrößen für die *Analyse* von Geschäftsprozeß-Entwürfen aus Abschnitt 2.2.3 von Zeit auf Kosten übertragen. Entsprechend können Kostengrößen auf

---

<sup>26</sup>Zur Ermittlung und Zuordnung dieser Sollwerte in der Praxis vgl. beispielsweise [Sch96]. Dort wird die Bewertung von Geschäftsprozessen mittels *kostenstellenorientierten Teilkostenrechnungssystemen* beschrieben.

- die Bearbeitung eines Case,
- die Ausführung einer Aktivität,
- den Einsatz einer Ressource zur Ausführung einer Aktivität oder
- die Bearbeitung eines Auftrags (beispielsweise Erzeugung einer bestimmten Anzahl an Fertigprodukten)

bezogen werden.

## Ziele

Wie bei der Zeitanalyse ist die Ermittlung aller variablen Kostengrößen, die erst im Rahmen der Simulation des Verhaltens von Geschäftsprozessen mit Werten belegt werden können (zum Beispiel echte Wartekosten), erstes Ziel einer Kostenanalyse. Von dieser Ermittlung ausgehend lassen sich auch bei der Kostenanalyse von Geschäftsprozessen beliebige Fragestellungen untersuchen. Als Beispiele seien hier

- die Ermittlung von Durchschnittskosten,
- die Ermittlung von Kostentreibern,
- die Ermittlung von Stückkosten oder die
- Optimierung von Kostengrößen

genannt.

Abschließend soll kurz erläutert werden, wie die beschriebenen Zeit- und Kostengrößen als Ausgangspunkt zur Beantwortung weiterer Fragestellungen genutzt werden können.

## Quantitative Merkmale (III): Weitergehende Analysen

Ausgehend von der Analyse der in den beiden vorangegangenen Abschnitten beschriebenen Zeit- und Kostengrößen ist es mit der Wahl der entsprechenden Bezugsgrößen möglich, eine ganze Reihe weiterer Fragestellungen zu beantworten. Aufgrund ihrer Bedeutung für Geschäftsprozesse (vgl. Abschnitt 2.1) werden zunächst *Ressourcen* einer gesonderten Betrachtung unterzogen.

### Merkmale von Ressourcen

Ressourcen werden zur Ausführung von Aktivitäten benötigt. In den vorangegangenen Abschnitten zu Zeit- und Kostengrößen sind Ressourcen bereits als Bezugsgröße genannt worden, jedoch gibt es noch weitere interessierende Merkmale beziehungsweise *Kenngrößen*:

- Wie hoch ist der *Auslastungsgrad*<sup>27</sup> [Aal98d] der Ressourcen?
- Kommt es zu einer *Nichtverfügbarkeit* [Reu95] von Ressourcen?

---

<sup>27</sup>Auch als *Ressourcenauslastung* [PRO01, Reu95] oder *Auslastung von Ressourcen* [Jae96] bezeichnet.

Die *Bezugsgrößen* bei Beantwortung dieser Fragestellungen können wie auch schon bei Zeit und Kosten beliebig gewählt werden.

Als Beispiele für *Ziele der Analyse* seien hier genannt:

- Ermittlung der durchschnittlichen Auslastung einer Ressource,
- Ermittlung von Engpässen,
- Maximierung der Auslastung einer Ressource.

### **Sonstige quantitative Merkmale, abgeleitete Kenngrößen**

Die bisher eingeführten Kenngrößen lassen sich beliebig zu neuen Kenngrößen kombinieren. An dieser Stelle sollen nur einige wenige Kombinationsmöglichkeiten erwähnt werden:

- Durchsatz des Systems [KL96]:
  - Produkt-Output zu Mitarbeiterzahl,
  - Produkt-Output zu Ressourceneinsatz,
  - maximale Kapazitätsauslastung des Systems;
- Minstdauer<sup>28</sup> [Bre97, Obe96],
- Prozentsatz pünktlicher Lieferungen [KL96].

#### **2.2.4 Anforderungen an Formalismen und Methoden für den Entwurf**

Ausgehend von der Betrachtung der verschiedenen Merkmale, die für die Beurteilung der Qualität eines Geschäftsprozeß-Entwurfs relevant sind, wird in diesem Abschnitt auf den Zusammenhang zwischen Zielen bei der Analyse von Geschäftsprozeß-Entwürfen und den sich daraus ergebenden Anforderungen an die verwendeten Modelle und Werkzeuge eingegangen<sup>29</sup>.

Einige Fragestellungen bei der *Validierung* von Geschäftsprozessen lassen sich ausschließlich durch Betrachtung des Geschäftsprozesses durch die am Reengineering-Prozeß Beteiligten beantworten (zum Beispiel Berücksichtigung der Kundenbedürfnisse), bei anderen (beispielsweise bei der Ermittlung von Medienbrüchen) ist der Einsatz von (rechnergestützten) Analyseverfahren dagegen durchaus sinnvoll. Im ersten Fall reichen im Prinzip Papier und Bleistift oder ein herkömmliches Graphikprogramm als Werkzeuge aus, ohne daß eine vorherige Vereinbarung einer Syntax und einer Semantik bei der Modellierung nötig ist. Der Einsatz von Analyseverfahren im zweiten Fall setzt solche Vereinbarungen den Anforderungen der jeweiligen Verfahren entsprechend voraus. Für die Beantwortung von Fragestellungen, die im Zusammenhang mit dem Verhalten des Modells stehen (beispielsweise die Frage nach der Häufigkeit ausgeführter Aktivitäten), ist die Ausführbarkeit

---

<sup>28</sup>Auch als *kritischer Pfad* bezeichnet (vgl. Abschnitt 7.3.2).

<sup>29</sup>Umfangreiche Anforderungskataloge für Modellierungformalismen finden sich beispielsweise in [Obe96] und [Zel96].

des Modells im Rahmen einer Simulation erforderlich. Dies trifft in besonderem Maße auf die *Visualisierung* des Verhaltens zu. Hier ist es nicht nur notwendig, daß das Verhalten im Rahmen einer Simulation generiert werden kann, es ist darüber sehr hilfreich, wenn über die Möglichkeit einer *Animation* der Simulation hinaus das generierte Verhalten nach Abschluß der Simulation ebenfalls in graphischer Form repräsentiert werden kann. Auf diese Weise wird bei großen Modellen, vor allem solchen mit Zyklen, die Interpretation der Simulationsergebnisse in Bezug auf den Geschäftsprozeß-Entwurf erheblich erleichtert.

Für die *Verifikation* eines Geschäftsprozeßmodells sind eine formale Syntax und eine formale Semantik, die den Anforderungen des Verfahrens zur Verifikation entsprechen, zwingende Voraussetzung. Die weiter unten erläuterte Problematik „Anwenderfreundlichkeit vs. formale Semantik“ ist an dieser Stelle von entscheidender Bedeutung. Bei der Gestaltung vieler Modelle und Werkzeuge ist das Hauptaugenmerk der Entwickler auf eine möglichst einfache, *intuitive* Modellierbarkeit der Geschäftsprozesse durch den Anwender im Reengineering-Prozeß gerichtet, wodurch ein Einsatz der erstellten Modelle zur Verifikation erschwert (oder unmöglich gemacht) wird.

Bei der *Leistungsbewertung* wird anhand des Geschäftsprozeßmodells die Leistungsfähigkeit des Geschäftsprozesses analysiert.

Da bei der Analyse quantitativer Merkmale im Rahmen einer *Leistungsbewertung* ebenfalls das Verhalten des Geschäftsprozesses Gegenstand der Analyse ist, werden auch in diesem Bereich Möglichkeiten zur Simulation benötigt (s.o.).

Auch wenn ihre Rolle mittlerweile nicht mehr ganz unumstritten ist [LPG98], kommt den *Werkzeugen* zur Modellierung und Analyse von Geschäftsprozessen weiterhin große Bedeutung bei der Gestaltung von Geschäftsprozessen zu. Zwar darf man nicht übersehen, daß es sich beim Entwurf von Geschäftsprozessen in erster Linie um einen kreativen Prozeß handelt, dessen Erfolg entscheidend davon abhängt, ob die in der Praxis für die Durchführung eines Geschäftsprozesses Verantwortlichen die Gestaltung des Geschäftsprozesses mittragen. Spätestens bei der Schaffung einer Diskussionsgrundlage in Form einer (graphischen) Modellierung des interessierenden Geschäftsprozesses wird jedoch ein entsprechendes Werkzeug zur Unterstützung des kreativen Prozesses benötigt werden. Eine Abschätzung vor allem quantitativ bewertbarer Folgen von Entwurfs-Entscheidungen durch Simulation des Verhaltens der entwickelten Entwurfs-Alternativen ist bei der Entscheidung über die letztendliche Gestaltung eines Geschäftsprozesses ein wertvolles Hilfsmittel.

Die *Anwenderfreundlichkeit* eines Werkzeugs für den Geschäftsprozeß-Entwurf wird als Grundvoraussetzung für eine tatsächliche Anwendung in der Praxis angesehen. Falls das entsprechende Modell eines Geschäftsprozesses für die in der Praxis Verantwortlichen nicht intuitiv verständlich ist, wird es kaum als Hilfsmittel für die tägliche Arbeit akzeptiert werden. Diese Einsicht hat dazu geführt, daß beispielsweise für die *ereignisgesteuerten Prozeßketten (EPK)*, die sich mittlerweile als Quasi-Standard für Referenzmodelle von Geschäftsprozessen im Bereich von SAP-(R3)-Implementierungen durchgesetzt haben, „intuitiv verständliche“ Modellierungsprimitive entwickelt wurden, um Geschäftsprozesse einfach und dennoch relativ detailgetreu in Modellen abbilden zu können [Sch95].

Leider hat die Betonung der Anwenderfreundlichkeit oft zur Folge, daß eine *formale Semantik* des mit einem Werkzeug erstellten Modells verlorengeht. Dadurch wird die Aussagekraft des Modells verwässert, der modellierte Geschäftsprozeß ist nicht mehr eindeutig zu interpretieren. Soll das Modell lediglich zu Dokumentations- oder Diskussionszwecken verwendet werden, stellt die Interpretierbarkeit modellierter Geschäftsprozesse kein Problem dar. Anders sieht es für den Fall aus, daß das Modell des Geschäftsprozesses im Rahmen einer Simulation *ausführbar* sein soll (beispielsweise zur *Leistungsbewertung* modellierter Prozeßalternativen).

Ausgehend von diesen Betrachtungen lassen sich zusammenfassend die folgenden Anforderungen formulieren, denen ein Modellierungsformalismus sowie unterstützende Werkzeuge genügen müssen, um für den Entwurf von Geschäftsprozessen unter Berücksichtigung der relevanten Qualitätsaspekte geeignet zu sein:

- Modellierung
  - Das Geschäftsprozeßmodell kann graphisch erzeugt werden.
  - Alle relevanten Merkmale, insbesondere quantitativer Art, können dargestellt werden.
  - Verschiedene Typen von Geschäftsprozessen können modelliert werden.
  - Die zur Modellierung verwendeten Bausteine verfügen über eine formale Semantik, die eine eindeutige Interpretation des Geschäftsprozeßmodells durch einen Rechner erlauben.
  - Das Geschäftsprozeßmodell beinhaltet dynamische Aspekte, die eine Ausführung im Rahmen von Simulation erlauben.
- Validierung
  - Das Verhalten des Geschäftsprozeßmodells kann generiert werden.
  - Während der Simulation kann das Verhalten als Animation visualisiert werden.
  - Das Verhalten des Geschäftsprozeßmodells kann auch nach Durchführung der Simulation visualisiert werden.
- Verifikation
  - Business Rules können formuliert, deren Erfüllung oder nicht-Erfüllung untersucht beziehungsweise garantiert werden.
  - Das Geschäftsprozeßmodell ist bezüglich grundlegender Eigenschaften überprüfbar, beispielsweise bezüglich *Soundness*.
- Leistungsbewertung
  - Alle relevanten Kenngrößen aus den Bereichen *Zeit*, *Kosten* und *Ressourcen* können ermittelt werden.
  - Beim Entwurf verwendete Schätz- und Sollwerte können systematisch getestet werden.
  - Unterschiedliche Szenarien (beispielsweise unterschiedliche Ausprägungen für Fälle, unterschiedliche Auslastungen) können überprüft werden.

- Aufbauend auf den ermittelten Ergebnissen wird gegebenenfalls eine (optimale) Konfiguration des Geschäftsprozeßmodells unterstützt.

In Abschnitt 3.2.1 dienen diese Anforderungen als Grundlage zur Formulierung eines Anforderungsprofils für Werkzeuge. Zuvor werden noch einige bekannte Ansätze für den Entwurf von Geschäftsprozessen vorgestellt und im Hinblick auf die formulierten Anforderungen diskutiert.

## 2.3 Ansätze für den Entwurf von Geschäftsprozessen

Im vorangegangenen Abschnitt sind eine Reihe wichtiger Anforderungen an Formalismen und Methoden für den Entwurf von Geschäftsprozessen formuliert worden, denen insbesondere die verwendeten Modelle und Werkzeuge gerecht werden sollten. In den folgenden Abschnitten werden daher eine Reihe bekannter Ansätze für den Entwurf von Geschäftsprozessen bezüglich dieser Anforderungen untersucht.

Dazu werden in Abschnitt 2.3.1 zunächst einige allgemeinere Ansätze, vor allem aus dem Bereich des *Business Process (Re-)Engineering* vorgestellt. Diese meist der Betriebswirtschaftslehre nahen Ansätze behandeln Geschäftsprozesse auf einem sehr hohen Abstraktions-Niveau und verzichten mitunter auf eine explizite Darstellung der Prozesse mittels Modellierungs-Formalismen. Im Hinblick auf die im Rahmen dieser Arbeit formulierten Anforderungen an Entwurfsmethoden- und -werkzeuge sind diese Ansätze damit eher uninteressant. Sie stellen jedoch den Ausgangspunkt für die Entwicklung modellbasierter Ansätze dar und enthalten zudem hilfreiche Empfehlungen für das *Vorgehen* beim Entwurf von Geschäftsprozessen.

Da im Kontext der Verknüpfung von Geschäftsprozessen und Informationssystemen die explizite Darstellung von Geschäftsprozessen unabdingbar ist (vgl. Abschnitt 1.1), werden in den Abschnitten 2.3.2 und 2.3.3 eine Reihe *modellbasierter* Ansätze vorgestellt. Ein Schwerpunkt wird dabei auf *petrinetzbasierte* Ansätze gelegt, denen aufgrund ihrer besonderen Eignung im Hinblick auf die formulierten Anforderungen ein eigener Abschnitt (2.3.3) gewidmet ist.

Ein kurzes Fazit in Abschnitt 2.3.4 schließt die Betrachtung bekannter Ansätze ab.

### 2.3.1 Allgemeinere Arbeiten

Seit Beginn der 90er Jahre ist mit den Arbeiten von Hammer und Champy [HC93] sowie Davenport [Dav93] die Diskussion um einen prozeßorientierten Fokus bei der Gestaltung der Organisation von Unternehmen neu eröffnet worden [Bec99]. Auch wenn die Grundlagen dieser ablauforientierten Sichtweise nicht neu sind und spätestens seit den Arbeiten von Gaitanides [Gai83] unter dem Begriff der *Prozeßorganisation* als etabliert gelten, haben prozeßorientierte Konzepte durch die Veröffentlichung dieser Arbeiten einen entscheidenden Schub erhalten [Bec99, Uth01].

Neben der Etablierung von Geschäftsprozessen als Paradigma zur Organisation von Unternehmen liegt der Schwerpunkt dieser Arbeiten vor allem auf dem *Vorgehen* bei der

(Neu-)Gestaltung von Geschäftsprozessen, beispielsweise im Rahmen des *Business Process Engineering* oder verwandter Ansätze (vgl. [HC93, Dav93, JMPW93, MK96, MB93, PH96]). Zwar werden beispielsweise im Rahmen der *Entwurfsempfehlungen für Geschäftsprozesse* (vgl. [HC93] und Abschnitt 2.2.1) einige konkrete Hinweise für die Gestaltung von Geschäftsprozessen gegeben, diese sind jedoch eher allgemein gehalten. Eine Unterstützung beim Entwurf hinsichtlich der Aspekte *Konsistenz* und *Effizienz* fehlt in den meisten Fällen komplett. Hinsichtlich des in dieser Arbeit zugrunde gelegten Qualitätsbegriffs und der daraus resultierenden Anforderungen erweisen sich diese Ansätze daher für den Entwurf von Geschäftsprozessen im Kontext von Informationssystemen als nicht besonders hilfreich.

Auch wenn, beispielsweise bei Davenport [Dav93], die Rolle der Informationstechnologie für den Entwurf und die Unterstützung von Geschäftsprozessen bereits Erwähnung findet, fehlt es an konkreten Methoden und Formalismen, die eine praktische Umsetzung der formulierten *Vorgehensmodelle* ermöglichen würde. Sofern die explizite Darstellung von Geschäftsprozessen überhaupt berücksichtigt wird, handelt es sich bei den verwendeten Beschreibungsmethoden im Prinzip um „Bilder“. Diese eignen sich dank ihrer Einfachheit zwar als Diskussionsgrundlage in sehr frühen Phasen des Entwurfs, bieten aber aufgrund fehlender formaler Semantik keinerlei Möglichkeiten zur Simulation und den damit verbundenen Analysen (vgl. Abschnitt 2.2.4).

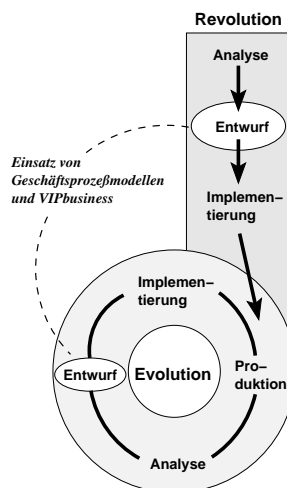


Abbildung 2.3: Revolution und Evolution beim Business Process Reengineering [Öst95b].

Hinsichtlich der Erfüllung der in Abschnitt 2.2.4 formulierten Anforderungen sind im Prinzip nur Ansätze von Interesse, die ein Vorgehensmodell mit konkreten Vorschlägen zu dessen Umsetzung durch Bereitstellung der geeigneten Methoden und Modelle versehen. An dieser Stelle seien exemplarisch die Arbeiten von Österle genannt [Öst95a, Öst96], in denen ausgehend von einem Vorgehensmodell (siehe Abbildung 2.3) eine umfangreiche Sammlung an Methoden und Modellen entwickelt wird, um die Gestaltung von Geschäftsprozessen im Kontext der unterstützenden Informationstechnologie zu ermöglichen [Öst95b, HBs95a].



Obwohl in den allgemeineren Arbeiten zum Entwurf von Geschäftsprozessen konkrete Methoden und Modelle fehlen, wird mit Hilfe der entwickelten Vorgehensmodelle ein *Rahmen* geschaffen, der die Anwendung verschiedener, auf spezielle Bedürfnisse zugeschnittene, Methoden erlaubt. In diesem Sinne stellt der in den nächsten Kapiteln entwickelte *VIPbusiness-Ansatz* Methoden zur Modellierung und Analyse von Geschäftsprozeß-Entwürfen zur Verfügung, die in Kombination mit etablierten Vorgehensmodellen, beispielsweise *Lebenszyklus-Modellen* (vgl. [Hei96] oder [Dei00] sowie Abbildung 2.4), den Entwurf qualitativ hochwertiger Geschäftsprozesse ermöglichen. Die Phasen oder Schritte des jeweiligen Vorgehensmodells, die der *VIPbusiness-Ansatz* nicht unterstützt, werden durch Hinzunahme weiterer Methoden, beispielsweise Interviewtechniken bei der Erhebung der Ist-Situation, bewältigt.

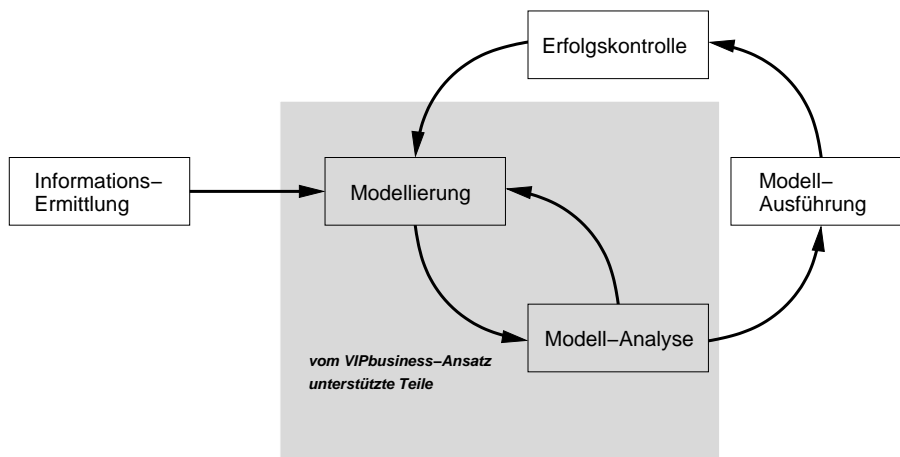


Abbildung 2.4: Lebenszyklus eines Geschäftsprozesses [Dei00].

Zusammenfassend läßt sich feststellen, daß durch die allgemeineren Arbeiten zum Thema Geschäftsprozesse lediglich einzelne Qualitäts-Aspekte abgedeckt werden. Aufgrund der informellen Natur der verwendeten Formalismen sind keine eindeutigen Interpretationen der erstellten Geschäftsprozeßmodelle und somit keine weitergehenden Analysen möglich. Das Fehlen solcher Analysemöglichkeiten stellt die Ursache für das Scheitern vieler Projekte vor allem im BPR-Bereich dar [EKO96]. Die Rolle deskriptiver Modelle von Geschäftsprozessen als wichtige Schnittstelle beim Entwurf betrieblicher Informationssysteme wurde bereits in Abschnitt 1.1 diskutiert. Da die allgemeineren Arbeiten hier keine oder nur unzureichende Unterstützung bieten, wenden wir uns in den folgenden Abschnitten Ansätzen zu, bei denen eine graphische Modellierung der betrachteten Geschäftsprozesse im Mittelpunkt steht.

### 2.3.2 Modellbasierte Ansätze

Seit Beginn der 90er Jahre ist eine Vielzahl an Methoden und Modellen zur Beschreibung von Geschäftsprozessen vorgestellt worden. Für unsere Zwecke sind dabei vor allem solche Ansätze von Interesse, die eine *Methode*, ein *Vorgehensmodell* oder eine *Architektur* mit graphischen Notationen zur Modellierung verbinden. Im Folgenden werden einige wichtige

Ansätze aus der Literatur vorgestellt und bezüglich der in Abschnitt 2.2.4 formulierten Anforderungen diskutiert<sup>30</sup>. Methoden aus dem Kontext von *Werkzeugen* für den Entwurf von Geschäftsprozessen werden im Rahmen der Untersuchung kommerzieller Werkzeuge in Abschnitt 3.2.2 behandelt.

Kernstück eines Geschäftsprozesses sind die Aktivitäten, die zur Bearbeitung von Cases auszuführen sind (vgl. Abschnitt 2.1). Bei der Ausführung der Aktivitäten sind verschiedene *Aspekte*<sup>31</sup> zu berücksichtigen (vgl. auch [DIN96]):

- Aktivitäten sind (teilweise) in einer bestimmten *Reihenfolge* auszuführen.
- Zur Ausführung der Aktivitäten werden *Ressourcen* (zum Beispiel Mitarbeiter oder Maschinen) benötigt.
- Die Ausführung wird von einem *Informationsfluß* begleitet: Daten entstehen, werden ausgetauscht, weiterverarbeitet und vernichtet.
- In der Regel sind verschiedene *organisatorische Bereiche* eines Unternehmens (zum Beispiel Einkauf und Fertigung) betroffen.

Die Integration all dieser Aspekte in einem Modell führt in der Regel zu komplexen und unübersichtlichen Modellen. Daher werden üblicherweise mehrere, den einzelnen Aspekten entsprechende Sichten gebildet, die in verschiedenen (Teil-)Modellen dargestellt werden. Auf diese Weise ist es beispielsweise möglich, funktionale und organisatorische Aspekte unabhängig voneinander zu untersuchen. Die Art sowie die Anzahl der Sichten auf den Geschäftsprozeß variiert von Modell zu Modell.

### Das Semantische Objektmodell (SOM)

Ziel des zum *Semantischen Objektmodell (SOM)* von Ferstl und Sinz [FS93, FS98, FS95, FS96] gehörigen *Rahmenkonzepts* (vgl. Abbildung 2.5), das sich aus der *Unternehmens-Architektur* und dem *V(orgehens)-Modell* zusammensetzt, ist eine durchgängige Unterstützung der Umsetzung von Unternehmensplänen durch IT-gestützte Geschäftsprozesse. Die drei Teile der Unternehmens-Architektur (*U(nternehmens)plan*, *Geschäftsprozesse* und *Anwendungssysteme*) stellen unterschiedliche *Modell-Ebenen* dar und werden auf Seite des V-Modells mittels verschiedener Sichten und zugehöriger Modelle beschrieben.

Für Geschäftsprozesse wird eine *strukturorientierte* und eine *verhaltensorientierte* Sicht vorgeschlagen, umgesetzt durch ein *Interaktionsmodell* und ein *Aufgabensystem* (vgl. Abbildung 2.5).

Die *strukturorientierte Sicht* setzt sich aus der *Leistungssicht* und der *Lenkungssicht* zusammen und beschreibt die *betrieblichen Objekte* (zum Beispiel der Einkauf eines Unternehmens) und die *betrieblichen Transaktionen* (beispielsweise die Zahlung einer Rechnung) sowie die Interaktion zwischen Objekten und Transaktionen. Die Leistungssicht beschreibt die *Leistung*<sup>32</sup>, die bei der Durchführung eines Geschäftsprozesses erbracht wird.

---

<sup>30</sup>Eine umfassende Beschreibung verschiedener Ansätze findet sich beispielsweise in [HBs95b] oder [HB96].

<sup>31</sup>Auch als *Dimensionen* [Aal98d] oder *Sichten* [Sch94] bezeichnet.

<sup>32</sup>Ferstl und Sinz verwenden hier den Begriff *Beitrag zur Sachzielerfüllung der Unternehmung entlang der Wertschöpfungskette unter Beachtung von Formalzielen, Erfolgsfaktoren und Strategien* [FS96].

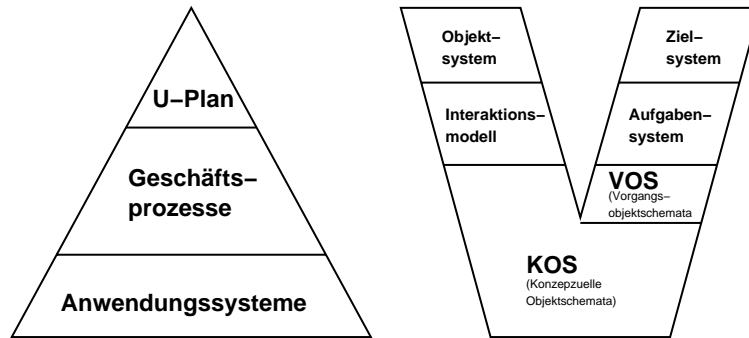


Abbildung 2.5: *SOM*-Rahmenkonzept [FS98].

Die Lenkungssicht verdeutlicht die Koordination<sup>33</sup> von betrieblichen Objekten bei der Durchführung betrieblicher Transaktionen.

Die *verhaltensorientierte Sicht* entspricht im wesentlichen der *Ablaufsicht*, durch die in *Vorgangs-Ereignis-Schemata* der Zusammenhang zwischen Aufgaben und Ereignissen dargestellt wird. Dieser Sichtweise liegt die Annahme zugrunde, daß die Durchführung eines Geschäftsprozesses einem ereignisgesteuerten Ablauf von Vorgängen entspricht.

In [FS96] wird die Modellierung von Geschäftsprozessen im Rahmen von *SOM* als *semi-formal* bezeichnet. Eine Ausführung der erstellten Modelle im Rahmen von Simulationen ist offenbar nicht vorgesehen. Eine Betrachtung quantitativer Merkmale oder eine *Verifikation* der Modelle findet nicht statt, der Schwerpunkt der Analyse liegt eindeutig auf Merkmalen aus dem Bereich der *Validierung* (vgl. Abschnitt 2.2.1), beispielsweise die Untersuchung von *Medienbrüchen* oder der prozeßbegleitenden Kommunikation und Koordination [FS96].

### Architektur integrierter Informationssysteme (ARIS)

Bei der *Architektur integrierter Informationssysteme (ARIS)* [KNS92, Sch95] handelt es sich ebenfalls um einen Ansatz, der die Integration von Geschäftsprozessen und unterstützenden Informationssystemen in den Vordergrund stellt. Dabei werden die drei Sichten

- *Funktionssicht*,
- *Datensicht* und
- *Organisationssicht*

über eine *Steuerungssicht* integriert (vgl. Abbildung 2.6).

Innerhalb dieser Sichten kommen je nach Fokus auf betriebswirtschaftliche oder informationstechnische Problemstellungen Beschreibungsmittel dreier Ebenen, dem *Fachkonzept*, dem *DV-Konzept* und der *Implementierung*, zum Einsatz [Rum99].

Die Modellierung von Geschäftsprozessen erfolgt auf der Ebene des Fachkonzepts der

<sup>33</sup>Als Formen der Koordination werden dabei das *Verhandlungs-* und das *Regelungsprinzip* aufgeführt.

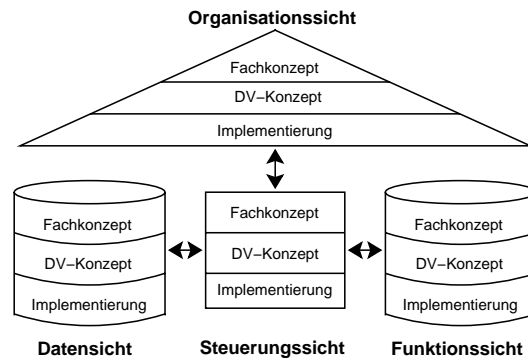


Abbildung 2.6: ARIS-Rahmenkonzept [Sch95].

Steuerungssicht. Als Beschreibungssprache werden dabei *ereignisgesteuerte Prozeßketten* (EPK) verwendet. Diese sind nicht zuletzt aufgrund ihrer Funktion als Referenzmodelle im Rahmen von SAP-Implementierungen in der Praxis weit verbreitet. Mit dem gleichnamigen Werkzeug ARIS der IDS Scheer AG [IDS00] steht ein bekanntes kommerzielles Werkzeug zur Verfügung. Dieses wird in Abschnitt 3.2.2 ausführlich untersucht.

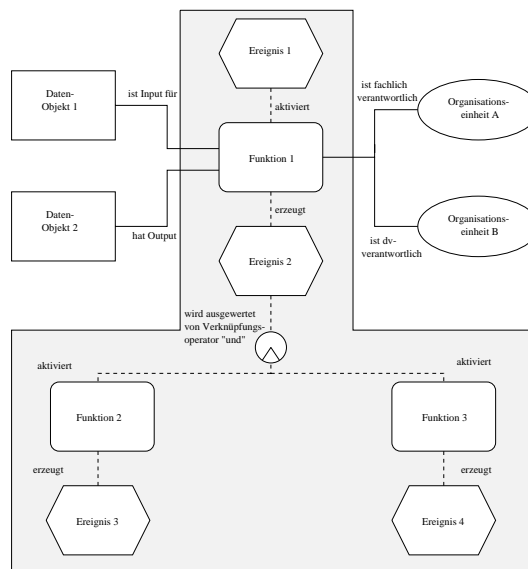


Abbildung 2.7: EPKs als Modell der Steuerungssicht [SJ96].

Aufgrund ihrer Bedeutung für die Praxis ist zum Thema EPKs eine Vielzahl an Beiträgen erschienen. Eine umfangreiche Sammlung von Literaturquellen zu den Themen ARIS und EPKs findet sich beispielsweise in [Nüt97]. Im Zusammenhang mit den Möglichkeiten zur Analyse von EPKs sind dabei eine Reihe von Arbeiten besonders interessant, die sich mit dem Problem einer formalen Semantik für EPKs beschäftigen [Aal98b, LSW97, ZR96, Rum97, Rum99]. Mit Einschränkungen können auf diese Weise sogar einige Fragestellungen aus dem Bereich *Verifikation* betrachtet werden [Aal98b, Rum99].

Bezüglich der in Abschnitt 2.2.4 formulierten Anforderungen weisen EPKs die folgenden Schwachstellen auf (vgl. auch Abschnitt 3.2.2, dort wird das Werkzeug *ARIS* eingehend untersucht):

- Die Modellierung nicht-strukturierter Teile von Geschäftsprozessen wird nicht unterstützt.
- Das Verhalten des Geschäftsprozeßmodells kann nach Durchführung der Simulation nicht mehr visualisiert werden.
- Da Simulationen für jeden neu verwendeten Schätz- oder Sollwert wiederholt werden müssen, kommt es zu Problemen bei der Vergleichbarkeit der Ergebnisse (vgl. auch Abschnitt 3.1.2). Insbesondere bei großen Geschäftsprozeßmodellen zeigt sich zudem die Ineffizienz dieses Simulationskonzepts.

## **Workflow Management und Operations Research**

Hauptanliegen des *Workflow Management* [JB96, JBS97, KM95, Kou95, Law97] ist die Unterstützung von Geschäftsprozessen (oder *Workflows*) durch *Workflow Management Systeme (WfMS)*. Das Spektrum der Aufgaben eines *WfMS* reicht dabei von der Unterstützung des Anwenders bei der Ausführung einzelner Aktivitäten über die vollautomatische Ausführung von Aktivitäten bis hin zu Aspekten der Datenhaltung und des Prozeß-Monitoring. Diese unterschiedlichen Aufgaben spiegeln sich in der WfMS-Referenzarchitektur der *Workflow Management Coalition (WfMC)* wider (vgl. Abbildung 2.8 sowie [WFM94, WFM96]), die entworfen wurde, um die Produkte verschiedener Anbieter für die verschiedenen Aufgabenbereiche über festgelegte Schnittstellen kombinieren zu können. Im Hinblick auf Unterstützung beim Entwurf von Geschäftsprozessen sind vor allem die *Process Definition Tools* interessant. Dabei handelt es sich um Werkzeuge (oft eingebettet in Methoden) für den Entwurf von Prozeßmodellen, die als Steuerung von *Workflow Engines* zur Laufzeit eines *WfMS* dienen.

Auch in diesem Gebiet existiert eine Vielzahl an Ansätzen (vgl. beispielsweise [Rol96, Göt95, JBS97, MS99, GHS95]), die jedoch vor allem auf Aspekte der *Ausführung* der Geschäftsprozesse durch das *WfMS* und die damit verbundenen technischen Fragestellungen und Probleme zugeschnitten sind. Aus diesem Grund erfolgt insbesondere die Modellierung der betroffenen Geschäftsprozesse auf einem sehr niedrigen Abstraktionsgrad, der den Einsatz der entsprechenden Methoden und Modelle für die konzeptuelle Arbeit in frühen Phasen des Geschäftsprozeß-Entwurfs als nicht ratsam erscheinen läßt [Rum99].

Für unsere Zwecke interessant sind in diesem Zusammenhang Arbeiten, die sich mit dem Problem der Modellierung nicht vollständig strukturierter Geschäftsprozesse und der Unterstützung dieser Prozesse durch *WfMS* auseinandersetzen (vgl. dazu beispielsweise [HHJ<sup>+</sup>98]). Diese Arbeiten bilden den Ausgangspunkt für die Entwicklung eines Konzepts zur Unterstützung des Entwurfs nicht vollständig strukturierter Geschäftsprozesse im Rahmen des *VIPbusiness*-Ansatzes (vgl. [DE00] sowie Abschnitt 5.6).

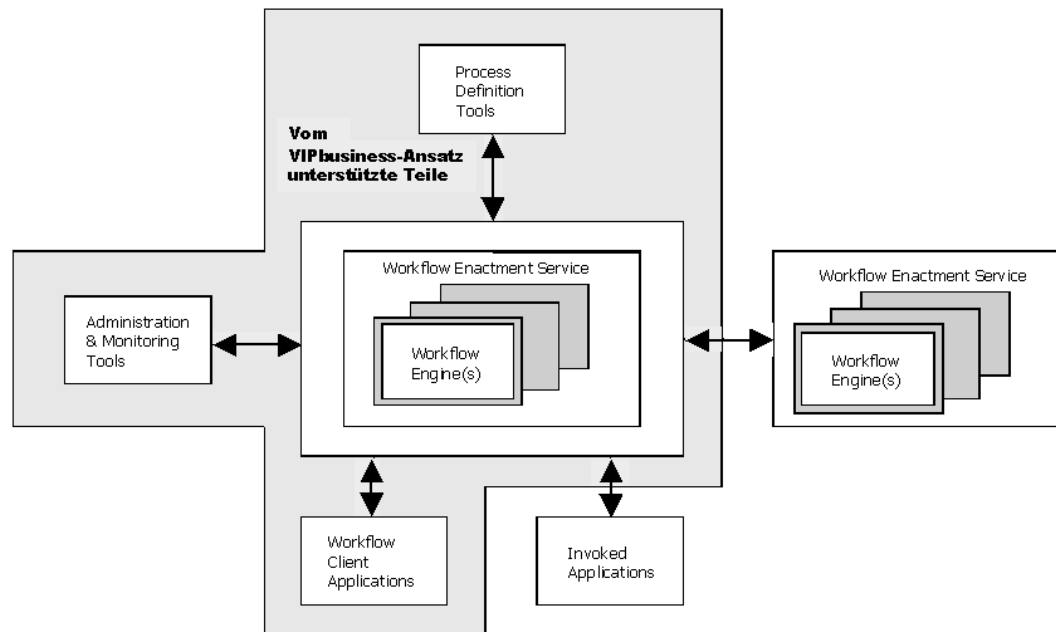


Abbildung 2.8: Referenz-Architektur der WfMC [Law97] und ihre Unterstützung durch *VIPbusiness*.

Im Bereich des Workflow Management herrscht ein besonderer Bedarf an Modellen und Werkzeugen, die sich sowohl für frühe, anwendungsorientierte Phasen des Entwurfs als auch für den Übergang zu systemorientierten Phasen mit den bereits erwähnten technischen Fragestellungen eignen. Ausgehend von diesem Problem sind beispielsweise in [Obe96] Methoden und Modelle entwickelt worden, die den Anspruch haben, einen *durchgängigen* Entwurf von Geschäftsprozessen im Kontext von Informationssystemen zu ermöglichen. Diese Ansätze basieren auf *Petrinetzen*.

Bevor wir uns daher in Abschnitt 2.3.3 den *petrinetzbasierten Modellen* zuwenden, seien noch eine Reihe von Methoden und Modellen aus dem Bereich des *Operations Research (OR)* (vgl. beispielsweise [NM93]) wie *Markovketten*, *Warteschlangenmodelle* sowie verschiedene Formen der *Netzplantechnik* genannt [LK66, NM93, NS79, Tij94]. Diese eignen sich zur Untersuchung spezieller quantitativer Problemstellungen, die zur Beurteilung der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs äußerst hilfreich sind. Die entsprechenden Techniken werden seit langem im Bereich der Fertigung und Produktionsplanung angewendet, im Kontext des Entwurfs von Geschäftsprozessen werden sie jedoch nur einzeln diskutiert (vgl. beispielsweise [Sch97]). Eine der Ausnahmen stellt [Buz96] dar, dort werden Geschäftsprozesse mit Hilfe quantitativer Methoden untersucht, die auf Heuristiken und Konzepten der Warteschlangentheorie basieren. Ein weiterer (petrinetzbasierter) Ansatz wird im nächsten Abschnitt diskutiert [vHR00], in Abschnitt 7.4.2 werden Techniken aus dem Bereich des OR zur Analyse von Geschäftsprozeß-Entwürfen im Rahmen des *VIPbusiness*-Ansatzes genutzt.

### 2.3.3 Petrinetzbasierte Ansätze

Petrinetze<sup>34</sup> haben sich in verschiedenen Anwendungsbereichen, etwa bei der Entwicklung von Informationssystemen (vgl. beispielsweise [SR93]) oder im Bereich der rechnergestützten Fertigung (vgl. beispielsweise [DA94]), als Hilfsmittel beim Entwurf, der Analyse und der Optimierung von Systemen durchsetzen können. Ausgehend von den ersten Arbeiten Carl Adam Petris [Pet62] sind in der Zwischenzeit eine Vielzahl weiterführender theoretischer und praktischer Ansätze mit zum Teil sehr unterschiedlichen Varianten von Petrinetzen entstanden (vgl. etwa [Bau96] oder [Rei86]). Der Einsatz von Petrinetzen ist mit einer Reihe von Vorteilen verbunden (vgl. beispielsweise [Aal98a, DO96, Obe96]):

- Bei Petrinetzen handelt es sich um eine graphische Beschreibungssprache, verknüpft mit einer formalen Semantik. Während die graphische Modellierung den Entwurf erleichtert, ermöglicht die formale Semantik die Simulation der erstellten Modelle sowie deren Analyse mittels Verfahren aus dem Bereich der *Verifikation*.
- Petrinetze eignen sich sowohl zur Erstellung erster Entwürfe als auch für detaillierte Modelle mit technischen Details. Sie unterstützen damit sowohl Top-Down- als auch Bottom-Up-Vorgehensweisen beim Entwurf.
- Petrinetze sind nicht proprietär. Sie können unabhängig von den Werkzeugen einzelner Anbieter verwendet werden. Eine Vielzahl entwickelter Analyseverfahren und etliche frei verfügbare Werkzeuge bieten umfangreiche Unterstützung bei Entwurf.

Da diese Vorteile auch für den Entwurf von Geschäftsprozessen von Bedeutung sind, haben sich Petrinetze in diesem Bereich etablieren können (vgl. beispielsweise [Aal98a, AAH98, DO96, EKR95]). Im Folgenden werden daher einige bekannte petrinetzbasierte Ansätze für den Geschäftsprozeß-Entwurf vorgestellt.

#### INCOME

Ausgehend vom *INCOME/STAR*-Projekt zum Entwurf einer Umgebung zur kooperativen Entwicklung verteilter Informationssysteme [OSS94] wurden an der Universität Karlsruhe im Rahmen des Nachfolge-Projekts *INCOME/WF* Konzepte für die Entwicklung von Workflow Management-Systemen auf Basis höherer Petrinetze erstellt [OSS<sup>+</sup>97]. Dabei werden zur Beschreibung der Geschäftsprozesse *NF<sup>2</sup>-Relationen/Transitionen-Netze* (*NR/T-Netze*) benutzt, die eine Modellierung nebenläufiger Prozesse mit komplexen Objektstrukturen als Marken ermöglichen. *NR/T-Netze* eignen sich besonders zur Modellierung der *Informationsaspekte* von Geschäftsprozessen, da die mit der Bearbeitung von Cases verbundene Verarbeitung von Informationen durch Informationssysteme, beispielsweise Workflow Management-Systeme oder Datenbanken, sehr präzise dargestellt werden kann. Ergänzt werden die *INCOME*-Konzepte durch ein *4-Schichten-Modell* für Ablaufbeschreibungssprachen, das den durchgängigen Entwurf von Geschäftsprozessen von der anwendungsorientierten bis hin zur maschinenorientierten (Programmier-)Sicht durch Verwendung jeweils geeigneter Klassen von Petrinetzen ermöglicht [EKO96, Obe96].

---

<sup>34</sup>Einen guten Überblick zu Petrinetzen geben beispielsweise [Rei86, RR98a, RR98b], eine ausführliche Vorstellung findet sich zudem in Kapitel 4.

Neuere, auf den Ergebnissen von *INCOME*/*WF* aufsetzende Arbeiten ermöglichen eine integrierte Dokumenten- und Ablaufmodellierung im Kontext des *Electronic Commerce* durch Verwendung von *SGML*-Netzen [Wei98, Wei00] oder die Entwicklung eines ereignisbasierten Workflow Management-Ansatzes [Sch00].

All diesen Ansätzen im Umfeld von *INCOME*/*WF* ist die Betonung der *Informationssicht* gemeinsam. Quantitative Aspekte wie Zeit und Kosten werden nicht betrachtet. Bezüglich des Entwurfs von Geschäftsprozessen wird vor allem die *Modellierung* unterstützt. Zwar sind die erstellten Modelle ausführbar, eine Simulation und insbesondere eine Analyse simulierten Verhaltens findet jedoch nicht statt.

Ebenfalls aus dem *INCOME*-Projekt hervorgegangen ist das gleichnamige Werkzeug für „ganzheitliches Geschäftsprozeßmanagement“ der *Promatis AG* [Jae96, PRO01]. Beim Entwurf von Geschäftsprozessen mit diesem Werkzeug werden ähnlich wie bei *ARIS* (vgl. Abschnitt 2.3.2) vier *Sichten* unterschieden, die in verschiedenen (Teil-)Modellen dargestellt werden können:

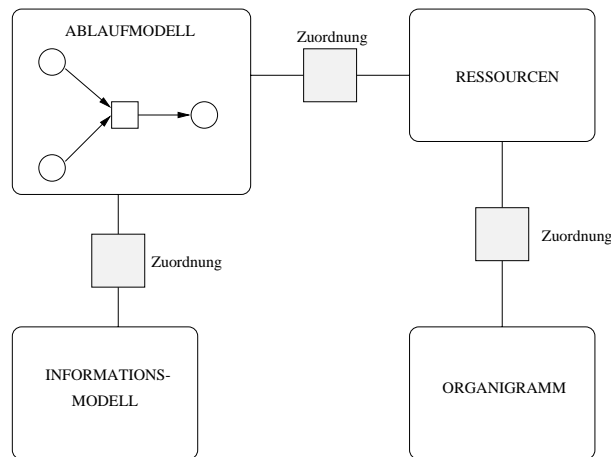


Abbildung 2.9: Integration der Teilmodelle in *INCOME* [PRO01].

- *Ablaufmodell*: Abbildung der Aktivitäten des Geschäftsprozesses,
- *Organigramm*: Beschreibung der Aufbauorganisation der Unternehmung,
- *Ressourcen*: Zuordnung zu Organisationseinheiten und Aktivitäten,
- *Informationsmodell*: Modellierung der für den Geschäftsprozeß relevanten Objekte, Beziehungen und Attribute.

Das Ablaufmodell dient dabei ähnlich wie die *EPKs* als *integrierendes* Modell (vgl. Abbildung 2.9). Das Werkzeug *INCOME* wird im Rahmen der Untersuchung kommerzieller Werkzeuge in Abschnitt 3.2.2 eingehend bezüglich der Anforderungen aus Abschnitt 2.2.4 untersucht.



## FUNSOFT

Im Rahmen des *FUNSOFT*-Ansatzes wurde eine Klasse höherer Petrinetze zunächst für die Modellierung, Analyse und Ausführung von Software-Prozessen und später von Geschäftsprozessen entwickelt [VG92, DGK95, GK96]. Die resultierenden *FUNSOFT*-Netze können zur Modellierung von Geschäftsprozessen auf verschiedenen Abstraktionsstufen verwendet werden, die Modelle sind (durch Verfeinerung) ineinander überführbar. Im Rahmen des *FUNSOFT*-Ansatzes werden 3 Schichten unterschieden [DGK95]:

- eine *Ablaufsicht* zur Darstellung von Wertschöpfungsketten unter Berücksichtigung sequentieller und paralleler Abläufe, Input-/Output-Beziehungen sowie von Daten- und Kontrollflüssen mit Entscheidungen,
- eine *Datensicht* für verschiedene Objekttypen wie Formulare oder Dokumente sowie
- eine *Organisationssicht* zur Darstellung von Prozeßverantwortlichkeiten oder Rollen.

Eingebettet sind die Entwurfsmethoden in ein Vorgehensmodell zur Unterstützung des kompletten Lebenszyklus von Geschäftsprozessen (vgl. Abbildung 2.4 sowie [Dei00]).

*FUNSOFT*-Netze kommen unter anderem in den Workflow Management-Systemen *LEU* der *LION GmbH* [Bre97] und *Melmac* [DG90] zum Einsatz, ihre Anwendung in konkreten Projekten ist jedoch offenbar mit Problemen auf Seiten der Anwender mit der Notation der Netze verbunden [DGK95]. Die Modellierung der Geschäftsprozesse wird somit nur unzureichend unterstützt [Rum99]. Eine Modellierung nicht vollständig strukturierter Geschäftsprozesse durch *FUNSOFT*-Netze ist nicht möglich.

In [BSG99] findet sich ein Ansatz zur „systematischen“ Simulation der Geschäftsprozeßmodelle im Rahmen einer Leistungsbewertung der jeweiligen Entwürfe. Da dieser Ansatz jedoch keine Trennung von Verhalten und Zeit-/Kostenkonzept beinhaltet, kommt es auch hier zu den schon für *ARIS* formulierten Problemen bezüglich Vergleichbarkeit der Ergebnisse und Effizienz des Vorgehens (vgl. Abschnitt 2.3.2 und Abschnitt 3.1.2).

## Workflow Nets

Im Umfeld der TU Eindhoven wurde mit *Workflow-Netzen*<sup>35</sup> eine Klasse von Petrinetzen zur Modellierung von Geschäftsprozessen entwickelt, bei der einfache Stellen/Transitionen-Netze um unterscheidbare Marken, Zeit und Hierarchie sowie spezielle Routing-Konstrukte erweitert wurden (vgl. [AHH94, AH95] sowie Abschnitt 7.1.1)

In [Aal98d] werden drei *Dimensionen* bei der Modellierung von Geschäftsprozessen (vgl. Abbildung 2.10) unterschieden. Die *Prozeß-Dimension* spezifiziert die Aufgaben, die zur Ausführung eines Geschäftsprozesses nötig sind, sowie die Abfolge der Aufgaben bei der Ausführung. Die *Ressourcen-Dimension* legt fest, welche Ressourcen bei der Ausführung der Aufgaben benötigt werden. Mittels der *Case-Dimension* läßt sich der Umstand beschreiben, daß jeder (konkrete) Geschäftsvorfall individuell gehandhabt wird.

---

<sup>35</sup>Im englischen Original: *workflow nets*.

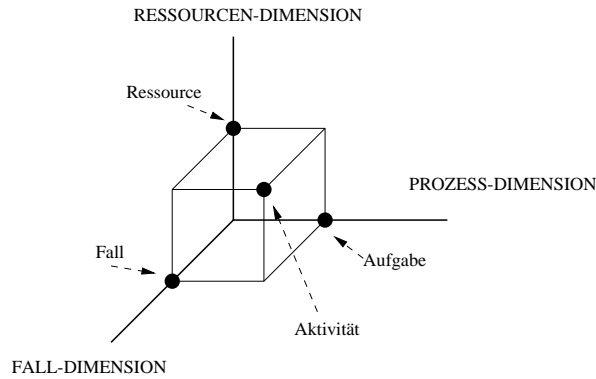


Abbildung 2.10: Dimensionen von *Workflow Nets* [Aal98d].

*Workflow-Netze* wurden zur Modellierung von Geschäftsprozessen unter Berücksichtigung aller drei Dimensionen entwickelt. Die Prozess- sowie die Ressourcen-Dimension werden dabei durch die Struktur der erstellten Netze berücksichtigt, gefärbte Marken dienen der Unterscheidbarkeit der einzelnen Cases, deren gleichzeitige Bearbeitung in einem *Workflow-Netz* dargestellt werden kann.

Für *Workflow-Netze* wurden eine Reihe von Methoden entwickelt, die für den Entwurf von Geschäftsprozessen hilfreich sind. So wird in [Aal98d, VBA01] ein Konzept sowie ein zugehöriges Werkzeug (*Woflan*) vorgestellt, mit deren Hilfe eine Analyse von Geschäftsprozess-Entwürfen bezüglich Merkmalen aus dem Bereich der *Verifikation* möglich ist. In Abschnitt 7.1.1 werden diese Konzepte zur Überprüfung der Konsistenz von Geschäftsprozess-Entwürfen im Rahmen des *VIPbusiness*-Ansatzes verwendet. Durch die Einführung von *Hierarchie*-Konzepten ist es möglich, *Workflow-Netze* zur Modellierung auf verschiedenen Abstraktionsstufen und damit in verschiedenen Phasen des Geschäftsprozess-Entwurfs zu verwenden [AB97, Aal98d, BA01]. Durch Verwendung von Konzepten aus der Warteschlangentheorie und Einführung eines Zeitkonzepts lassen sich einzelne Kenngrößen aus dem Bereich *Leistungsbewertung* wie beispielsweise die Durchlaufzeit eines Cases ermitteln [Aal95, HR98, vHR00]. Mit *ExSpect* steht auch hier ein entsprechendes Werkzeug zur Unterstützung der Konzepte zur Verfügung [AW91, Bak98].

Die verschiedenen für *Workflow-Netze* zur Verfügung stehenden Konzepte werden jedoch nicht in einem einzelnen Ansatz integriert. Vielmehr handelt es sich in den meisten Fällen um mehr oder weniger isolierte Betrachtungen einzelner Problemstellungen. Insbesondere werden bei den Analyseverfahren die zur Modellierung eingeführten Erweiterungen nicht berücksichtigt. So werden beispielsweise die in [Aal98d] vorgestellten Methoden zur Verifikation lediglich für einfache Petrinetze vorgestellt. Wie die Methoden auf Modelle mit den für *Workflow-Netze* eingeführten Erweiterungen wie unterscheidbaren Marken oder Routing-Konstrukten angewendet werden können, wird nicht erläutert. Gleiches gilt für die Ansätze zur Leistungsbewertung, beispielsweise in [vHR00], die ebenfalls ausschließlich für einfache Petrinetze entwickelt werden.

Zwar wird beispielsweise bei der Verifikation von *Workflow-Netzen* in [Aal98d] das dynamische Verhalten der Geschäftsprozeßmodelle untersucht, eine explizite graphische Darstellung des Verhaltens ist jedoch nicht möglich.

### **Andere petrinetzbasierte Modelle**

Abgesehen von den in diesem Abschnitt ausführlicher behandelten petrinetzbasierten Ansätzen *INCOME*, *FUNSOFT* und *Workflow Nets* existieren noch eine ganze Reihe weiterer Arbeiten, die Petrinetze zur Bearbeitung verschiedener, meist jedoch spezieller Problemstellungen im Kontext des Entwurfs von Geschäftsprozessen verwenden.

Besonders beliebt sind Petrinetze beim Entwurf von Geschäftsprozessen im Rahmen des Workflow Management (vgl. beispielsweise [AAH98, Aal98d] sowie die Ausführungen in Abschnitt 2.3.2.). Dort werden sie insbesondere zur Modellierung und Analyse einzelner Fragestellungen wie dem *Dynamic Change*-Problem benutzt (vgl. beispielsweise [AM98, EN93, EKR95]). Eine Übersicht beziehungsweise eine ganze Sammlung von Artikeln zur Nutzung von Petrinetzen im Bereich Workflow Management findet sich in [AME98].

Darüber hinaus haben sich Petrinetze vor allem für die Modellierung und Analyse von Geschäftsprozessen im Bereich der Produktion bewährt [DA94, DHP<sup>+</sup>93, PX96]. So werden beispielsweise in [Reu95] spezielle Petrinetze zur Analyse rechnergestützter Fertigungssysteme genutzt.

In der Literatur finden sich jedoch auch durchaus kritische Anmerkungen zur Verwendung von Petrinetzen für den Geschäftsprozeß-Entwurf. So stoßen Petrinetze aufgrund ihrer vermeintlichen Komplexität auf teilweise erhebliche Akzeptanzprobleme in der Praxis [Uth01]. Zudem betonen Petrinetze vor allem die Aktivitäten eines Geschäftsprozesses, trotz einzelner Erweiterungen zur Integration von Informations- oder Organisationsaspekten gestaltet sich die Modellierung dieser Aspekte dennoch schwierig [Kel99, Rum99]. Petrinetze eignen sich damit vor allem als integrierende Modelle der *Steuerungssicht* (vgl. Abschnitt 2.3.2), für die Modellierung der übrigen Sichten muß gegebenenfalls auf andere Methoden und Formalismen zurückgegriffen werden.

### **2.3.4 Fazit**

Keiner der in den vorangegangenen Abschnitten untersuchten Ansätze für den Entwurf von Geschäftsprozessen deckt die in Abschnitt 2.2.4 formulierten Anforderungen für den Entwurf von Geschäftsprozessen unter Berücksichtigung aller in Abschnitt 2.2 als relevant identifizierten Qualitätsmerkmale in zufriedenstellender Weise ab.

Die Abschnitt 2.3.1 behandelten allgemeineren Ansätze haben meistens eher den Charakter von *Vorgehensmodellen* ohne konkrete Methoden zur Umsetzung der beschriebenen Schritte oder Phasen beim Entwurf von Geschäftsprozessen. Aber auch die verschiedenen modellbasierten Ansätze sowie die vorgestellten petrinetzbasierten Ansätze halten keine vollständige Lösung parat. Das Spektrum der festgestellten Defizite reicht dabei von unzureichender Unterstützung bei der Modellierung bis zur fehlenden Berücksichtigung quantitativer Aspekte im Rahmen einer Leistungsbewertung. Eine Reihe von Ansätzen

beschäftigt sich lediglich mit der Behandlung einzelner, isolierter Fragestellungen und eignet sich daher nur für ganz spezielle Aufgaben im Rahmen des Geschäftsprozeß-Entwurfs. Keiner der untersuchten Ansätze unterstützt in einer geeigneten Weise den Entwurf von Geschäftsprozessen mit nicht-strukturierten Teilen.

Beginnend mit dem nächsten Kapitel wird in dieser Arbeit mit dem *VIPbusiness*-Ansatz eine Methode vorgestellt, die ausgehend von einem auf höheren Petrinetzen basierenden Modellierungsfomalismus den Entwurf von Geschäftsprozessen unter Berücksichtigung aller relevanten Qualitätsmerkmale bezüglich der Aspekte Korrektheit, Konsistenz und Effizienz ermöglicht.

# Kapitel 3

## Der *VIPbusiness*-Ansatz

Mit *VIPbusiness* wird in diesem Kapitel ein 3-stufiger Ansatz für den Entwurf von Geschäftsprozessen eingeführt. Dazu werden in Abschnitt 3.1 zunächst die drei Hauptschritte *Modellierung*, *Simulation* sowie *Analyse und Optimierung* und die dabei verwendeten Konzepte und Techniken detailliert erläutert und anschließend in ein Rahmenkonzept gebettet, das sowohl den ersten Entwurf eines Geschäftsprozesses als auch dessen systematische Verbesserung zu einem qualitativ hochwertigen Entwurf unterstützt.

Da ein Ziel dieser Arbeit die Entwicklung eines *werkzeugunterstützten* Ansatzes für den Geschäftsprozeß-Entwurf ist, wird in Abschnitt 3.2 zunächst ein entsprechendes Anforderungsprofil erarbeitet. Ausgehend von diesem Anforderungsprofil werden kommerzielle Werkzeuge für den Geschäftsprozeß-Entwurf bezüglich ihrer Eignung zur Unterstützung des *VIPbusiness*-Ansatzes untersucht. Angesichts der dabei festgestellten Defizite erscheint die Entwicklung eines eigenen Werkzeugs in Form einer Weiterentwicklung des bereits existierenden *VIPtools* als sinnvolle Lösung für die Umsetzung der Werkzeugunterstützung des *VIPbusiness*-Ansatzes. Das daraus resultierende Profil des *VIPtools* mit *VIPbusiness*-Erweiterungen bildet den Abschluß dieses Kapitels.

### 3.1 Ein 3-stufiger Ansatz

Der *VIPbusiness*-Ansatz beinhaltet die drei Schritte *Modellierung*, *Simulation* sowie *Analyse und Optimierung*, die solange wiederholt werden, bis der Geschäftsprozeß-Entwurf den Qualitäts-Anforderungen entspricht (vgl. Abbildung 3.1).

Die Grundidee dieses Ansatzes ist einfach: in *Schritt 1* wird ein Modell des Geschäftsprozesses erstellt. In *Schritt 2* wird im Rahmen einer Simulation das Verhalten des Geschäftsprozeßmodells generiert und gespeichert. Das generierte Verhalten bildet die Grundlage für die Anwendung einer Reihe von Verfahren zur Analyse und Optimierung in *Schritt 3*.

In den folgenden Abschnitten werden die drei Schritte des Ansatzes vorgestellt. Wie eingangs dieser Arbeit bereits erwähnt (vgl. Abschnitt 1.4), werden dabei die jeweils verwendeten Konzepte und Techniken kurz erläutert und die Vor- und Nachteile ihres Einsatzes diskutiert. Auf diese Weise können die in dieser Arbeit verwendeten Konzepte und Techniken und die damit verbundenen Neuheiten kompakt dargestellt werden. Die Erläuterung der in den folgenden Abschnitten benutzten Begriffe und Konzepte mit allen technischen Details erfolgt in den Kapiteln 4, 5, 6 und 7.

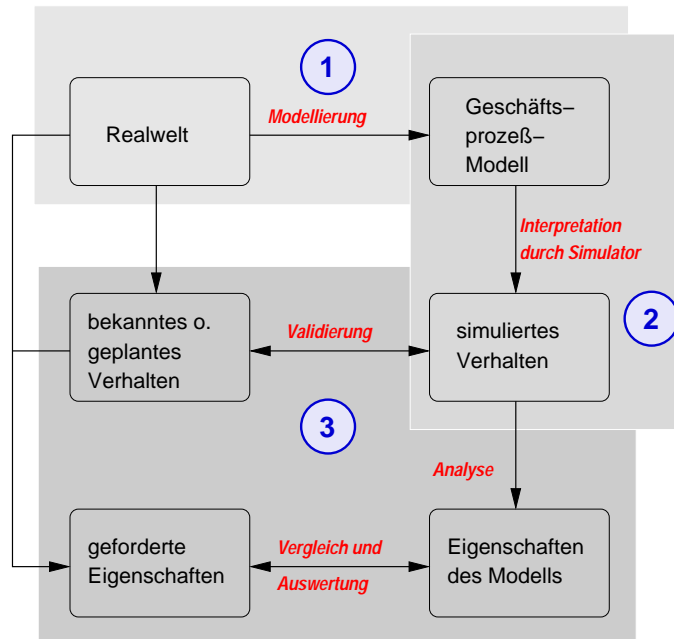


Abbildung 3.1: *VIPbusiness*: ein 3-stufiger Ansatz.

Welches Rahmenkonzept für die Durchführung der einzelnen Schritte vorgesehen ist, wird in Abschnitt 3.1.4 erläutert.

### 3.1.1 Schritt 1: Modellierung

Das Erstellen eines Geschäftsprozessmodells ist das Ziel von *Schritt 1* des *VIPbusiness*-Ansatzes. In diesem Abschnitt wird erläutert, welche Eigenschaften und Besonderheiten der in dieser Arbeit verwendete Modellierungsfomalismus hat.

#### Graphische Modellierung mit formaler Semantik

Die Grundlage für den in dieser Arbeit zur Erstellung von Geschäftsprozessmodellen verwendeten Formalismus bildet mit *VIPbusiness*-Netzen eine neu entwickelt Variante höherer Petrinetze. Wie in Abschnitt 2.3.2 bereits geschildert, ist die Verwendung von Petrinetzen zur Modellierung von Geschäftsprozessen und Workflows bereits etabliert. Einige Vorteile von Petrinetzen in diesem Kontext sind offensichtlich (vgl. [Aal98d, AAH98, EKR95]):

- Petrinetze verbinden die Möglichkeit der graphischen Modellierung mit einer formalen Semantik.

Auf diese Weise erleichtern Petrinetze dem Modellierer seine Arbeit, die erstellten Modelle eignen sich als Diskussionsgrundlage für alle Beteiligten. Die formale Semantik erlaubt eine Interpretation durch eine Maschine und somit die Simulation sowie Analyse und Optimierung der Entwürfe mit Hilfe von Werkzeugen. Darüber hinaus führt eine formale Semantik dazu, daß die erstellten Modelle von jedem Betrachter in gleicher Weise interpretiert werden.

- Petrinetze verfügen über eine große Ausdrucksmächtigkeit.

Durch geeignete Erweiterungen der jeweiligen Klasse von Petrinetzen lassen sich Modelle für sehr unterschiedliche Anwendungs-Kontexte erstellen. In dieser Arbeit werden dazu eine Reihe von Erweiterungen eingeführt, die jedoch auf herkömmliche Petrinetz-Konzepte zurückgeführt werden können, die Semantik ist somit eindeutig via dieser Transformationen. Die so entstandene Klasse der *VIPbusiness*-Netze weist dabei neben einem (später noch erläuterten) Zeit- und Kosten-Konzept sowie der Unterstützung nicht vollständig strukturierter Geschäftsprozesse die folgenden Erweiterungen auf:

- Routing-Konstrukte

Für die Modellierung der Reihenfolgebeziehungen der Aktivitäten werden entsprechende Bausteine benötigt. Dazu wird in dieser Arbeit die in [Aal98d] vorgeschlagene graphische Notation für Routing-Konstrukte übernommen und mit einer Semantik versehen, die eine Simulation der erstellten Modelle ermöglicht.

- Aktivitäts-Transitionen als Module

Zur Modellierung von Aktivitäten werden mit Aktivitäts-Transitionen Bausteine entwickelt, die eine Kapselung von Modellierungs-Details durch Verwendung von vordefinierten Modulen erlauben. Darüber hinaus wird durch Verwendung dieser Bausteine die Konstruktion von Modellen ermöglicht, die zustandsbasiert (*state-based*, vgl. [Aal98d]) sind.

- Explizite Modellierung von Ressourcen

Der Verbrauch von Ressourcen bei der Ausführung von Aktivitäten wird bei den meisten Modellierungs-Ansätzen nur implizit modelliert, beispielsweise als Teil eines Warteschlangenmodells [Kle75]. Dieses Vorgehen hat den Nachteil, daß die Effekte des Ressourcen-Einsatzes ebenfalls nur implizit dargestellt werden können. So lassen sich beispielsweise Engpässe nur anhand eines hohen Auslastungsgrads oder hoher Durchlaufzeiten beobachten, die Ursache eines Engpasses bleibt jedoch unklar. In dieser Arbeit wird daher ein andere Ansatz gewählt: durch explizite Modellierung der Ressourcen in Form spezieller *Ressourcen-Stellen*<sup>1</sup> werden in Kombination mit den Möglichkeiten zur expliziten Visualisierung des Modellverhaltens (siehe S. 3.1.2) die Effekte des Ressourcen-Einsatzes bei der Simulation der Geschäftsprozeßmodelle unmittelbar sichtbar. Auf diese Weise wird die Verbesserung von Entwurfsmängeln erheblich erleichtert.

- Petrinetze sind nicht-proprietär.

Es besteht keine Abhängigkeit von einem bestimmten Hersteller für Werkzeuge.

- Für Petrinetze existiert eine Vielzahl von Analyseverfahren.

Die Übersetzbarkeit der verwendeten Netzklasse vorausgesetzt, können auf diese Weise viele bereits für Petrinetze existierende Verfahren und Werkzeuge für die Analyse der eigenen Modelle verwendet werden. In Abschnitt 8.1.3 wird dieser Umstand für eine Verknüpfung der weiterentwickelten Version des *VIPtools* mit anderen Werkzeugen genutzt.

---

<sup>1</sup>Wie eingangs bereits erwähnt, werden insbesondere die Begriffe aus der Petrinetz-Theorie in den nachfolgenden Kapiteln 4, 5 und 6 erläutert.

## Zustandsbasierte Modellierung

Bei den meisten Modellierungsformalismen werden keine Zustände zwischen Aktivitäten berücksichtigt, sie sind ausschließlich *ereignisbasiert* (*event-based*, vgl. [Aal98d] und Abschnitt 3.2.2). Dem vermeintlichen Gewinn an Übersichtlichkeit stehen jedoch eine Reihe von Nachteilen gegenüber. So können beispielsweise implizite Auswahlentscheidungen (vgl. Abschnitt 5.2) nicht modelliert werden (vgl. auch [AHKB00a]). In Verbindung mit einem Zeit- und Kostenkonzept kommt es zu Schwierigkeiten bei der Berücksichtigung echter Wartezeit (vgl. Abschnitt 5.4).

Die mit Hilfe der *VIPbusiness*-Bausteine erstellten Modelle sind daher ausschließlich zustandsbasiert, in der dynamischen Betrachtung handelt es sich um Modelle mit diskreten Zustandsübergängen. Die Ausführung einer Aktivität wird in diesem Zusammenhang als „Zustand, in dem eine Aktivität ausgeführt wird“ interpretiert. Die Ereignisse, die zu Zustandsübergängen führen, sind der Beginn oder das Ende der Ausführung einer Aktivität. Bei der Modellierung von Aktivitäten wird dieses Konzept mit Hilfe der Aktivitäts-Transitionen umgesetzt. Darauf aufbauend kann ein zustandsbasiertes Zeit- und Kostenkonzept entwickelt werden.

## Zeit- und Kostenkonzept

Ausgehend von den Nachteilen existierender Zeitkonzepte für Petrinetze (vgl. dazu auch Abschnitt 5.4.2) wird in dieser Arbeit ein neues, ausschließlich auf Zuständen bzw. Stellen basierendes Zeitkonzept entwickelt. Dieses Zeitkonzept verwendet lokale Zeiten und vermeidet auf diese Weise den Gebrauch von Uhren oder einer Systemzeit. Die Berücksichtigung von Zuständen zwischen Aktivitäten ermöglicht zudem eine Zuordnung eventuell bei der Ausführung der Aktivitäten anfallender echter Wartezeit.

Durch die Trennung von Zeit und Ereignissen wird erreicht, daß die Simulation der Geschäftsprozeßmodelle unabhängig von gewählten Zeit- oder Kostengrößen erfolgen kann (siehe auch Abschnitt 3.1.2). Durch die Verwendung verschiedener, auch zufällig erzeugter Werte können damit auf effiziente Weise die Auswirkungen verschiedener Zeit- und Kostenbewertungen auf wichtige Aspekte der Leistungsbewertung wie Durchlaufzeit oder Ressourcenauslastung überprüft werden (vgl. Abschnitt 7.5).

## Modellierung nicht-strukturierter Segmente und hybride Spezifikationen

Eines der Ziele dieser Arbeit ist die Unterstützung des Entwurfs von Geschäftsprozessen, die nicht vollständig strukturiert sind (vgl. Abschnitt 1.3). Dazu wird ein deklarativer Ansatz verwendet. Statt alle Möglichkeiten der Ausführung der Aktivitäten eines nicht-strukturierten Teils eines Geschäftsprozesses durch explizite Modellierung aller Alternativen zu berücksichtigen, wird nicht versucht, dem nicht-strukturierten Teil mittels konstruktiver Modellierungs-Ansätze eine Struktur aufzuzwingen. Die Aktivitäten eines nicht-strukturierten Teils werden vielmehr durch eine ungeordnete Menge von Netzfragmenten modelliert, die bei der Ausführung des Geschäftsprozeßmodells miteinander kombiniert werden können. Um nicht erwünschte Kombinationen der Fragmente zu verhindern, werden zusätzlich Regeln spezifiziert, die bei der Ausführung des Geschäftsprozeßmodells



grundsätzlich nicht verletzt werden dürfen. Anschaulich wird dieses Konzept durch die *Röhre* in Abbildung 3.2 dargestellt. Die bei der Bearbeitung von Cases nötigen Aktivitäten können in beliebiger Weise ausgeführt werden, jedoch nur innerhalb der durch Business Rules aufgespannten Wände der Röhre.



Abbildung 3.2: Das Röhren-Konzept.

Da nicht vollständig strukturierte Geschäftsprozesse in der Regel sowohl strukturierte als auch nicht-strukturierte Teile enthalten, werden die mittels Netzfragmenten und Business Rules spezifizierten nicht-strukturierten Teile der Geschäftsprozeßmodelle mit strukturierten Teilen zu *hybriden Spezifikationen* kombiniert (vgl. Abschnitt 5.6).

### 3.1.2 Schritt 2: Simulation

In *Schritt 2* des *VIPbusiness*-Ansatzes wird das Verhalten des Geschäftsprozeßmodells im Rahmen von Simulation generiert. Die Grundlage für die entsprechenden Verfahren bilden die im Rahmen des *VIP*-Projekts entwickelten Simulationskonzepte (vgl. Abschnitt 6.2).

#### Kausale Semantik und Halbordnungen

Grundsätzlich wird bei der Beschreibung des Verhaltens von Petrinetzen zwischen *sequentieller* Semantik (in Form von Schaltfolgen) und *kausaler* Semantik (in Form von halbgeordneten Abläufen) unterschieden. Bei *VIPbusiness* wird eine kausale Semantik verwendet, um die Probleme sequentieller Semantik bei der Beschreibung nebenläufigen Systemverhaltens zu vermeiden (vgl. Abschnitt 6.1.2).

#### Graphische Darstellung des generierten Verhaltens

Halbgeordnete Abläufe von Petrinetzen lassen sich mit Hilfe einer speziellen Klasse von Petrinetzen, den Kausalnetzen, graphisch darstellen. Auf diese Weise wird die Validierung des generierten Systemverhaltens erheblich erleichtert und die explizite Analyse bestimmter Effekte, beispielsweise die Auswirkungen beschränkter Ressourcen, ermöglicht.

## Integration der VIPbusiness-Erweiterungen

Da alle für die Erstellung der Geschäftsprozeßmodelle eingeführten Modellierungs-Bausteine auf üblichen Konzepten höherer Petrinetze basieren (beziehungsweise auf einfache Art auf diese zurückgeführt werden können), bereitet die Simulation der *VIPbusiness*-Netze keine grundsätzlichen Schwierigkeiten. Für die einzelnen Erweiterungen, insbesondere die zur Modellierung von nicht vollständig strukturierten Geschäftsprozessen eingeführten Konzepte, muß jedoch geklärt werden, wie sie in den Abläufen dargestellt werden sollen und wie ihre Verwendung im Modell auf Ebene der Abläufe zu interpretieren ist (vgl. Abschnitt 6.3).

## Effizienz und Vergleichbarkeit

Durch die im vorangegangenen Abschnitt bereits beschriebene Zuordnung von Zeit und Kosten zu Stellen kann die Simulation des Geschäftsprozeßmodells ohne Berücksichtigung von Zeit- und Kostenwerten durchgeführt werden. Dieser Vorgehensweise liegt folgende Idee zugrunde:

- Kernstück der Modellierung eines Geschäftsprozesses ist zunächst die Modellierung der *kausalen Struktur*, d.h. es wird festgelegt, welche Aktivitäten in welcher Reihenfolge auszuführen sind.
- Ausgehend von der modellierten kausalen Struktur werden halbgeordnete Abläufe erzeugt. Damit wird der ursprünglichen Intention von Carl-Adam Petri Rechnung getragen, daß das Verhalten von Petrinetzen nur durch die kausale Struktur des modellierten Systems determiniert sein soll [ABC<sup>+</sup>95]. Da bei der Generierung der Abläufe Konfliktsituationen gelöst und Zyklen durchlaufen werden müssen, nimmt dieser Vorgang in der Regel sehr viel Rechenzeit in Anspruch.
- Ist eine entsprechende Menge von Abläufen erzeugt worden, können Zeit- und Kostenbewertungen aus dem Systemnetz in die erzeugten Abläufe integriert werden. Für die einzelnen Abläufe lassen sich jetzt Kenngrößen wie beispielsweise die Durchlaufzeit ermitteln.

Dieses Vorgehen ist mit zwei entscheidenden Vorteilen verbunden: Effizienz und Vergleichbarkeit.

Da die zur Leistungsbewertung benötigten Algorithmen sehr viel weniger Rechenzeit in Anspruch nehmen als die Generierung der Abläufe, ist es auf diese Weise möglich, mit verschiedenen Zeit- und Kostenbewertungen im Systemnetz zu experimentieren. Die Auswirkungen unterschiedlicher Bewertungen auf das Verhalten des Geschäftsprozesses lassen sich durch Analyse der Abläufe sehr schnell ermitteln.

Ein weiterer und wesentlicher Vorteil liegt in der Vergleichbarkeit der Ergebnisse. Für den Fall, daß Zeit und Kosten das Verhalten des modellierten Systems beeinflussen, ist das im Rahmen der Simulation mit unterschiedlichen Zeit- und Kostenwerten generierte Verhalten ebenfalls unterschiedlich. Die sich durch Anwendung der Verfahren zur Leistungsbewertung ergebenden Werte, beispielsweise für Durchlaufzeiten, sind damit nur noch bedingt

vergleichbar. Der *VIPbusiness*-Ansatz vermeidet diesen Nachteil und ermöglicht so die systematische Analyse des modellierten Geschäftsprozesses mit verschiedenen Zeit- und Kostenbewertungen.

### 3.1.3 Schritt 3: Analyse und Optimierung

Ein entscheidender Vorteil des *VIPbusiness*-Ansatzes besteht in der Möglichkeit, alle drei als relevant identifizierten Arten der Analyse - Validierung, Verifikation und Leistungsbewertung - anhand eines einzelnen Modells durchführen zu können. Dazu werden im Rahmen dieser Arbeit eine Reihe bekannter Verfahren zur Analyse von Petrinetzen für die Analyse von petrinetzbasierten Geschäftsprozeßmodellen nutzbar gemacht. Für den Bereich der Leistungsbewertung wurden zudem eine Reihe neuer Verfahren und Konzepte entwickelt.

#### Validierung

Im Bereich der Validierung spielt die oben bereits erwähnte explizite Darstellung des generierten Verhaltens mittels Kausalnetzen eine herausragende Rolle. In Verbindung mit der Unterstützung durch das *VIPtool*, das unter anderem einen Browser für eine systematische Durchsicht der einzelnen Abläufe zur Verfügung stellt, wird so ein überprüfender Vergleich des generierten mit dem intendierten Verhalten erheblich erleichtert.

#### Verifikation

Die Vielzahl existierender Verfahren für den formalen Nachweis von Eigenschaften ist einer der großen Vorteile, den die Verwendung von Petrinetzen mit sich bringt. Durch die Verknüpfung des *VIPtools* mit dem Software-Werkzeug *Woflan* (vgl. Abschnitt 8.1.3) sind Eigenschaften wie *Lebendigkeit*, *Beschränktheit* oder *Free-Choice-Eigenschaft* überprüfbar. Die Relevanz dieser formalen Merkmale als wünschenswerte Eigenschaften von Geschäftsprozeßmodellen wird in [Aal98d] mit Hilfe des Begriffs der *Soundness* eines Modells erläutert.

Eine besondere Rolle im Zusammenhang mit der Überprüfung formaler Eigenschaften von Geschäftsprozeßmodellen spielen *Business Rules*. Diese wurden bereits für die Einschränkung des möglichen Verhaltens nicht-strukturierter Teile von Geschäftsprozessen verwendet. Durch eine Erweiterung des im Rahmen des *VIP*-Projekts erarbeiteten Konzepts der *Anfrage-Transitionen* lassen sich logische Anforderungen an das Modellverhalten spezifizieren. Die Erfüllung oder Verletzung dieser Anforderungen kann anhand der generierten Abläufe überprüft werden. Insbesondere läßt sich auf diese Weise die *restriktive Korrektheit* (der Verlauf „innerhalb“ der festgelegten Röhre) von Abläufen hybrider Spezifikationen untersuchen.

#### Leistungsbewertung

Die Verfahren des *VIPbusiness*-Ansatzes für den Bereich Leistungsbewertung erlauben die Analyse aller relevanten Merkmale in diesem Bereich (vgl. Abschnitt 2.2.3), beispielsweise die Ermittlung der Durchlaufzeit oder des Auslastungsgrads von Ressourcen.

Durch die explizite Modellierung von Zuständen wird die Ermittlung und die exakte Zuordnung echter Wartezeit ermöglicht. Bei der Zuordnung kann zusätzlich eine kostenoptimale Verteilung vorgenommen werden.

Die Unabhängigkeit der Simulation von Zeit- und Kostenwerten erlaubt die systematische Analyse der Effekte verschiedener Schätz- und Sollwerte auf die Leistungsbewertung des Geschäftsprozeß-Entwurfs. Mit Hilfe von *Konfigurationen* lassen sich dabei verschiedene, insbesondere zufällig erzeugte, Werte testen, bei Verwendung des *VIPtools* auch automatisch.

Die explizite Modellierung von Ressourcen mittels Ressourcen-Stellen sowie die graphische Darstellung des generierten Verhaltens ermöglichen eine genaue Analyse der Effekte des Einsatzes knapper Ressourcen. Dabei kann es beim gemeinsamen Zugriff verschiedener Aktivitäten auf dieselben Ressourcen zu *Linearisierungs-Effekten* kommen, die sich üblicherweise nur durch höhere Auslastung der Ressourcen oder höhere Durchlaufzeiten nachweisen lassen. Mit Hilfe des *VIPbusiness*-Ansatzes werden diese Effekte explizit dargestellt, Ressourcen-Engpässe können so gezielt beseitigt werden (vgl. Abschnitt 7.4).

Im Zusammenhang mit der Verwendung von Geschäftsprozeßmodellen zur Steuerung von Informationssystemen, beispielsweise im Zusammenhang mit Workflow-Systemen [Obe96, AE01], ist die Verwendung von Kontrollparametern zur Beeinflussung von Auswahlentscheidungen bei der Bearbeitung von Cases von großer Bedeutung. So kann beispielsweise festgelegt werden, für welchen Anteil der Cases eine Überprüfung von Bearbeitungsschritten vorgenommen werden soll. Der *VIPbusiness*-Ansatz beinhaltet dazu ein Verfahren, das eine Konfiguration entsprechender Kontrollparameter unter Berücksichtigung anderer Leistungsparameter in Abhängigkeit von verfügbaren Ressourcen ermöglicht. Mit Hilfe der Kontrollparameter lassen sich zudem verschiedene Szenarien realisieren, die bei der Bearbeitung echter Cases zur Laufzeit eintreten können, beispielsweise ein besonders hoher Anteil fehlerhafter Bearbeitungen, die eine Wiederholung von Aktivitäten erfordern. Durch Analyse der Geschäftsprozeßmodelle für die jeweilige Kontrollparameter-Konfiguration läßt sich die Eignung des Geschäftsprozeß-Entwurfs für derartige Szenarien überprüfen.

### 3.1.4 Arbeiten mit *VIPbusiness*

In diesem Abschnitt wird das Vorgehen beim Entwurf von Geschäftsprozessen, das mit der Durchführung der drei Schritte des *VIPbusiness*-Ansatzes verbunden ist, noch einmal zusammenfassend behandelt. Dabei wird auf den Entwurf von Geschäftsprozessen als iterativen Prozeß eingegangen, es werden die mit der Anwendung der *VIPbusiness*-Konzepte verbundenen Vorteile beleuchtet sowie mögliche Kontexte für den Einsatz des *VIPbusiness*-Ansatzes betrachtet.

#### **Durchführung der Schritte als iterativer Prozeß**

Aus der Anwendung der in den vorangegangenen Abschnitten beschriebenen Konzepte der einzelnen Schritte des *VIPbusiness*-Ansatzes läßt sich folgendes Vorgehen für den Entwurf von Geschäftsprozessen entwickeln (vgl. Abbildung 3.3):

- *Schritt 1*: Der Geschäftsprozeß wird als *VIPbusiness*-Netz modelliert.

Ausgangspunkt dieses Schrittes ist eine Analyse der Ist-Situation beziehungsweise eine Formulierung einer Soll-Situation mit Hilfe geeigneter Techniken, beispielsweise der Informationsbedarfs-Ermittlung und -Analyse oder Interviewtechniken (vgl. dazu auch Abschnitt 2.3). Die Soll-Situation wird mittels einer Reihe von Anforderungen formuliert, die im Geschäftsprozeßmodell umgesetzt beziehungsweise im Rahmen der Analyse des erstellten Modells überprüft werden müssen.

Kosten- und Zeitaspekte werden bei der Modellierung zunächst außer acht gelassen. Damit kann beim Modellieren der Schwerpunkt auf die *logische Struktur*, also auf die auszuführenden Aktivitäten, die entsprechenden Vor- und Nachbedingungen der Ausführung sowie die sich daraus ergebende Reihenfolge der Aktivitäten (vgl. Kapitel 2) des Geschäftsprozesses gelegt werden. Die Diskussion von Detailfragen der Art „Was soll die Durchführung der Aktivität kosten?“ kann zu diesem Zeitpunkt vermieden werden. Insbesondere sind keine Entscheidungen über anzusetzende Schätz- und Sollwerte im Modell zu fällen (vgl. dazu die Ausführungen in Abschnitt 2.2.3 und 2.2.3).

Wünschenswerte strukturelle Eigenschaften von Geschäftsprozeß-Entwürfen wie beispielsweise die *well-handledness* (vgl. Abschnitt 2.2.2) können bereits anhand der modellierten *VIPbusiness*-Netze überprüft werden. In speziellen Fällen ist es darüber hinaus möglich, aus den strukturellen Eigenschaften des *VIPbusiness*-Netzes Rückschlüsse auf das mögliche Verhalten des modellierten Geschäftsprozesses zu ziehen. Durch Anwendung entsprechender Analyseverfahren können potentielle Fehler beim Entwurf ausgeschlossen und somit die unnötige Simulation fehlerhafter Geschäftsprozeßmodelle vermieden werden.

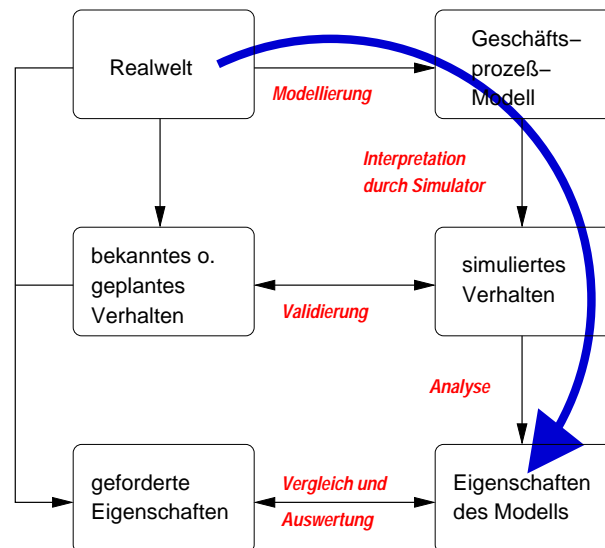


Abbildung 3.3: *VIPbusiness*: Durchführung der drei Schritte.

- *Schritt 2:* Die Abläufe zu dem Geschäftsprozeßmodell werden generiert und zur späteren Analyse gespeichert.

Je nach Größe des Geschäftsprozeßmodells wird dieser rechen- und zeitintensive Vorgang unter Ausschluß der an der Gestaltung des Geschäftsprozesses Beteiligten („über Nacht“) erfolgen. Dadurch wird unproduktive Wartezeit und die damit verbundene Frustration vermieden. Die Menge der erzeugten Abläufe ist das Ergebnis der in Schritt 1 modellierten logischen Struktur des Geschäftsprozesses. Solange die logische Struktur nicht verändert wird, muß die Menge der Abläufe nur einmal erzeugt werden. Alle weiteren Analysen lassen sich anhand dieser Menge durchführen.

Durch eine Darstellung der halbgeordneten Abläufe mit Hilfe von Kausalnetzen ist eine graphische Repräsentation des Systemverhaltens möglich. Damit ist eine anschauliche Grundlage für die Analyse des Geschäftsprozeßverhaltens im Rahmen einer Diskussion der Beteiligten geschaffen. Eine wichtige Anforderung für die Validierung von Geschäftsprozeßmodellen ist damit erfüllt (vgl. die Abschnitte 2.2.1 und 2.2.4).

- *Schritt 3:* Die generierten Abläufe werden zur Analyse des Geschäftsprozeß-Entwurfs herangezogen.

Der Schwerpunkt liegt hier auf der Analyse bezüglich Merkmalen aus dem Bereich der Leistungsbewertung. Dazu wird das Geschäftsprozeßmodell um Zeit- und/oder Kostenbewertungen ergänzt. Diese Bewertungen können (mittels geeigneter, im weiteren Verlauf dieser Arbeit eingeführter Abbildungsfunktionen) auf die generierten Abläufe übertragen werden. Da die Verfahren zur Leistungsbewertung auf die bereits erzeugten Abläufe zurückgreifen können, ist der Rechenaufwand bei Anwendung dieser Verfahren erfahrungsgemäß geringer als bei der Generierung der Abläufe in Schritt 2. Auf diese Weise ist es möglich, die Auswirkungen der vorgenommenen Zeit- und Kostenbewertungen sofort zu überprüfen.

Darüber hinaus eignen sich halbgeordnete Abläufe auch zur Beantwortung von Fragestellungen aus dem Bereich Verifikation, beispielsweise kann für Abläufe von Geschäftsprozeßmodellen mit nicht-strukturierten Segmenten die restriktive Korrektheit überprüft werden.

In den wenigsten Fällen wird die einmalige Durchführung dieser drei Schritte dazu führen, daß der Geschäftsprozeß-Entwurf allen Anforderungen genügt. Häufiger ist vielmehr, daß zunächst Anforderungen vergessen oder falsch formuliert werden, Anforderungen geändert und Fehler bei der Umsetzung im Geschäftsprozeßmodell beseitigt werden müssen. In diesem Fall werden die bei der Durchführung der drei Schritte gewonnenen Erkenntnisse genutzt, um gegebenenfalls eine Anpassung der Soll-Situation und der damit verbundenen Anforderungen vorzunehmen. Anschließend, beziehungsweise unmittelbar nach Durchführung einer Iteration (für den Fall, daß der erste Entwurf den bereits formulierten Anforderungen nicht genügt, bei der Modellierung also Fehler gemacht wurden), wird das Geschäftsprozeßmodell entsprechend geändert und wieder analysiert (vgl. Abbildung 3.4). Dieses Vorgehen wird wiederholt, bis der Geschäftsprozeß-Entwurf allen für die Soll-Situation formulierten Qualitäts-Anforderungen entspricht.

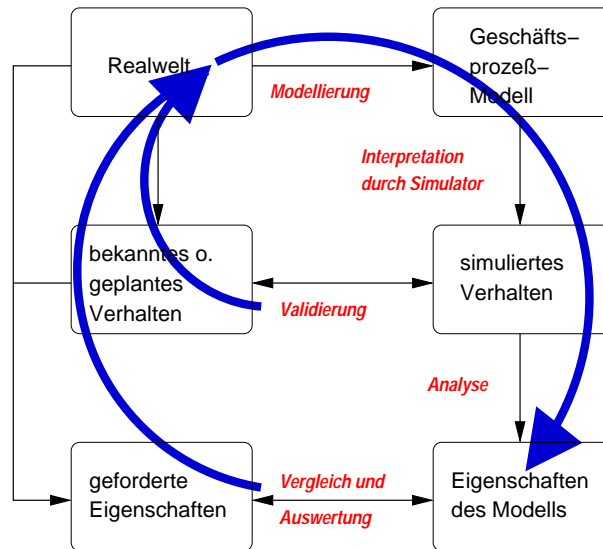


Abbildung 3.4: *VIPbusiness*: Iteration.

### Vorteile des *VIPbusiness*-Ansatzes

Die Vorteile der bei der Umsetzung der drei Schritte des *VIPbusiness*-Ansatzes verwendeten Konzepte wurde bereits in den Abschnitten 3.1.1, 3.1.2 und 3.1.3 erläutert. In diesem Abschnitt werden daher nur die Möglichkeiten betrachtet, die sich aus der Anwendung des *VIPbusiness*-Ansatzes als Ganzes ergeben<sup>2</sup>.

Insgesamt unterstützt die Anwendung des *VIPbusiness*-Ansatzes den Entwurf von Geschäftsprozessen durch Möglichkeiten zur Erstellung von Geschäftsprozessmodellen sowie deren *interaktive* Analyse anhand der generierten Abläufe. Daraus ergeben sich eine Reihe von *Vorteilen*:

- Die an einen Geschäftsprozeß-Entwurf gestellten Qualitäts-Anforderungen lassen sich sofort überprüfen.

Insbesondere bei der Diskussion alternativer Gestaltungsmöglichkeiten eines Geschäftsprozesses sollten Entscheidungen nicht ausschließlich „aus dem Bauch heraus“ gefällt werden. Die Nutzung der Analysemöglichkeiten des *VIPbusiness*-Ansatzes ist dabei eine große Hilfe. Durch die Möglichkeiten zur Leistungsbewertung von Geschäftsprozeß-Entwürfen in frühen Phasen des Entwurfs können insbesondere Effizienzbehauptungen sehr schnell bestätigt oder widerlegt werden. So läßt sich für einen Geschäftsprozeß-Entwurf beispielsweise die Erfüllung eines Service Levels oder die strikte Einhaltung von Business Rules garantieren.

- Die Abhängigkeit der Leistungsbewertung des modellierten Geschäftsprozesses von Kosten- und Zeitbewertung kann untersucht werden.

<sup>2</sup>Eine klare Trennung ist hier natürlich nicht möglich, einige der angeführten Punkte ergeben sich unmittelbar aus den zur Durchführung der jeweiligen Schritte verwendeten Konzepten.

Für die Bewertungen im Geschäftsprozeßmodell muß auf Schätz- und Sollwerte zurückgegriffen werden (vgl. Abschnitt 2.2.3 und Abschnitt 2.2.3). Es muß daher untersucht werden, welche Auswirkungen die Variation der Bewertungen auf die Leistungsbewertung des modellierten Geschäftsprozesses hat. Dabei ist vor allem der Vergleich eventueller Gestaltungsalternativen von Interesse. Je nach Risikofreudigkeit der Entscheider wird dabei der Alternative mit der absolut besten Leistungsbewertung (für eine bestimmte Zeit- und Kostenbewertung) oder der Alternative mit der (durchschnittlich) besten Leistungsbewertung für verschiedene (als realistisch erachtete) Zeit- und Kostenbewertungen der Vorzug gegeben werden. Wichtig ist in diesem Zusammenhang die bereits erläuterte *Vergleichbarkeit* der Analyseergebnisse für unterschiedliche Zeit- und Kostenbewertungen durch deren strikte Trennung von der Durchführung der Simulation (vgl. Abschnitt 3.1.2).

- Das Risiko schwerwiegender Fehler durch falsche Schätz- oder Sollwerte kann reduziert werden.

Durch die Möglichkeit, verschiedene Bewertungen relativ schnell ausprobieren zu können, läßt sich verhindern, daß die Entscheidung für eine Gestaltungsalternative aufgrund einiger weniger (eventuell unrealistischer) Zeit- und Kostenbewertungen gefällt werden muß.

Insgesamt wird also das Risiko einer Fehlentscheidung bei der letztendlichen Gestaltung eines Geschäftsprozesses wesentlich reduziert. Der Diskussion über die Gestaltungsalternativen wird mit der sofortigen Überprüfbarkeit von Qualitätsanforderungen, insbesondere Effizienzbehauptungen und „Aber was ist, falls...“-Einwürfen eine sachliche Grundlage gegeben. Die Motivation der Beteiligten wird durch die Möglichkeit zum interaktiven *Spiele*n mit unterschiedlichen Werte-Konfigurationen erhöht.

### **Kontext für den Einsatz von VIPbusiness**

Prinzipiell kann der *VIPbusiness*-Ansatz natürlich als alleiniges Werkzeug beim Entwurf von Geschäftsprozessen eingesetzt werden. Problematisch ist dabei jedoch die (gewollte) Spezialisierung der verwendeten Methoden auf den Entwurf von Geschäftsprozessen ausgehend von einem bereits bestehenden Anforderungsprofil. Konzepte zur Generierung eines solchen Anforderungsprofils fehlen ebenso wie allgemeine Konzepte des Projektmanagements oder Konzepte zur Umsetzung eines Geschäftsprozeß-Entwurfs. Im Kontext der Integration von Geschäftsprozessen und Informationssystemen und den damit verbundenen vielfältigen Anforderungen an Entwurfsmethoden und -werkzeuge erscheint eine Verwendung von *VIPbusiness* als *stand-alone*-Anwendung daher wenig empfehlenswert.

Viel sinnvoller ist der Einsatz von *VIPbusiness* als spezielle Methode zur Unterstützung einzelner Schritte eines auf die Durchführung entsprechender Projekte angelegten Vorgehensmodells. So ermöglicht die Anwendung der *VIPbusiness*-Konzepte die Erstellung von qualitativ hochwertigen Geschäftsprozeßmodellen, die beim Entwurf von Informationssystemen aufgrund ihrer Schnittstellenfunktion eine zentrale Rolle spielen (vgl. dazu Abschnitt 1.1 sowie Abbildung 1.1). Damit wird jedoch nur eine (wenn auch zentrale) der fünf Stufen des entsprechenden Ansatzes abgedeckt (die Stufe *Geschäftsbereichanalyse*), für alle anderen Stufen sind andere geeignete Methoden zu verwenden. Unabhängig von der Verknüpfung mit Informationssystemen eignet sich der *VIPbusiness*-Ansatz darüber hinaus zur Umsetzung wichtiger Teile der in Abschnitt 2.3 vorgestellten Vorgehensmodelle für



das Management von Geschäftsprozessen. Diese Verwendung der *VIPbusiness*-Konzepte als *plug-in* für andere Methoden mit einem weiter gefaßten Einsatzspektrum wird durch die Umsetzung in Form eines modular aufgebauten Werkzeugs auf ideale Weise unterstützt (vgl. dazu die Ausführungen im weiteren Verlauf dieses Kapitels beziehungsweise in Kapitel 8).

In den restlichen Abschnitten dieses Kapitels wenden wir uns den Grundlagen für eine Werkzeug-Unterstützung der in dieser Arbeit entwickelten Konzepte und Techniken zu.

## 3.2 Werkzeugunterstützung des *VIPbusiness*-Ansatzes

Ein wichtiges Ziel dieser Arbeit ist die Unterstützung der im Rahmen des *VIPbusiness*-Ansatzes entwickelten Methoden und Formalismen durch ein Werkzeug. Ausgehend von den in Abschnitt 2.2.4 im Zusammenhang mit der Untersuchung unterschiedlicher Qualitätsmerkmale identifizierten Anforderungen an die beim Entwurf von Geschäftsprozessen verwendeten Methoden und Formalismen wird in den folgenden Abschnitten untersucht, welchen Anforderungen ein Werkzeug zur Unterstützung dieser Methoden und Formalismen gerecht werden muß. Dazu wird in Abschnitt 3.2.1 zunächst ein Anforderungsprofil für Werkzeuge formuliert. Für die einzelnen Punkte des Profils werden Fragen formuliert, mit denen in Abschnitt 3.2.2 eine Reihe kommerzieller Werkzeuge für den Geschäftsprozeß-Entwurf untersucht werden. Auf Grundlage der Ergebnisse dieser Untersuchung werden in Abschnitt 3.2.3 die Profile der einzelnen Werkzeuge erstellt und in Abschnitt 3.2.4 mit dem dort erarbeiteten Profil des *VIPtools* verglichen.

### 3.2.1 Ein Anforderungsprofil für ein Werkzeug für den Geschäftsprozeß-Entwurf

Die Grundlage für die Untersuchung von Werkzeugen für den Geschäftsprozeß-Entwurf bildet ein Katalog von Fragen. Mit Hilfe dieser Fragen wird untersucht, ob ein Werkzeug den in Abschnitt 2.2.4 formulierten Anforderungen an Methoden und Formalismen gerecht wird. Dazu wird zunächst jedes Merkmal aus entsprechenden Anforderungen abgeleitet und die bei der Untersuchung des jeweiligen Werkzeuges zu beantwortende Frage formuliert. Die einzelnen Fragen werden dabei in 8 Gruppen zusammengefaßt, die sich auf die zwei übergeordneten Bereiche *Modellierungsformalismus* sowie Analyse und Simulation verteilen.

#### **Modellierungsformalismus**

Geschäftsprozeßmodellierungswerkzeuge sollten es dem Anwender ermöglichen, die Eigenschaften der Geschäftsprozesse möglichst einfach und exakt auf ein Modell zu übertragen. Dabei wird bereits berücksichtigt, daß die erstellten Modelle prinzipiell bezüglich aller drei relevanten Qualitätsaspekte und den zugehörigen Merkmalen analysierbar sein sollten. Dies setzt beispielsweise bei Analysen zur Leistungsbewertung voraus, daß quantitative Parameter, beispielsweise aus den Bereichen *Zeit* und *Kosten*, überhaupt im Modell abgebildet werden können. Für die einzelnen Werkzeuge wird daher untersucht, welchen der folgenden Anforderungen die bei der Modellierung von Geschäftsprozessen verwendeten Methoden und Formalismen gerecht werden:

- einige *grundlegende Eigenschaften* des Formalismus,
- Unterstützung von *Dimensionen* von Geschäftsprozessen,
- Möglichkeiten zur Modellierung von *quantitativen Informationen*,
- Unterstützung *nicht vollständig strukturierter* Geschäftsprozesse,
- Bausteine zur Realisierung von *Routing-Konstrukten*.

#### Merkmal 1: Grundlegende Eigenschaften

Als erstes wird untersucht, auf welche Weise der Anwender die Modelle der Geschäftsprozesse erstellen kann.

**Frage 1:** Gibt es eine graphische Benutzeroberfläche, die es mit Hilfe von Drag and Drop oder ähnlicher Methoden erlaubt, den Geschäftsprozeß mit Hilfe graphischer Objekte zu modellieren? Wird eine Programmiersprache zur Verfügung gestellt, mit der Eigenschaften des Geschäftsprozesse spezifiziert werden können (z.B. Entscheidungen bei Alternativen)?

Da ein Geschäftsprozeßmodell beschreiben soll, welche Aufgaben zur Bearbeitung eines Cases durchgeführt werden müssen, stellt sich die Frage nach der Modellierung von Aufgaben:

**Frage 2:** Werden Aufgaben explizit modelliert?

Einige Anforderungen bei der Modellierung und Analyse von Geschäftsprozessen erfordern die explizite Modellierbarkeit der Zustände zwischen den Aufgaben (vgl. Abschnitt 2.2.4). Als Beispiele seien hier die exakte Modellierung von Entscheidungen oder die Analyse von Wartezeiten genannt.

**Frage 3:** Ist es möglich, Zustände zwischen Aufgaben explizit zu modellieren?

Die Ergebnisse der Untersuchung bezüglich der Fragen 1 bis 3 finden sich in Tabelle 3.1 in Abschnitt 3.2.2.

#### Merkmal 2: Dimensionen

Bei der Durchführung eines Geschäftsprozesses müssen verschiedene *Dimensionen* (vgl. Abschnitt 2.3) berücksichtigt werden:

**Frage 4:** Welche Dimensionen von Geschäftsprozessen werden bei der Modellierung berücksichtigt?

Die Möglichkeit vorausgesetzt, verschiedene Aspekte von Geschäftsprozessen modellieren zu können, stellt sich die Frage, auf welche Weise die Modellierung der verschiedenen Aspekte erfolgt.

**Frage 5:** Werden bei der Abbildung der verschiedenen Aspekte von Geschäftsprozessen Teilmodelle benutzt?

□

Die Ergebnisse der Untersuchung bezüglich der Fragen 4 und 5 finden sich in Tabelle 3.2 in Abschnitt 3.2.2.

**Merkmals 3:** Modellierung quantitativer Information

Aktivitäten verursachen Kosten und verbrauchen Zeit. Diese zwei Faktoren sind die wichtigsten Parameter, die bei der Leistungsbewertung eines Geschäftsprozesses betrachtet werden. Je differenzierter diese Parameter in das Prozeßmodell eingebracht werden können, desto größer ist die Anzahl der berechenbaren Kennzahlen aus einem Geschäftsprozeßmodell.

**Frage 6:** Welche quantitativen Informationen können modelliert werden?

□

Die Ergebnisse der Untersuchung bezüglich Frage 6 finden sich in Tabelle 3.3 in Abschnitt 3.2.2.

**Merkmals 4:** Strukturiert vs. nicht-strukturiert

Wie in Abschnitt 2.1 bereits erläutert, unterscheiden sich Geschäftsprozesse in vielfacher Hinsicht, insbesondere bezüglich ihrer *Strukturiertheit*. Bei den meisten Geschäftsprozessen handelt es sich um Mischformen, d.h., sie enthalten sowohl strukturierte als auch nicht-strukturierte Teile. Die Möglichkeit, nicht-strukturierte Teile von Geschäftsprozessen modellieren zu können und dabei noch ein zur Simulation geeignetes Modell zu erhalten, stellt eine besondere Herausforderung für Modellierungsformalismus und -werkzeug dar.

**Frage 7:** Können nicht-strukturierte Teile von Geschäftsprozessen modelliert werden? Können strukturierte und nicht-strukturierte Teile von Geschäftsprozessen kombiniert werden?

□

Die Ergebnisse der Untersuchung bezüglich Frage 7 finden sich in Tabelle 3.5 in Abschnitt 3.2.2.

**Merkmals 5:** Routing-Konstrukte

Da Geschäftsprozesse nur in seltenen Fällen aus einer einfachen Sequenz von Aufgaben bestehen, werden *Routing-Konstrukte* benötigt, um die Reihenfolge der Aufgaben bei der Bearbeitung von Cases modellieren zu können. Routing-Konstrukte sind somit der wich-

tigste Baustein bei der Modellierung der funktionalen Dimension von Geschäftsprozessen (vgl. Abschnitt 2.3.2).

**Frage 8:** Welche Routing-Konstrukte stehen bei der Modellierung zur Verfügung?

Während des Prozeßverlaufs kann es vorkommen, daß einzelne Aufgaben oder ganze Teilprozesse ein oder mehrmals wiederholt werden, z.B. in Abhängigkeit von sogenannten *Check-Tasks*, bei denen die Qualität der vorangegangenen Bearbeitung überprüft wird. Die Modellierung von Zyklen ist in Verbindung mit simulierbaren Modellen eine weitere Herausforderung für Formalismus und Werkzeug.

**Frage 9:** Lassen sich Zyklen modellieren?

Die Ergebnisse der Untersuchung bezüglich der Fragen 8 und 9 finden sich in Tabelle 3.4 in Abschnitt 3.2.2.

### **Analyse und Simulation der Modelle**

Wie in Abschnitt 2.2 bereits erläutert, sind bei der Beurteilung der Qualität eines Geschäftsprozeß-Entwurfs drei Aspekte von Bedeutung:

- Korrektheit in Bezug auf die Anwendung,
- Konsistenz in Bezug auf (logische) Spezifikationen,
- Effizienz in Bezug auf festgelegte Leistungsparameter.

Diese Qualitätsaspekte lassen sich jeweils mit Hilfe von Verfahren aus den Bereichen *Validierung*, *Verifikation* und *Leistungsbewertung* untersuchen. Ein Werkzeug für den Geschäftsprozeß-Entwurf sollte Verfahren aus allen drei Bereichen unterstützen.

Merkmal 6: Validierung: Simulation/Animation

Im Bereich der *Validierung* konzentrieren wir uns auf die Möglichkeiten zur Simulation und Visualisierung des Verhaltens. Da alle untersuchten Tools eine Animation während der Simulation anbieten, wird im weiteren unter dem Begriff *Simulation* auch die Animation einbezogen. In einer Simulation wird das Verhalten eines Geschäftsprozesses durch Ausführung eines Geschäftsprozeßmodells nachvollzogen. Ausgehend von einem Startzustand werden die einzelnen Aktivitäten gemäß der Spezifikation der logischen Struktur mit Hilfe von Routing-Konstrukten ausgeführt. Es werden folgende Funktionalitäten des Simulators unterschieden:

- Besteht die Möglichkeit, die Simulation Schritt für Schritt, also *benutzergeführt*, auszuführen?
- Kann die Simulation gestartet werden und läuft dann *autonom* ab, bis sie vom Benutzer gestoppt wird oder ihr vordefiniertes Ende erreicht?

- Wird während der Simulation die Bearbeitung eines *einzelnen* Case durchgespielt, oder kann die Bearbeitung *mehrerer* Cases und damit das gesamte Systemverhalten simuliert werden?
- Kann die Simulation abgebrochen werden, falls *nichtendliche Zustandsräume* vorliegen?

**Frage 10:** Welche Funktionalitäten bietet die Simulation/Animation?

□

Die Ergebnisse der Untersuchung bezüglich Frage 10 finden sich in Tabelle 3.5 in Abschnitt 3.2.2.

Merkmal 7: Verifikation

Bei der *Verifikation* geht es um den formalen Nachweis von Eigenschaften. Gegenstand der Untersuchung können dabei sowohl Eigenschaften der *Struktur* des Geschäftsprozeßmodells als auch Eigenschaften dessen *Verhaltens* sein.

**Frage 11:** Welche Möglichkeiten der Verifikation werden bereitgestellt?

□

Die Ergebnisse der Untersuchung bezüglich Frage 11 finden sich in Tabelle 3.6 in Abschnitt 3.2.2.

Merkmal 8: Leistungsbewertung

Bei der *Leistungsbewertung* wird anhand des Geschäftsprozeßmodells die Leistungsfähigkeit des Geschäftsprozesses analysiert. Darunter fallen beispielsweise Möglichkeiten zur Ermittlung der *Durchlaufzeit* oder des *Auslastungsgrads* von Ressourcen.

**Frage 12:** Welche Möglichkeiten zur Leistungsbewertung von Geschäftsprozessen werden bereitgestellt?

□

Die Ergebnisse der Untersuchung bezüglich Frage 12 finden sich in Tabelle 3.6 in Abschnitt 3.2.2.

### 3.2.2 Untersuchung kommerzieller Werkzeuge

Mit dem im vorangegangenen Abschnitt entworfenen Fragenkatalog kann das grundsätzliche Profil von Werkzeugen für den Entwurf von Geschäftsprozessen untersucht werden. In diesem Abschnitt wird dies für eine Auswahl bekannter kommerziell verfügbarer Werkzeuge getan. Anschließend wird das Profil des *VIPtools* als Werkzeug zur Unterstützung des *VIPbusiness-Ansatzes* skizziert.

Auf dem Markt wird eine Vielzahl von Werkzeugen für die Beschreibung von Geschäftsprozessen angeboten. Eine erste Liste mit Werkzeugen wurde durch das Sichten mehrerer

zu diesem Thema im Internet befindlicher Werkzeuglisten erstellt<sup>3</sup>. Diese Liste wurde um Werkzeuge ergänzt, die sich bei der weiteren Literatur- und Internetrecherche als möglicherweise geeignet erwiesen<sup>4</sup>.

In einem ersten Schritt wurde die Menge der genauer zu untersuchenden Werkzeuge durch das Kriterium *Workflow-bezogen* eingegrenzt. Der Wahl dieses Kriteriums liegt die Annahme zugrunde, daß im Kontext von Workflow-Projekten eher Geschäftsprozesse betrachtet werden, die sich durch einen vergleichsweise hohen Grad an Strukturiertheit und damit einfache Formalisierbarkeit und Fähigkeit zur Simulation auszeichnen. Zudem werden diese Geschäftsprozesse häufig genug ausgeführt, so daß eine genaue Analyse insbesondere quantitativer Merkmale lohnend erscheint. Stichproben bei nicht Workflow-bezogenen Werkzeugen etwa aus dem *Groupware*- oder dem „reinen“ *BPR*-Bereich haben gezeigt, daß dort die Schwerpunkte eher bezüglich der Aspekte *Organisation* und *Kommunikation der Beteiligten* gesetzt werden. Dabei steht vor allem die *Modellierung* von Prozessen im Vordergrund, Analysen der entworfenen Modelle bezüglich der in dieser Arbeit relevanten Qualitäts-Merkmale werden so gut wie nicht unterstützt.

Von der Liste entfernt wurden in einem zweiten Schritt Werkzeuge, die ein komplettes *Workflow-Management (WFM)*-System darstellen. Zum einen verursachen diese Werkzeuge bei der Evaluation einen unverhältnismäßig hohen Installationsaufwand, zum anderen liegt der Fokus dieser Werkzeuge nicht in der Modellierung und Analyse von Geschäftsprozessen sondern vielmehr in der Unterstützung ihrer Ausführung. Das Streichen von diesbezüglich ungeeigneten Werkzeugen erfolgte durch den Vergleich der vorläufigen Liste mit [EH95]. Die übrig gebliebenen Werkzeuge (*workflow-bezogen* und *kein* komplettes WFM-System) wurden anhand von Herstellerinformationen (Websites, Informationsmaterial) auf die Unterstützung von *Simulation* bei der Analyse des dynamischen Verhaltens der erstellten Modelle untersucht. Das Ergebnis der Auswahl ist die folgende, 13 Werkzeuge umfassende, Produktliste:

- All Clear
- Aris Toolset
- Bonapart
- Corporate Modeler99
- Extend
- First Step
- Grade
- Income
- Ivy Frame
- Optima

---

<sup>3</sup><http://www.workflowsoftware.com>, <http://www.vtt.fi/tte/staff/ojp/process.htm>, <http://is.twi.tudelft.nl/~hommes>.

<sup>4</sup>Die Liste findet sich in Anhang A.

- Process V3
- Simprocess
- Workflow BPR

Alle Werkzeuge wurden (größtenteils als Testversionen) installiert und evaluiert. Ein kurze Beschreibung der untersuchten Werkzeuge findet sich in Anhang B.

## Modellierungsformalismus

### Ergebnis 1: Grundlegende Eigenschaften des Formalismus

Alle Werkzeuge unterstützen die Modellierung mittels *Drag&Drop*-Techniken und graphischen Symbolen (vgl. Tabelle 3.1). Zwei der Werkzeuge bieten darüber hinaus die Möglichkeit, mittels Skript- oder Programmiersprachen (beispielsweise Informationen zur Entscheidung bei Alternativen während der Simulation) zu modellieren. Während alle Werkzeuge Aktivitäten als Modellbausteine verwenden, erlaubt lediglich *Income* die explizite Modellierung von Zuständen zwischen Aktivitäten.

Produkt	Graphisch	Programmieren	Aktivitäten	Zustände
All Clear	X	X	X	
Aris Toolset	X		X	
Bonapart	X		X	
Corporate Modeler99	X		X	
Extend	X		X	
First Step	X		X	
Grade	X		X	
Income	X		X	X
Ivy Frame	X		X	
Optima	X	X	X	
Process V3	X		X	
Simprocess	X		X	
Workflow BPR	X		X	

Tabelle 3.1: Grundlegende Eigenschaften des Formalismus

### Ergebnis 2: Berücksichtigte Dimensionen

Bei den berücksichtigten Dimensionen von Geschäftsprozessen ergibt sich ein uneinheitliches Bild (vgl. Tabelle 3.2). Alle Werkzeuge bilden die *Ablaufsicht* ab, die *Organisationsicht* findet bei drei Werkzeugen bereits keine Unterstützung mehr. Die Modellierung einer expliziten *Datensicht* ist nur bei vier Werkzeugen möglich.

Angesichts der oft geäußerten Kritik an unübersichtlichen, mit Informationen überladenen Modellen etwas überraschend, wird lediglich bei 6 der 13 Werkzeuge die Trennung der verschiedenen Dimensionen von Geschäftsprozessen in entsprechende Teilmodelle unterstützt, die dann gegebenenfalls, beispielsweise zu Simulationszwecken, über eine *Steuerungssicht* (bei *ARIS*) integriert werden. Bei den übrigen Werkzeugen (so sie denn überhaupt andere Dimensionen als die *Ablaufsicht* unterstützen) müssen alle Informationen in ein Modell integriert werden.

□

Werkzeug	Organisation	Daten	Ablauf	Teilmodelle
All Clear	X		X	
Aris Toolset	X	X	X	X
Bonapart	X	X	X	X
Corporate Modeler99	X		X	
Extend	X		X	
First Step	X		X	X
Grade	X		X	X
Income	X	X	X	X
Ivy Frame	X		X	X
Optima	X		X	
Process V3			X	
Simprocess			X	
Workflow BPR		X	X	

Tabelle 3.2: Berücksichtigte Dimensionen und ihre Integration.

### Ergebnis 3: Modellierbare quantitative Informationen

Alle Werkzeuge unterstützen die Modellierung von quantitativen Informationen (vgl. Tabelle 3.3). Bei sieben der 13 Werkzeuge kann im Bereich *Zeit* allerdings nur eine *Verweilzeit* ( $Vz$ ) der Fälle für *Aktivitäten* ( $A$ ) angegeben werden. Die übrigen Werkzeuge erlauben teilweise eine Differenzierung der *Verweilzeit* in *Bearbeitungszeit* ( $Bz$ ), *Transportzeit* ( $Tz$ ), *Einarbeitungszeit* ( $Ez$ ), *Liegezeit* ( $Lz$ ). *Grade* unterstützt zudem die Festlegung von absoluten *Anfangs-* und *Endzeitpunkten*. Als Einheiten im Bereich *Zeit* werden je nachdem sowohl *Zeiteinheiten* ( $ZE$ ) als auch die üblichen Abkürzungen für *Tage* ( $d$ ), *Stunden* ( $h$ ), *Minuten* ( $m$ ) und *Sekunden* ( $s$ ) verwendet. Im Bereich *Kosten* sind in erster Linie Kosten in Abhängigkeit von den *Zeitdauern* der *Aktivitäten* modellierbar, lediglich fünf Werkzeuge erlauben die Modellierung von *fixen* Kosten.

□

Werkzeug	Zeitgrößen	Kostengrößen
All Clear	<ul style="list-style-type: none"> <li>• <math>Vz</math> (h)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro h</li> </ul>
Aris Toolset	<ul style="list-style-type: none"> <li>• <math>Bz</math> (ZE)</li> <li>• <math>Ez</math> (ZE)</li> <li>• <math>Lz</math> (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Materialkosten</li> <li>• Personalkosten</li> <li>• Sonstige Kosten</li> </ul>
Bonapart	<ul style="list-style-type: none"> <li>• <math>Tz</math> (d:h:m:s)</li> <li>• <math>Bz</math> (d:h:m:s)</li> <li>• <math>Lz</math> (d:h:m:s)</li> </ul>	<ul style="list-style-type: none"> <li>• Fixe Kosten pro A</li> <li>• Kostensatz pro h für A</li> </ul>
Corporate Modeler99	<ul style="list-style-type: none"> <li>• <math>Bz</math> (ZE)</li> <li>• <math>Tz</math> (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Indirekte Kosten pro ZE</li> <li>• Direkte Kosten pro ZE</li> </ul>
Fortsetzung nächste Seite...		



Fortsetzung...		
Werkzeug	Zeitgrößen	Kostengrößen
Extend	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro ZE</li> <li>• Kosten pro Einheit</li> </ul>
First Step	<ul style="list-style-type: none"> <li>• Vz (d:h:m:s)</li> </ul>	<ul style="list-style-type: none"> <li>• Fixe Kosten pro A</li> <li>• Kosten pro ZE</li> </ul>
Grade	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> <li>• Anfangs- und Endzeitpunkt</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro h</li> </ul>
Income	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> <li>• Tz (ZE)</li> <li>• Bz (ZE)</li> <li>• Andere Zeiten (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Verweilkosten pro ZE</li> <li>• Transportkosten pro ZE</li> <li>• Ausführungskosten pro ZE</li> <li>• Andere Kosten pro ZE</li> <li>• Materialkosten pro Einheit</li> <li>• Ressourcenkosten pro ZE</li> </ul>
Ivy Frame	<ul style="list-style-type: none"> <li>• Vz (h:m:s)</li> <li>• Bz (h:m:s)</li> </ul>	
Optima	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Fixe Kosten</li> <li>• Kosten pro h</li> <li>• Kosten pro Gebrauch</li> </ul>
Process V3	<ul style="list-style-type: none"> <li>• Vz (d:h:m:s)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro h</li> <li>• Überstundenkosten pro h</li> <li>• Fixe Kosten pro A</li> </ul>
Simprocess	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro Person</li> <li>• Kosten pro Einheit</li> <li>• Kosten pro Stunde</li> <li>• Fixe Kosten</li> </ul>

Fortsetzung nächste Seite...

Fortsetzung...		
Werkzeug	Zeitgrößen	Kostengrößen
Workflow BPR	<ul style="list-style-type: none"> <li>• Vz (ZE)</li> <li>• Bz (ZE)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten pro ZE</li> <li>• Überstundenkosten pro ZE</li> <li>• Kosten für einmaligen Einsatz</li> <li>• Kosten pro Einheit</li> </ul>

Tabelle 3.3: Modellierung quantitativer Informationen.

#### Ergebnis 4: Strukturiert vs. nicht-strukturiert.

Keines der Werkzeuge unterstützt die Modellierung nicht-strukturierter Prozesse. □

#### Ergebnis 5: Routing-Konstrukte

Alle Werkzeuge stellen Bausteine zur Modellierung von AND-Konstrukten und exklusiven OR<sup>5</sup>-Konstrukten (XOR) zur Verfügung (vgl. Tabelle 3.4). Zwei der Werkzeuge unterstützen zusätzlich die Modellierung von inklusiven OR-Konstrukten (OR). Die Modellierung von Zyklen ist mit 10 Werkzeugen möglich. □

Produkt	XOR	AND	OR	Zyklen
All Clear	X	X		X
Aris Toolset	X	X	X	X
Bonapart	X	X		X
Corporate Modeler99	X	X		X
Extend	X	X		X
First Step	X	X		X
Grade	X	X	X	X
Income	X	X		X
Ivy Frame	X	X		
Optima	X	X		X
Process V3	X	X		
Simprocess	X	X		X
Workflow BPR	X	X		

Tabelle 3.4: Routing-Konstrukte und Zyklen.

#### Ergebnis 6: Simulation

Alle Werkzeuge bieten die Möglichkeit, die Simulation autonom ablaufen zu lassen (vgl. Tabelle 3.5). Die Interaktion zwischen Simulator und Benutzer während der Simulation wird nur von fünf Werkzeugen unterstützt. Bei 11 der Werkzeuge ist es möglich, die Bearbeitung mehrerer Fälle zu simulieren. Schlecht sieht es dagegen bei nichtendlichen Zustandsräumen aus: der Abbruch der Simulation wird für diesen Fall nur von zwei Werkzeugen explizit unterstützt.

<sup>5</sup>Wie bei dem Begriff *Case/Fall* wird in dieser Arbeit für das logische *ODER* das englische *OR* (und für das logische *UND* analog *AND*) verwendet, um Verwechslungen mit der sonstigen Verwendung der deutschen Begriffe zu vermeiden.

□

Werkzeug	Benutzer	Autonom	Gesamt	Nichtendlich
All Clear		X	X	
Aris Toolset <sup>6</sup>				
Bonapart	X	X		X
Corporate Modeler99		X	X	
Extend		X	X	
First Step		X	X	
Grade	X	X	X	
Income		X	X	
Ivy Frame	X	X	X	
Optima		X	X	X
Process V3	X	X	X	
Simprocess		X	X	
Workflow BPR	X	X	X	

Tabelle 3.5: Funktionalität der Simulation/Animation.

### Ergebnis 7: Möglichkeiten zur Verifikation

Keines der Werkzeuge unterstützte die angesprochenen Punkte.

□

### Ergebnis 10: Quantitative Analyse

Im Bereich *Zeit* bieten alle Werkzeuge Möglichkeiten zur Berechnung der *Durchlaufzeit* (vgl. Tabelle 3.6). Große Unterschiede gibt es dabei jedoch bezüglich der Möglichkeiten zur Auswahl verschiedener Bezugsgrößen (beispielsweise pro Case, pro Auftrag) sowie den Möglichkeiten, die Bestandteile der Durchlaufzeit (beispielsweise Transport- oder Rüstzeit) zu betrachten.

Bei der Ermittlung von Kennzahlen aus dem Bereich *Kosten* wird von den meisten Werkzeugen nur die Berechnung variabler Kostengrößen in Abhängigkeit von *Zeit* unterstützt<sup>7</sup>. Auch hier existieren große Unterschiede bezüglich der wählbaren Bezugsgrößen sowie Möglichkeiten zur Berechnung von Teilkosten.

Die meisten Werkzeuge (7) bieten die Möglichkeit, Zeit- und Kostengrößen mit Ressourcen als expliziten Bezugsgrößen zu ermitteln (beispielsweise durch Ressourcen verursachte Kosten). Mit so gut wie allen Werkzeugen lassen sich Aussagen bezüglich des Auslastungsgrads der Ressourcen treffen. Dabei werden in den meisten Fällen Aktivitäten als Warteschlangen modelliert, entsprechende Kenngrößen können mit Hilfe von Methoden aus der Warteschlangentheorie ermittelt werden. Die Effekte von knappen Ressourcen lassen sich ausschließlich indirekt ermitteln, beispielsweise über hohe Wartezeiten bzw. lange Warteschlangen.

□

<sup>7</sup>Die Auswertung konnte aufgrund der zum Teil eingeschränkten Funktionalität der zur Verfügung stehenden Test-Versionen nur anhand von Dokumentationen der Werkzeuge vorgenommen werden.

Werkzeug	Zeitanalyse	Kostenanalyse
All Clear	<ul style="list-style-type: none"> <li>• Dz jedes einzelnen Pfades</li> <li>• Dz (Ges.)</li> <li>• Vz (Ges.) in einer A</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten jedes einzelnen Pfades</li> <li>• Gesamtkosten für Prozeßdurchlauf</li> <li>• Gesamtkosten je A</li> </ul>
Aris Toolset <sup>8</sup>	-	-
Bonapart	<ul style="list-style-type: none"> <li>• Bz (Ges.)</li> <li>• Tz (Ges.)</li> <li>• WF (Ges.)</li> <li>• Dz (Mittl.)</li> <li>• Dz Einzelvorgang</li> <li>• Anteil (%) Bz an Dz</li> </ul>	<ul style="list-style-type: none"> <li>• Gesamtkosten für Prozeßdurchlauf</li> <li>• Gesamtkosten je A</li> <li>• Kostenartenaufspaltung (fix, variabel) je A</li> </ul>
Corporate Modeler99	<ul style="list-style-type: none"> <li>• Dz</li> <li>• Bz in A (Max., Min., Mittl., Ges.)</li> <li>• Prozentuale Auslastung einer A</li> <li>• Warteschlangenlänge pro A (Max., Min., Mittl., Ges.)</li> </ul>	<ul style="list-style-type: none"> <li>• Direkte Kosten (Max., Min., Mittl., Ges.)</li> <li>• Indirekte Kosten (Max., Min., Mittl., Ges.)</li> </ul>
Extend	<ul style="list-style-type: none"> <li>• Zeit in Warteschlange (Max., Mittl.)</li> <li>• Prozentuale Auslastung einer A</li> </ul>	<ul style="list-style-type: none"> <li>• Gesamtkosten je A</li> </ul>
First Step	<ul style="list-style-type: none"> <li>• Dz (Max., Min., Mittl.)</li> <li>• Warteschlangenlänge von Aufträgen vor Ressource</li> <li>• Prozentuale Auslastung der Ressource</li> <li>• Wz (Mittl.)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten (Max., Min., Mittl.)</li> <li>• Variable Kosten</li> <li>• Fixe Kosten</li> </ul>
Fortsetzung nächste Seite...		

<sup>8</sup>Die Testversion stellte keine Möglichkeiten zur Simulation zur Verfügung.

Fortsetzung...		
Werkzeug	Zeitanalyse	Kostenanalyse
Grade	<ul style="list-style-type: none"> <li>• Vz (Max., Min., Mittl., Ges.)</li> <li>• Zeit in Warteschlange(Max., Min., Mittl.)</li> <li>• Zeit zwischen dem Eintreffen an Warteschlange (Max., Min., Mittl.)</li> <li>• Wz an A auf Ressource (Max., Min., Mittl., Ges.)</li> <li>• ungenutzte Zeit pro Ressource(Max., Min., Mittl., Ges.)</li> <li>• Auslastungszeit pro Ressource(Max., Min., Mittl., Ges.)</li> </ul>	<ul style="list-style-type: none"> <li>• Kosten für A (Max., Min., Mittl., Ges.)</li> </ul>
Income	<ul style="list-style-type: none"> <li>• Vz (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Bz (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Tz (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Andere Zeit (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Vz (Ges.) pro A</li> <li>• Bz (Ges.) pro A</li> <li>• Tz (Ges.) pro A</li> <li>• Andere Zeit (Ges.) pro A</li> <li>• Auslastung Ressource</li> <li>• Auslastung Objektspeicher (Zustand)</li> </ul>	<ul style="list-style-type: none"> <li>• Verweilkosten (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Bearbeitungskosten (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Transportkosten (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Andere Kosten (Mittl., Standardabweichung) je Teilprozeß</li> <li>• Verweilkosten (Ges.) pro A</li> <li>• Bearbeitungskosten (Ges.) pro A</li> <li>• Transportkosten (Ges.) pro A</li> <li>• Andere Kosten (Ges.) pro A</li> </ul>
Ivy Frame <sup>9</sup>	-	-
Fortsetzung nächste Seite...		

<sup>9</sup>Das Werkzeug bot keine Möglichkeiten für eine Leistungsbewertung.

Fortsetzung...		
Werkzeug	Zeitanalyse	Kostenanalyse
Optima	<ul style="list-style-type: none"> <li>• gesamte Dz</li> <li>• gesamte Bz</li> <li>• Wz auf bestimmte Ressource</li> <li>• Zeitdauer für blockierte Ressource</li> <li>• Zeitdauer für unbenutzte Ressource</li> <li>• gesamte Wz</li> <li>• mittlere Wz</li> <li>• maximale Wz</li> </ul>	<ul style="list-style-type: none"> <li>• Personalkosten</li> <li>• Einrichtungskosten</li> <li>• Anderweitige Kosten</li> <li>• Fixe A Kosten</li> </ul>
Process V3	<ul style="list-style-type: none"> <li>• Zeit für A ohne Nutzung</li> <li>• Zeit für A unter Auslastung</li> <li>• Wz je A (Max., Mittl.)</li> </ul>	<ul style="list-style-type: none"> <li>• Gesamtkosten für A</li> <li>• Materialkosten für A</li> <li>• Personalkosten für A</li> <li>• Andere Kosten für A</li> <li>• Gesamtkosten für Prozeß</li> <li>• Materialkosten für Prozeß</li> <li>• Personalkosten für Prozeß</li> <li>• Andere Kosten für Prozeß</li> </ul>
Simprocess	<ul style="list-style-type: none"> <li>• Mittlere Dz</li> <li>• Maximale Dz</li> <li>• Mittlere Vz (Max., Mittl.)</li> <li>• Wz auf Ressource (Max., Mittl.)</li> <li>• Wz auf Eintreffen von Bedingungen</li> <li>• prozentuale Auslastung von Ressource</li> </ul>	<ul style="list-style-type: none"> <li>• Prozeßkosten</li> <li>• Kosten für Ressourcennutzung</li> </ul>

Fortsetzung nächste Seite...

Fortsetzung...		
Werkzeug	Zeitanalyse	Kostenanalyse
Workflow BPR	<ul style="list-style-type: none"> <li>• gesamte Dz (Min., Ges.)</li> <li>• Gesamtnutzungszeit jeder Ressource</li> <li>• gesamte Tz</li> <li>• Tz je Pfad</li> <li>• gesamte Vz</li> <li>• Vz je Pfad</li> <li>• gesamte Bz</li> <li>• Bz je Pfad</li> </ul>	<ul style="list-style-type: none"> <li>• Gesamtkosten pro Ressource</li> <li>• Kosten pro Ressource je Pfad</li> </ul>

Tabelle 3.6: Möglichkeiten zur Leistungsbewertung.

### 3.2.3 Profile der Werkzeuge

Ausgehend von den Ergebnissen des vorangegangenen Abschnitts wird nun eine Charakterisierung der untersuchten Werkzeuge vorgenommen. Dazu werden die untersuchten Merkmale zunächst auf einfache Art und Weise operationalisiert, um die Profile der einzelnen Werkzeuge in einem zweiten Schritt in kompakter graphischer Form darstellen zu können. Die Charakterisierung der Werkzeuge ist dabei nicht wertend zu verstehen, es geht an dieser Stelle in erster Linie um die Darstellung der verschiedenen Werkzeug-Profile für den einfachen Vergleich.

Diese Darstellungen der Profile bilden den Ausgangspunkt für die Erläuterung des *VIP-tool*-Profils im Vergleich mit kommerziellen Werkzeugen.

Abbildung 3.5 zeigt das Schema für die Darstellung der Werkzeug-Profile. Als Dimensionen wurden die in Abschnitt 3.2.1 identifizierten Dimensionen des Anforderungsprofils gewählt. Mittels einer Punkte-Skala mit Werten von 0 bis 8 wird ausgedrückt, wie gut (hoher Wert, der Wert „8“ bedeutet *vollständige* Unterstützung aller Punkte) oder wie schlecht (niedriger Wert, der Wert „0“ bedeutet *komplett fehlende* Unterstützung) ein Werkzeug die jeweiligen Anforderungen unterstützt<sup>10</sup>. Die Operationalisierung der qualitativen Merkmale soll an dieser Stelle kurz erläutert werden.

#### Grundlegende Eigenschaften

Bei diesem Merkmal werden je zwei Punkte für die Erfüllung der Anforderungen bezüglich *graphischer Modellierung*, *programmiersprachlichen Ergänzungen*, *explizite Modellierung von Aktivitäten* sowie *explizite Modellierung von Zuständen* vergeben.

<sup>10</sup>Der Wert „8“ als maximaler Wert ergibt sich als kleinster gemeinsamer Nenner der Anzahl der Merkmale der jeweiligen Dimension.

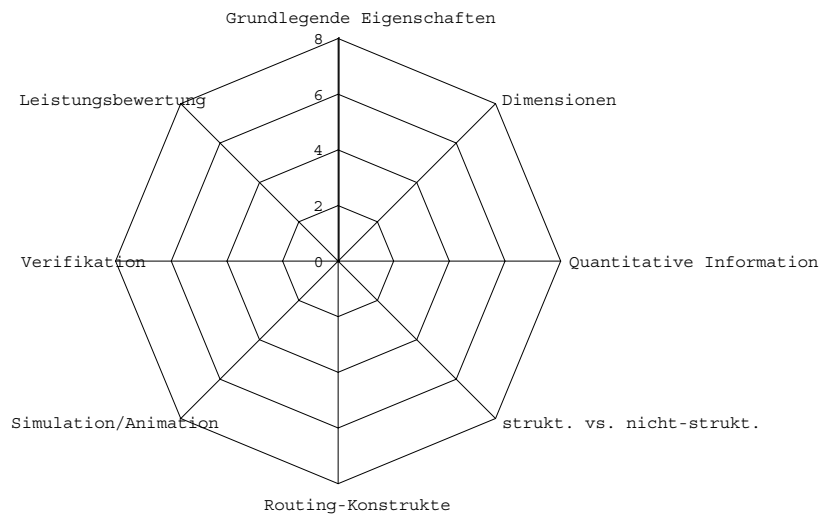


Abbildung 3.5: Schema für die Darstellung der Werkzeug-Profile.

#### Dimensionen

Hier werden je zwei Punkte für die jeweilige Unterstützung der Dimensionen *Organisation/Ressourcen*, *Daten* und *Ablauf* vergeben, zwei weitere Punkte gibt es für die Möglichkeit, die einzelnen Dimensionen in verschiedenen Teil-Modellen berücksichtigen zu können.

#### Modellierung quantitativer Information

Hier interessiert vor allem, welche Merkmale aus den Bereichen *Zeit* und *Kosten* modelliert werden können. Je zwei Punkte gibt es für die Möglichkeit zur Modellierung von *Durchlaufzeit*, den *Bestandteilen* der Durchlaufzeit (beispielsweise Transportzeit), *variablen* sowie *fixen* Kostengrößen.

#### Strukturiert vs. nicht-strukturiert

Hier werden vier Punkte für die Möglichkeit zur Modellierung nicht-strukturierter Segmente von Geschäftsprozessen vergeben. Weitere vier Punkte gibt es für die Möglichkeit, strukturierte und nicht-strukturierte Teile des Geschäftsprozesses miteinander zu kombinieren.

#### Routing-Konstrukte

Es gibt je zwei Punkte für die Möglichkeit, *AND-Konstrukte*, *exklusive* und *inklusive OR-Konstrukte* sowie *Zyklen* realisieren zu können.



## Validierung: Simulation/Animation

Hier gibt es je zwei Punkte für das Vorhandensein folgender Funktionalitäten des Simulators bzw. der Animation:

- *benutzergeführte* Simulation,
- *autonome* Simulation,
- Simulation der Bearbeitung mehrerer Fälle,
- (kontrollierter) Abbruch bei *nichtendlichen Zustandsräumen*.

## Verifikation

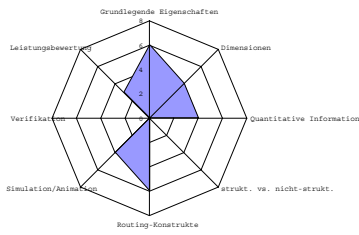
Für den Bereich *Verifikation* werden vier Punkte für die Möglichkeit zur Analyse von Eigenschaften der *Struktur* des Geschäftsprozeßmodells sowie vier Punkte für die Möglichkeit zur Analyse von Eigenschaften des *Verhaltens* vergeben.

## Leistungsbewertung

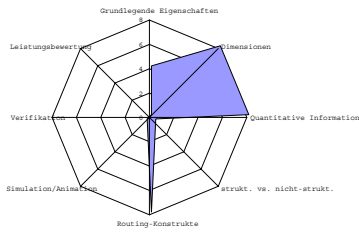
Bei der *Leistungsbewertung* wird je ein Punkt für folgende Analyse-Möglichkeiten vergeben:

- Zeit
  - Ermittlung von Zeitgrößen (grundsätzlich) (ein Punkt),
  - Ermittlung von Teilgrößen der Durchlaufzeit, unterschiedliche Bezugsgrößen möglich (ein Punkt),
- Kosten
  - Ermittlung von Kostengrößen (grundsätzlich) (ein Punkt),
  - Trennung von fixen und variablen Kosten, unterschiedliche Bezugsgrößen möglich (ein Punkt),
- Ressourcen
  - Ermittlung ressourcenbezogener Zeit- und Kostengrößen (ein Punkt),
  - Ermittlung des Auslastungsgrads (ein Punkt),
  - explizite Analyse der Effekte knapper Ressourcen (ein Punkt),
- Sonstige Kenngrößen (ein Punkt).

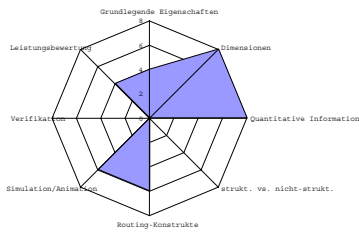
Damit können die Profile der einzelnen Werkzeuge jetzt in übersichtlicher Form dargestellt und kurz beschrieben werden.



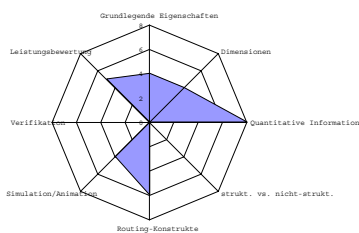
Das Werkzeug *all Clear* der Firma SPSS unterstützt den Entwurf von Geschäftsprozessen mit einem deutlichen Fokus auf Aspekte der Organisation. Die Möglichkeiten zur systematischen Analyse der Modelle sind eher gering.



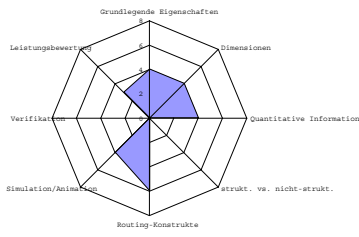
*ARIS* von *IDS Scheer* ist de facto das Standardwerkzeug für den Entwurf von Referenzmodellen im *SAP*-Kontext. Leider verfügte die Test-Version über keinerlei Simulations-Möglichkeiten.



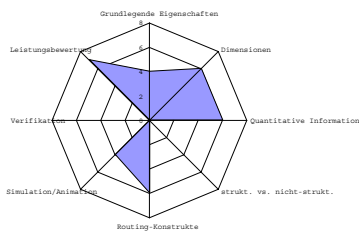
*Bonapart* von Proubis deckt vor allem die Modellierungs-Seite beim Entwurf von Geschäftsprozessen sehr gut ab. Als Schwäche ist die vergleichsweise geringe Unterstützung von Verfahren zur Leistungsbewertung der Entwürfe zu nennen.



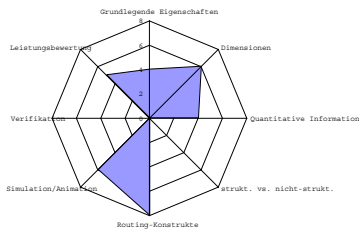
Die Stärken von *Corporate Modeler* (Case-wise) liegen im Bereich der Leistungsbewertung. Alle wichtigen quantitativen Informationen können modelliert werden, die Analyse unterstützt einen Großteil der wichtigen Verfahren.



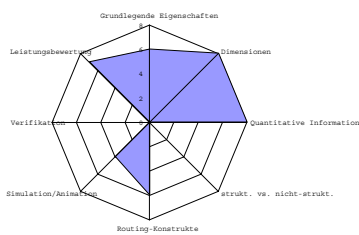
*Extend+BPR* von *Imagine That* unterstützt eine Reihe von Verfahren aus dem Gebiet der Warteschlangen-Theorie. Die Möglichkeiten zur Leistungsbewertung sind allerdings vergleichsweise gering, ebenso die Unterstützung bei der Erstellung der Modelle.



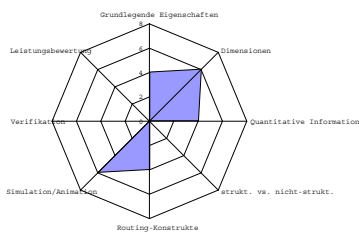
Mit *First Step* von *Interfacing Technologies* können fast alle wichtige Analysen im Bereich Leistungsbewertung durchgeführt werden. Dafür wurde auf einige Dimensionen bei der Modellierung verzichtet, die Möglichkeiten zur Variation bei der Simulation sind auch eher gering.



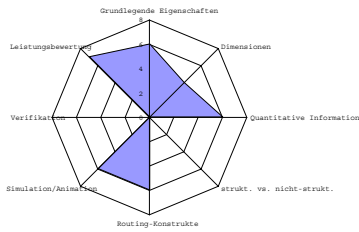
*Grade* von Infologistik bietet eine insgesamt gute Unterstützung bei der Simulation sowie der Leistungsbewertung der Entwürfe. Als Besonderheit sei hier die Möglichkeit zur Modellierung *inklusive OR-Konstrukte* genannt.



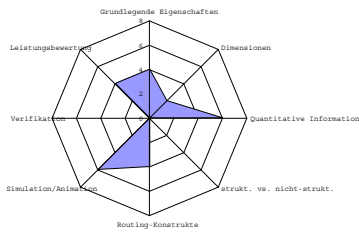
Durch die Verknüpfung mit *Oracle* bietet *Income* von Promatis vielfältige Möglichkeiten auch zur ex-post Analyse der bei Simulationen generierten Daten. Besondere Schwächen weist das Werkzeug lediglich in den Bereichen *Verifikation* und *nicht-strukturierte Segmente* auf.



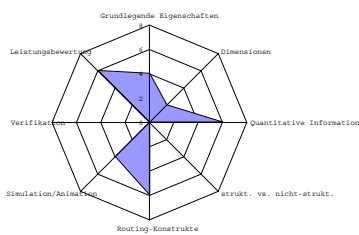
*Ivy Frame* von Ivy Team bietet keine Möglichkeiten für eine systematische Leistungsbewertung der entworfenen Geschäftsprozesse. Es können lediglich Werte während der Simulation verfolgt werden, eine Auswertung wird nicht unterstützt.



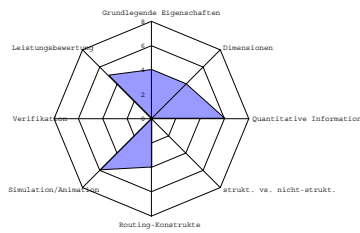
*Optima* von Micrografx hat seine Stärken im Bereich der Leistungsbewertung. Vor allem Analysen bezüglich Ressourcen werden sehr vielfältig unterstützt. Schwächen liegen auch hier in den Bereichen *Verifikation* und *nicht-strukturierte Segmente*.



Bei *Scitor* von der gleichnamigen Firma wird ausschließlich die Modellierung der *Ablauf*-Dimension unterstützt, alle übrigen Informationen können nur implizit modelliert werden. Der großen Zahl an Möglichkeiten zur Modellierung quantitativer Informationen steht eine vergleichsweise geringe Zahl an Analyse-Verfahren gegenüber.



*Simprocess* von CACI Products bietet vor allem im Bereich der Leistungsbewertung sehr gute Möglichkeiten. Bei der Modellierung wird dagegen ausschließlich die *Ablauf*-Dimension unterstützt. Interessant ist die Möglichkeit, die gleichzeitige Simulation mehrerer Modelle über *Synchronisationsbausteine* zu steuern.



*Workflow BPR* von Holosofx kann als Repräsentant des Durchschnittes dieser Untersuchung angesehen werden. Alle relevanten Punkte (bis auf die Bereiche *Verifikation* und *nicht-strukturierte Segmente*) werden in ausreichender bis guter Weise unterstützt. Eine interessante Analyse-Möglichkeit ergibt sich durch die Möglichkeit, für einen Geschäftsprozeß mehrere Entwürfe vergleichend untersuchen zu können.

Auf Grundlage der bisherigen Betrachtungen lassen sich die folgenden Punkte feststellen, die von den untersuchten Werkzeugen nur unzureichend unterstützt werden:

- Nicht-strukturierte Segmente von Geschäftsprozessen können nicht modelliert werden.

Da die wenigsten Geschäftsprozesse vollständig strukturiert sind (siehe auch Abschnitt 3.1.1), verursacht die Nicht-Unterstützung dieser Anforderung unnötige Probleme beim Entwurf vieler Geschäftsprozesse.

- Analyseverfahren aus dem Bereich *Verifikation* werden nicht unterstützt.

Offenbar spielt die *Konsistenz* von Geschäftsprozeß-Entwürfen in der (Modellierungs-)Praxis noch keine Rolle. Mit der fortschreitenden Verknüpfung von Geschäftsprozessen und Informationstechnologie wird es jedoch zunehmend wichtiger, für die der Verknüpfung zugrunde gelegten Geschäftsprozeß-Entwürfe die Erfüllung gewisser logischer Anforderungen gewährleisten zu können.

- Das Verhalten des Geschäftsprozeß-Entwurfs wird nicht explizit dargestellt.

Zwar läßt sich das Verhalten bei einigen Werkzeugen während der Ausführung der Geschäftsprozeßmodelle mit Hilfe von Animations-Techniken beobachten, eine umfassende Betrachtung des im Rahmen der Simulation generierten Verhaltens ist jedoch nicht vorgesehen. Insbesondere werden lediglich einzelne mögliche Ausführungspfade betrachtet. Effekte von Auswahlentscheidungen, Zyklen oder Parallelität bei der Ausführung der Geschäftsprozeßmodelle lassen sich auf diese Weise nur sehr schwer erkennen.

- Es ist keine systematische Analyse der Effekte unterschiedlicher Werte für Zeit und Kosten in den Geschäftsprozeßmodellen möglich.

Da für unterschiedliche Zeit- und Kostenwerte unterschiedliche Simulationen vorgenommen werden müssen, greifen an dieser Stelle die in Abschnitt 3.1.2 erläuterten Nachteile bezüglich Vergleichbarkeit und Effizienz.

- Nur wenige Werkzeuge erlauben die explizite Modellierung von Zuständen.

Damit sind einige Analyseverfahren nicht möglich (vgl. Abschnitt 3.1.1). Durch die Auswirkungen insbesondere auf die Möglichkeiten für Zeitkonzepte bei der Modellierung wird die Vergleichbarkeit von Simulationen für verschiedene Zeitwerte erschwert.

Als *Fazit* ergibt sich: Entscheidende Punkte des in dieser Arbeit identifizierten Anforderungsprofils für Methoden und Werkzeuge zum Geschäftsprozeß-Entwurf werden von kommerziellen Werkzeugen bisher nur unzureichend unterstützt.

### 3.2.4 Profil des *VIPtools*

Ausgehend von den im vorangegangenen Abschnitt identifizierten Profilen kommerzieller Werkzeuge wird nun untersucht, inwiefern das *VIPtool* den formulierten Anforderungen an ein Werkzeug für den Geschäftsprozeß-Entwurf gerecht wird. Zur Erläuterung des Profils werden wieder die bereits eingeführten Merkmale aus den Bereichen *Modellierung* sowie *Simulation und Analyse* verwendet.

#### Grundlegende Eigenschaften

Petrinetze als gewählter Formalismus erlauben die graphische Modellierung der Geschäftsprozeß-Entwürfe (*graphische Modellierung*). Falls notwendig, kann beispielsweise das Routing von Cases mittels prädikatenlogischen Ausdrücken gesteuert werden (*programmiersprachliche Ergänzungen*). Sowohl *Aktivitäten* als auch *Zustände* können *explizit* modelliert werden. Damit erfüllt das *VIPtool* alle Anforderungen in diesem Bereich.

#### Dimensionen

Einen Kompromiß stellt die Wahl der unterstützten Dimensionen dar. Aus Gründen des Implementierungs-Aufwands wurde darauf verzichtet, für jede Dimension einen eigenen Editor für den Teilmodell-Entwurf zur Verfügung zu stellen. Da die Berücksichtigung aller Dimensionen zu unübersichtlichen Modellen geführt hätte, wurde entschieden, nur die für die geplanten Analyseverfahren unbedingt benötigten Dimensionen im Modell zu berücksichtigen. Daher können nur Aspekte der *Ablauf-* sowie der *Organisations-* beziehungsweise *Ressourcen-*Dimension explizit modelliert werden. Auf Möglichkeiten zur expliziten Modellierung der *Daten-*Dimension wurde verzichtet<sup>11</sup>. Routing-relevante Information (vgl. [Aal98d]) kann im Modell jedoch problemlos über Marken modelliert werden.

#### Modellierung quantitativer Information

Alle relevanten quantitativen Informationen können problemlos modelliert werden. Details dazu finden sich in Kapitel 5.

---

<sup>11</sup>Die Integration der *Daten-*Dimension in petrinetzbasierte Geschäftsprozeßmodelle wird beispielsweise in [Obe96] und [Wei00] beschrieben.

## Strukturiert vs. nicht-strukturiert

Das *VIPtool* ist so konzipiert, daß alle in Abschnitt 3.1.1 formulierten Anforderungen an die Unterstützung des Entwurfs von Geschäftsprozessen mit nicht-strukturierten Segmenten erfüllt werden. Einzelheiten zur Unterstützung dieser Anforderungen durch den in dieser Arbeit verwendeten (und damit dem *VIPtool* zugrundeliegenden) Modellierungs-Formalismus finden sich in den Abschnitten 5.6 und 6.3.6.

## Routing-Konstrukte

Bis auf *inklusive* OR-Konstrukte werden alle relevanten Routing-Konstrukte unterstützt. In Abschnitt 5.2 werden alle diesbezüglichen Details erläutert.

## Validierung: Simulation/Animation

Das *VIPtool* bietet sowohl Möglichkeiten zur *benutzergeführten* als auch zur *autonomen* Simulation. Nichtendliche Zustandsräume können mittels Techniken aus dem Bereich der *halbordnungsbasierten* Simulation in effektiver und effizienter Weise gehandhabt werden (vgl. dazu die Ausführungen in Kapitel 6). Über die verwendeten Markierungen kann auf einfache Weise die Anzahl der bei der Simulation zu bearbeitenden Fälle variiert werden. Es zudem möglich, die Simulation in animierter Form durchzuführen.

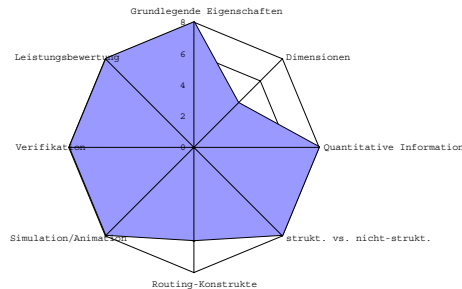


Abbildung 3.6: Profil des *VIPtools*.

## Verifikation

Für den verwendeten Modellierungs-Formalismus existiert eine Vielzahl an Analyseverfahren aus dem Bereich *Verifikation*. Dabei können sowohl Eigenschaften des Geschäftsprozeßmodells selbst (*Strukturanalyse*) als auch Eigenschaften des möglichen Verhaltens untersucht werden (vgl. Abschnitt 7.1).

## Leistungsbewertung

Bei der Leistungsbewertung der Geschäftsprozeß-Entwürfe können alle relevanten Merkmale systematisch untersucht werden. Die Vorteile des *VIPbusiness*-Ansatzes bezüglich *Effizienz* und *Vergleichbarkeit* (vgl. Abschnitt 3.1.4) werden durch den Einsatz des *VIPtools* voll unterstützt. Als zusätzliche Besonderheit sei an dieser Stelle die Visualisierung der Effekte des Einsatzes knapper Ressourcen genannt.

Abbildung 3.6 zeigt das Profil des *VIPtools* in der graphischen Darstellung. Dieses Profil bildet die Grundlage für die Weiterentwicklung des *VIPtools* zu einem Werkzeug für den Entwurf von Geschäftsprozessen. Einzelheiten zur (implementierungs-)technischen Realisierung dieses Profils sowie zum Einsatz des *VIPtools* zur Unterstützung des *VIPbusiness*-Ansatzes finden sich in Kapitel 8. Doch zunächst werden in den nächsten Kapiteln die drei Schritte des *VIPbusiness*-Ansatzes im Detail eingeführt. Den Anfang machen die Grundlagen der Modellierung von Geschäftsprozessen im nächsten Kapitel.



# Kapitel 4

## Modellierung: Grundlagen

Wie in Abschnitt 3.1 bereits kurz erläutert, werden in dieser Arbeit zur Erstellung von Geschäftsprozeßmodellen Bausteine benutzt, die auf Konzepten aus der Petrinetz-Theorie basieren. Bevor in Kapitel 5 auf die eigentlichen Modellierungs-Bausteine eingegangen wird, werden in diesem Kapitel zunächst einige grundlegende Konzepte aus der Petrinetz-Theorie eingeführt.

Seit den ersten Arbeiten von Carl Adam Petri aus dem Jahr 1962 [Pet62] sind in verschiedenen Anwendungsbereichen unterschiedliche Formen (auch: *Klassen*) von Petrinetzen entstanden, die sich in ihrer Modellierungsmächtigkeit stark unterscheiden. Das Spektrum reicht dabei von *Kanal/Instanzen-Netzen* ohne dynamisches Verhalten über Klassen *einfacher Petrinetze* wie *Bedingungs/Ereignis-Netze* oder *Stellen/Transitionen-Netze* bis hin zu Varianten *höherer Petrinetze* wie *Coloured Petri Nets* oder *Prädikate/Transitionen-Netze*. Übersichten der verschiedenen Netzklassen finden sich beispielsweise in [RR98a], [BRR87], [SR93] oder [DA94].

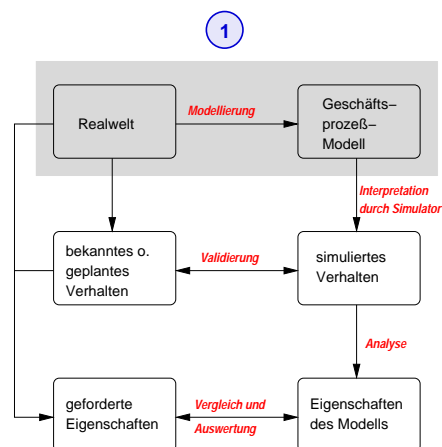


Abbildung 4.1: Schritt 1 (I): Grundlagen.

Die in dieser Arbeit zur Erstellung von Geschäftsprozeßmodellen verwendeten *VIPbusiness-Netze* basieren auf der Klasse der im Rahmen des *VIP-Projekts* entwickelten *VIPmodel-Netze*, die ihrerseits eine Erweiterung der bereits erwähnten High-Level-Varianten *Coloured Petri Nets* und *Prädikate/Transitionen-Netze* darstellen.

In Abschnitt 4.1 werden anhand einfacher Petrinetze einige wichtige Eigenschaften vorgestellt, die prinzipiell alle Klassen von Petrinetzen besitzen. Da die durch einfache Petrinetze zur Verfügung gestellten Bausteine für die Modellierung von Geschäftsprozessen nicht ausreichen, werden in Abschnitt 4.2 *höhere Petrinetze* vorgestellt, bevor in Abschnitt 4.3 kurz auf die Eigenschaften von *VIPmodel-Netzen* eingegangen wird. Die in diesem Kapitel verwendeten Formalismen basieren im wesentlichen auf [Bau96, DE95, Rei86] (Abschnitt 4.1) sowie [Des00, Obe96] (Abschnitt 4.2) und [Fre01] (Abschnitt 4.3).

## 4.1 Einfache Petrinetze

Unabhängig von der gewählten Netzklasse liegt allen Petrinetz-Varianten als gemeinsame Struktur ein gerichteter, bipartiter Graph zugrunde, der als *Netz* (auch: *Netzgraph*) bezeichnet wird:

### Definition 1 (Netz)

Ein *Netz*  $N = (S, T, F)$  besteht aus einer Menge  $S$  von *Stellen* (dargestellt durch Kreise), einer mit  $S$  disjunkten Menge  $T$  von *Transitionen* (Quadrate) und einer *Flußrelation* (Pfeile)  $F \subseteq (S \cup T) \times (S \cup T)$ , die  $F \cap (S \times S) = F \cap (T \times T) = \emptyset$  erfüllt.

*Knoten* von  $N$  sind alle Stellen und alle Transitionen. □

Zur Beschreibung der kausalen Beziehungen zwischen den Aktivitäten eines Geschäftsprozesses ist es notwendig, bezüglich der Flußrelation  $F$  von Reihenfolgen reden zu können. Dazu benötigen wir die Begriffe *Vorbereich* und *Nachbereich*:

### Definition 2 (Vorbereich, Nachbereich)

Für  $x$  aus  $N$  bezeichnet  $\bullet x = \{y \mid (y, x) \in F\}$  den *Vorbereich* von  $x$  und  $x \bullet = \{y \mid (x, y) \in F\}$  den *Nachbereich* von  $x$ . □

Um den Weg zwischen zwei Knoten eines Netzes beschreiben zu können, führen wir den Begriff des *gerichteten Pfads* ein:

### Definition 3 (gerichteter Pfad)

Für ein Netz  $N = (S, T, F)$  ist die Sequenz von Knoten  $(n_0, n_1, \dots, n_k)$  mit  $k \in \mathbb{N}$  ein *gerichteter Pfad* von  $n_0$  nach  $n_k$ , falls für alle  $i$  mit  $0 \leq i < k$  gilt:  $(n_i, n_{i+1}) \in F$ . □

Da alle Aktivitäten eines Geschäftsprozesse in irgendeinem kausalen Zusammenhang zueinander stehen sollten, erstellen wir im Rahmen dieser Arbeit ausschließlich *zusammenhängende* oder *stark zusammenhängende* Netze:

### Definition 4 (zusammenhängend, stark zusammenhängend)

Ein Netz  $N = (S, T, F)$  ist *zusammenhängend*, wenn keine Zerlegung  $S \cup T = X_1 \cup X_2$  mit  $X_1, X_2 \neq \emptyset$ ,  $X_1 \cap X_2 = \emptyset$ ,  $F \subseteq (X_1 \times X_1) \cup (X_2 \times X_2)$  existiert.

$N$  heißt *stark zusammenhängend*, wenn für je zwei Elemente  $x, y \in S \cup T$  mit  $x \neq y$  eine Sequenz  $(z_1, z_2), (z_2, z_3), \dots, (z_{n-1}, z_n) \in F$  existiert ( $n \geq 2$ ), so daß  $x = z_1$  und  $y = z_n$ . □

Für die spätere Konstruktion von Routing-Konstrukten benötigen wir noch die Begriffe *vorwärtsverzweigter* und *rückwärtsverzweigter* Knoten, um beispielsweise Auswahlentscheidungen bei der Bearbeitung von Cases modellieren zu können:

### Definition 5 (vorwärtsverzweigt, rückwärtsverzweigt)

Sei  $N = (S, T, F)$  ein Netz. Für jedes  $x \in (S \cup T)$  gilt:  $x$  heißt

- *vorwärtsverzweigt*, falls  $|x \bullet| > 1$ ,
- *rückwärtsverzweigt*, falls  $|\bullet x| > 1$ .

□

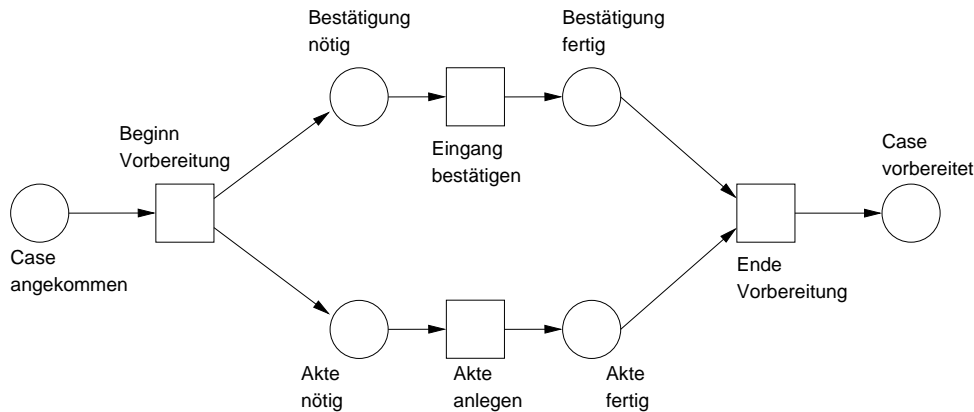


Abbildung 4.2: Ein erster Entwurf des Geschäftsprozeßmodells als einfaches Netz.

### Beispiel

Um die in dieser Arbeit eingeführten Bausteine und Konzepte zu illustrieren, benutzen wir eine vereinfachte Version eines Geschäftsprozesses zur Bearbeitung von Schadensfällen in einer Versicherung. Mit der Einführung neuer Konzepte wird dieses Beispiel sukzessive erweitert. Die Version des Beispiels am Ende von Kapitel 5 enthält schließlich alle zur Modellierung von Geschäftsprozessen benötigten Bausteine.

Abbildung 4.2 zeigt einen ersten Entwurf eines Teils dieses Geschäftsprozesses als einfaches Netz. Sobald eine Schadensmeldung bei der Versicherung eingeht (**Case angekommen**), wird mit der Vorbereitung der Bearbeitung begonnen (**Beginn Vorbereitung**). Anschließend wird (unabhängig voneinander, dazu später mehr) der Eingang der Schadensmeldung bestätigt (**Eingang bestätigen**) sowie eine Akte für die Bearbeitung des Cases angelegt (**Akte anlegen**). Nach Erledigung dieser beiden Aktivitäten wird die Vorbereitung abgeschlossen (**Ende Vorbereitung**).

Das Netz in Abbildung 4.2 enthält eine Menge von *Stellen*  $\{\text{Case angekommen, Bestätigung nötig, Bestätigung fertig, Akte nötig, Akte fertig, Case vorbereitet}\}$ , eine Menge von *Transitionen*  $\{\text{Beginn Vorbereitung, Eingang bestätigen, Akte anlegen, Ende Vorbereitung}\}$  und die Flußrelation, die in Form von gerichteten Kanten eingezeichnet ist. Zu der Transition **Beginn Vorbereitung** besteht der *Vorbereich* aus der Stelle **Case angekommen**, der *Nachbereich* aus den Stellen **Bestätigung nötig** und **Akte nötig**.

Die Transition **Beginn Vorbereitung** ist ein *vorwärtsverzweigter*, die Transition **Ende Vorbereitung** ein *rückwärtsverzweigter* Knoten. □

Die Dynamik eines Systems äußert sich in Änderungen des *Systemzustands*. Bevor Änderungen des Systemzustands beschrieben werden können, muß erst der Systemzustand selbst dargestellt werden. Bei Petrinetzen geschieht dies durch *Markierungen*:

**Definition 6 (Markierung)**

Sei  $N = (S, T, F)$  ein Netz. Eine *Markierung* von  $N$  ist eine Abbildung  $m : S \rightarrow \mathbb{N}$ . Eine Stelle  $s$  heißt *markiert* von  $m$ , wenn  $m(s) > 0$ . Eine Menge von Stellen heißt *markiert*, wenn wenigstens eines ihrer Elemente markiert ist. □

Eine *Änderung* des Systemzustands wird durch das *Schalten* einer Transition bewirkt und mittels einer *Schaltregel* beschrieben. Bevor ein solches Schalten stattfinden kann, muß für die entsprechende Transition eine *Aktivierungsbedingung* erfüllt sein:

**Definition 7 (aktiviert, Schaltregel)**

Eine Transition  $\tau$  ist *aktiviert* von einer Markierung  $m$ , wenn  $m(s) \geq 1$  für jede Stelle  $s$  in  $\bullet\tau$ . Falls die Transition  $\tau$  aktiviert ist, kann sie schalten und überführt  $m$  in die Markierung  $m'$ , die durch folgende *Schaltregel* definiert ist:

$$m'(s) = \begin{cases} m(s) - 1 & \text{falls } s \in \bullet\tau \setminus \tau\bullet \\ m(s) + 1 & \text{falls } s \in \tau\bullet \setminus \bullet\tau \\ m(s) & \text{sonst.} \end{cases}$$

$m'$  wird auch als (unmittelbare) *Folgemarkierung* von  $m$  bezüglich  $\tau$  bezeichnet. Der Schaltvorgang wird in folgender Schreibweise dargestellt:  $m \xrightarrow{\tau} m'$ . Die Menge aller von  $m$  aus erreichbaren Folgemarkierungen wird mit  $[m\rangle$  bezeichnet. □

Anschaulich besagt die Schaltregel, daß die Transition  $\tau$  beim Schalten je eine Marke von jeder Stelle ihres Vorbereichs konsumiert und je eine Marke auf jeder Stelle ihres Nachbereichs erzeugt.

Ein Netz mit einer Markierung wird auch als *markiertes Petrinetz* bezeichnet<sup>1</sup>:

**Definition 8 (Markiertes Petrinetz)**

Ein *markiertes Petrinetz*  $(N, m_0)$  besteht aus einem Netz  $N = (S, T, F)$  und einer *Anfangsmarkierung*  $m_0 : S \rightarrow \mathbb{N}$ . Graphisch wird die Anfangsmarkierung durch eine entsprechende Anzahl von Marken pro Stelle (in Form schwarzer Punkte) dargestellt. □

---

<sup>1</sup>In der Literatur findet sich auch häufig der Begriff *System* (vgl. beispielsweise [DE95]).

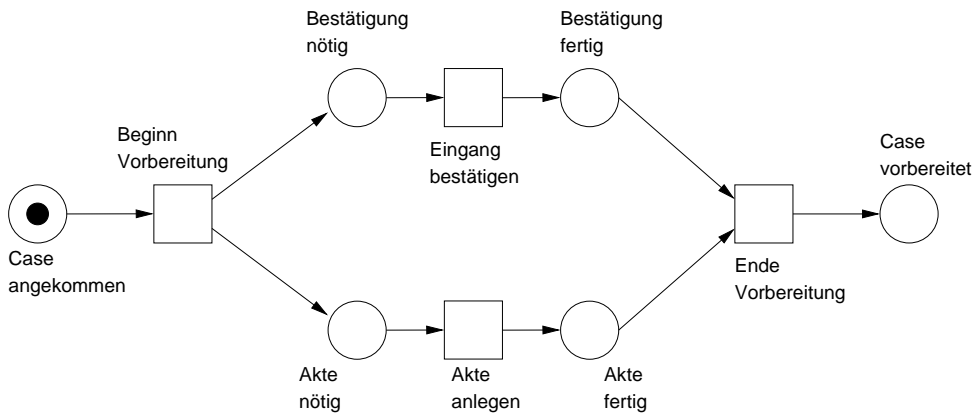


Abbildung 4.3: Markierung.

### Beispiel(Fortsetzung)

In Abbildung 4.3 ist das Geschäftsprozessmodell aus Abbildung 4.2 um eine Markierung ergänzt worden.

Mit dieser Markierung ist die Transition **Beginn Vorbereitung** aktiviert und kann schalten. Der Schaltvorgang entspricht der Ausführung der entsprechenden Aktivität des Geschäftsprozesses. Der Schaltregel entsprechend wird dabei die ursprüngliche Markierung in die Folgemarkierung überführt (vgl. Abbildung 4.4).

□

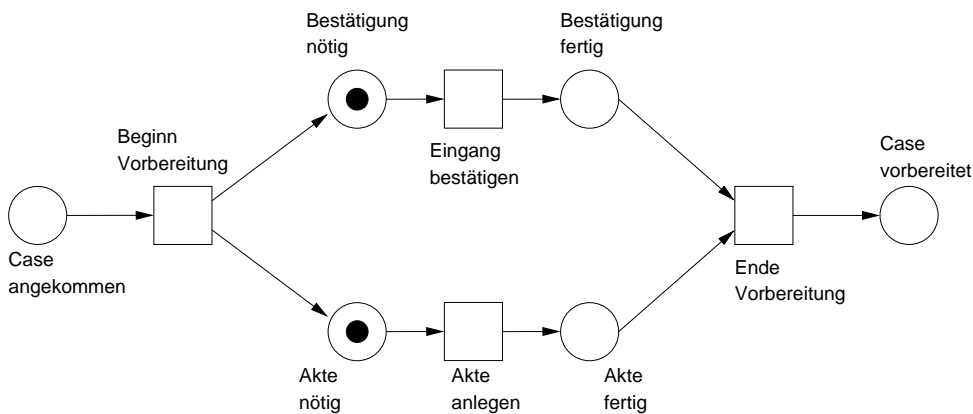


Abbildung 4.4: Folgemarkierung.

Bisher sind Marken als nicht unterscheidbare Objekte behandelt worden, dargestellt durch schwarze Punkte. Beim Entwurf von Geschäftsprozessen im Rahmen des *VIPbusiness*-Ansatzes ergeben sich jedoch einige Anforderungen, die von dieser Art von Marken nicht erfüllt werden.

- Cases müssen unterscheidbar sein.

Soll mit Hilfe des Geschäftsprozeßmodells die Bearbeitung mehrerer Cases untersucht werden, ist es notwendig, den Stand der Bearbeitung für jeden Case einzeln betrachten zu können. Dies ist nur möglich, wenn die Marken unterschieden werden können, beispielsweise durch Verwendung einer Identifikationsnummer.

- Bei der Bearbeitung von Cases müssen Attribute berücksichtigt werden können.

Neben der Verwendung von Merkmalen zur bloßen Identifikation sind häufig weitere Attribute der zu bearbeitenden Cases von Interesse:

- Routing-relevante Informationen

Sind bei der Bearbeitung von Cases Auswahlentscheidungen bezüglich der Ausführung von Aktivitäten zu treffen (beispielsweise ist zu entscheiden, ob eine Rechnung bezahlt wird oder nicht), werden diese Auswahlentscheidungen für unterschiedliche Cases in der Regel auch unterschiedlich ausfallen. Häufig hängt die Auswahlentscheidung dabei direkt von Attributen des Case ab (z.B. der Höhe des gemeldeten Schadens). Um überprüfen zu können, ob sich ein Geschäftsprozeß-Entwurf zur Handhabung unterschiedlicher Cases eignet, ist es notwendig, die entsprechenden, für das Routing dieser Cases relevanten Attribute modellieren zu können (vgl. [Aal98d]<sup>2</sup>). Schwarze Punkte als Marken reichen dazu nicht mehr aus.

- Abhängige Zeit- und Kostenbewertungen

Für die Leistungsbewertung eines Geschäftsprozeß-Entwurfs kann es interessant sein, Zeit- und Kostenbewertungen in Abhängigkeit der Attribute eines Case und somit in Abhängigkeit von Marken zu gestalten. So könnte beispielsweise modelliert werden, daß die Bearbeitung einer Schadensmeldung in Abhängigkeit von der Höhe des gemeldeten Schadens unterschiedliche Kosten verursacht.

- Darstellung des Informations- und Datenflusses

Soll der Weg, den Marken in Petrinetzen zurücklegen, als Bearbeitung konkreter Cases durch Ausführung eines Geschäftsprozeßmodells im Kontext eines betrieblichen Informationssystems interpretiert werden, sind vor allem die bei der Bearbeitung entstehenden Informations- bzw. Datenflüsse und ihre Bewältigung durch das Informationssystem von Interesse. Entsprechend müssen hier Marken verwendet werden, die sich zur Darstellung komplexer Informations-Objekte eignen. Ein bekanntes Beispiel ist der bereits in Abschnitt 2.3.2 beschriebene *INCOME*-Ansatz [Obe96] mit seinen Erweiterungen. In den dort verwendeten Modellen werden mit Hilfe spezieller Marken beispielsweise Operationen auf Datenbank-Tabellen oder strukturierten Dokumenten [Wei00] realisiert.

Der Übergang von Marken als nicht unterscheidbare Objekte hin zu Marken als unterscheidbare Objekte mit beliebigen Attributen bedeutet zugleich den Übergang von der Klasse der *einfachen* Petrinetze (auch: *Low-Level-Petrinetze*) zu den *höheren* Petrinetzen (auch: *High-Level-Petrinetze*). Die zusätzliche Modellierungsmächtigkeit dieser Art von Petrinetzen ist leider mit einem höheren Formalisierungsaufwand insbesondere für die Schaltregel verbunden. In der Literatur existieren eine ganze Reihe von Ansätzen

---

<sup>2</sup>Dort werden diese Attribute werden jedoch nicht explizit modelliert. Eine Simulation der Geschäftsprozeßmodelle für unterschiedliche Cases ist damit nicht möglich (vgl. Abschnitt 5.2).

zur Beschreibung verschiedener Varianten höherer Petrinetze, beispielsweise *allgemeine Prädikate/Transitionsnetze* [GL81, Gen87, Smi98], *gefärbte Petrinetze* (auch: *Colored Petri Nets* [Jen96, Jen95]) oder *Workflow Nets* [Aal98d]. Im folgenden Abschnitt wird eine Klasse höherer Petrinetze beschrieben, die im wesentlichen auf den Ausführungen in [Des00] basiert. Diese Netzklasse bildet die Grundlage für die Klasse der *VIPmodel-Netze*, der Netzklasse, die dem *VIP-Projekt* und insbesondere der Funktionalität des *VIPtools* zugrundeliegt. Bezüglich ihrer Modellierungsmächtigkeit stellen *VIPmodel-Netze* eine Einschränkung der im folgenden beschriebenen höheren Petrinetze dar. Welche Einschränkungen dies sind, wird in Abschnitt 4.3 beschrieben.

## 4.2 Höhere Petrinetze

In höheren Petrinetzen tragen Stellen unterscheidbare Marken verschiedener Sorten (auch: *gefärbte Marken*). Da dabei auch Marken derselben Sorte in mehreren Exemplaren auftreten können, benötigen wir zur Beschreibung höherer Petrinetze zunächst den Begriff der *Multimenge*:

### Definition 9 (Multimenge)

Sei  $\mathcal{D}$  eine Menge. Eine Multimenge  $\mu : \mathcal{D} \rightarrow \mathbb{N}_0$  ist eine Abbildung, die jedem Element  $d \in \mathcal{D}$  eine natürliche Zahl  $n \in \mathbb{N}_0$  zuordnet und damit dessen Multiplizität beschreibt. Multimengen sind insofern eine Verallgemeinerung von Mengen, als daß sie mehrere Kopien eines Elementes enthalten können.

Eine Multimenge  $\mu$  heißt *endlich*, wenn nur endlich vielen  $d \in \mathcal{D}$  eine Zahl  $n > 0$  zugeordnet wird. Für den Definitionsbereich  $\mathcal{D}$  einer Multimenge  $\mu$  wird auch die Schreibweise  $dom(\mu)$  verwendet. Für die Menge aller Multimengen über  $\mathcal{D}$  schreiben wir  $\mathcal{M}(\mathcal{D})$ .

Vergleich, Summe und Differenz zweier Multimengen  $\mu_1$  und  $\mu_2$  sind wie folgt elementweise definiert:

$$\begin{aligned} \mu_1 = \mu_2 &: dom(\mu_1) = dom(\mu_2) \wedge \forall x \in dom(\mu_1) : \mu_1(x) = \mu_2(x) \\ \mu_1 \geq \mu_2 &: dom(\mu_1) \supseteq dom(\mu_2) \wedge \forall x \in dom(\mu_2) : \mu_1(x) \geq \mu_2(x) \\ \mu_1 + \mu_2 &: dom(\mu_1 + \mu_2) = dom(\mu_1) \cup dom(\mu_2) \wedge \forall x \in dom(\mu_1 + \mu_2) : \\ &(\mu_1 + \mu_2)(x) = \begin{cases} \mu_1(x) + \mu_2(x) & \text{falls } x \in (dom(\mu_1) \cap dom(\mu_2)) \\ \mu_1(x) & \text{falls } x \notin dom(\mu_2) \\ \mu_2(x) & \text{falls } x \notin dom(\mu_1) \end{cases} \\ \mu_1 - \mu_2 &: dom(\mu_1 - \mu_2) = dom(\mu_1) \wedge \forall x \in dom(\mu_1 - \mu_2) : \\ &(\mu_1 - \mu_2)(x) = \begin{cases} \mu_1(x) - \mu_2(x) & \text{falls } x \in dom(\mu_2) \\ \mu_1(x) & \text{sonst} \end{cases} \end{aligned}$$

wobei die Differenz nur für  $\mu_1 \geq \mu_2$  definiert ist.

□

Damit können jetzt *Prädikate/Transitionen-Netze* eingeführt werden:

**Definition 10 (Prädikate/Transitionen-Netz)**

Ein *Prädikate/Transitionen-Netz* ist ein Tupel  $(S, T, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  mit

- $(S, T, F)$  ist ein Netz.
- $\mathcal{D}$  ist eine *Menge farbiger Marken*, wobei für jede Stelle  $s \in S$  eine *Domäne*  $\mathcal{D}_s \subseteq \mathcal{D}$  definiert ist. Domänen sind verallgemeinerte kartesische Produkte, Elemente der Domänen sind  $n$ -stellige Tupel. Eine Stelle  $s$  kann nur Marken aus der zugehörigen Domäne  $\mathcal{D}_s$  tragen.
- $m_0$  ist eine *Anfangsmarkierung*. Allgemein ordnet eine Markierung  $m : S \rightarrow \mathcal{M}(\mathcal{D})$  jeder Stelle  $s \in S$  eine endliche Multimenge  $\mu_s \in \mathcal{M}(\mathcal{D}_s)$  zu, welche für jede der Marken aus  $\mathcal{D}_s$  die Anzahl ihrer Kopien festlegt, die auf der Stelle enthalten sind<sup>3</sup>.
- $\mathcal{P}$  ist eine Menge von *Schaltmodi*, wobei für jede Transition  $\tau \in T$  eine spezifische Menge  $\mathcal{P}_\tau \subseteq \mathcal{P}$  von Schaltmodi definiert ist.
- $\Phi$  ist eine Menge von Abbildungen, welche für jede Kante  $(s, \tau)$  bzw.  $(\tau, s)$  des Netzes beim Schalten einer Transition  $\tau$  in einem Modus aus  $\mathcal{P}_\tau$  die von den Eingangsstellen konsumierten bzw. die auf den Ausgangsstellen produzierten Marken festlegt:

$$\Phi = \{ \varphi_{s,\tau} : \mathcal{P}_\tau \rightarrow \mathcal{M}(\mathcal{D}_s) \mid (s, \tau) \in F \cap (S \times T) \} \cup \{ \varphi_{\tau,s} : \mathcal{P}_\tau \rightarrow \mathcal{M}(\mathcal{D}_s) \mid (\tau, s) \in F \cap (T \times S) \}$$

- Die Schaltmodi  $\mathcal{P}_\tau$  einer Transition  $\tau$  können jeweils durch eine *Schaltbedingung* (auch: *Schaltformel* oder *Guard*) eingeschränkt werden. Diese wird mit  $guard(\tau)$  bezeichnet.

□

Die Markierung einer Stelle wird in dieser Arbeit in der Form

$$\{ \text{Anzahl} \langle \text{Marke Typ } 1 \rangle, \dots, \text{Anzahl} \langle \text{Marke Typ } n \rangle \}$$

angegeben. Ist von einem Typ genau eine Marke vorhanden, so wird die Anzahl nicht angegeben. Es sei darauf hingewiesen, daß eine Marke aus mehreren Komponenten bestehen kann. In Abbildung 4.5 haben Marken zwei Komponenten: einen Identifikator für den Case sowie einen Wert für die Höhe des gemeldeten Schadens.

Die Abbildungen aus  $\Phi$  werden im Netz mittels *Kantenbeschriftungen* dargestellt. Eine Kantenbeschriftung ist ein Tupel von Variablen, die mit Werten aus den Wertebereichen für die Komponenten der Marken belegt werden können. Dabei muß die Belegung der Variablen mit der Domäne der Stelle konsistent sein. Kantenbeschriftungen haben die Form

$$\text{Anzahl} \langle \text{Variable } 1, \dots, \text{Variable } n \rangle.$$

---

<sup>3</sup>Mitunter wird bei der Definition von Prädikate/Transitionen-Netzen auch die Potenzmenge  $\wp(\mathcal{D})$  verwendet. Dabei handelt es sich um einen Spezialfall des hier gewählten Ansatzes, der insofern eine Einschränkung darstellt, als daß in  $\wp(\mathcal{D})$  nur echte Mengen auftreten, die keine Kopien von Elementen zulassen.



Die Stelligkeit des Variablen-Tupels entspricht der Stelligkeit der Elemente der Domäne der entsprechenden Stelle und damit der Anzahl der Komponenten einer Marke. Die Zuordnung von Komponenten und Variablen erfolgt über die Position im jeweiligen Tupel. Durch die Menge der Schaltmodi werden alle Belegungen der Variablen beschrieben, die ein Schalten der Transition ermöglichen.

Details zur Formulierung der Schaltbedingungen von Transitionen finden sich beispielsweise in [Fre01, Obe96, Wei00]. In dieser Arbeit werden nur sehr einfache Schaltbedingungen verwendet, so daß auf eine ausführliche Formalisierung verzichtet wird. Für unsere Zwecke reicht es aus festzustellen, daß die Schaltbedingung einer Transition mit Hilfe der Variablen in den Kantenbeschriftungen der Ein- und Ausgangskanten der jeweiligen Transition sowie Konstanten aus dem Wertebereich der Variablen gebildet werden kann. Für Belegungen der Variablen mit den Werten der Marken der entsprechenden Stellen kann entschieden werden, ob die Schaltbedingung *wahr* oder *falsch* ist. Die Transitions-Symbole erhalten die Schaltbedingung als Anschrift.

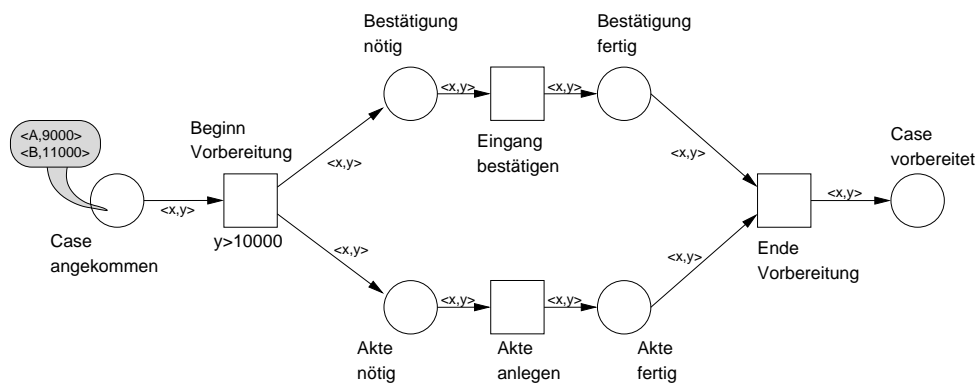


Abbildung 4.5: Das Geschäftsprozeßmodell als Prädikate/Transitionen-Netz.

### Beispiel

Abbildung 4.5 zeigt das Geschäftsprozeßmodell aus Abbildung 4.2 als Prädikate/Transitionen-Netz. Als Anfangsmarkierung wurde für die Stelle **Case angekommen** die Markierung  $\{ \langle A, 9000 \rangle, \langle B, 11000 \rangle \}$  hinzugefügt. Damit sind zwei Cases A und B zu bearbeiten, mit einem Schadenswert von 9000 bzw. 11000. Die Kanten haben entsprechende Kantenbeschriftungen der Form  $\langle x, y \rangle$  erhalten.

Zusätzlich ist der Transition **Beginn Verarbeitung** eine Schaltbedingung  $y > 10000$  zugewiesen worden, es werden also nur noch Schadensfälle bearbeitet, bei denen die Schadenshöhe über 10000 liegt. Damit scheidet die Marke  $\langle A, 9000 \rangle$  für die Belegung der Variablen in der Kantenbeschriftung der Eingangskante von **Beginn Verarbeitung** aus.  $\square$

Wie bei einfachen Petrinetzen ist das dynamische Verhalten von Prädikate/Transitionen-Netzen ebenfalls durch eine *Schaltregel* festgelegt:

**Definition 11 (Schaltregel für Prädikate/Transitionen-Netze)**

Sei  $N = (S, T, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein markiertes Prädikate/Transitionen-Netz. Eine Transition  $\tau$  ist im Modus  $p_\tau \in \mathcal{P}_\tau$  unter einer Markierung  $m$  aktiviert, wenn  $m(s) \geq \varphi_{s,\tau}(p_\tau)$  für jede Stelle  $s \in \bullet\tau$  gilt und  $guard(\tau)$  für  $p_\tau$  wahr ist.

Wenn  $\tau$  im Modus  $p_\tau$  schaltet, so ist die Folgemarkierung  $m'$  gegeben durch:

$$m'(s) = \begin{cases} m(s) - \varphi_{s,\tau}(p_\tau) & \text{falls } s \in \bullet\tau \text{ und } s \notin \tau\bullet \\ m(s) + \varphi_{\tau,s}(p_\tau) & \text{falls } s \notin \bullet\tau \text{ und } s \in \tau\bullet \\ m(s) - \varphi_{s,\tau}(p_\tau) + \varphi_{\tau,s}(p_\tau) & \text{falls } s \in \bullet\tau \text{ und } s \in \tau\bullet \\ m(s) & \text{falls } s \notin \bullet\tau \text{ und } s \notin \tau\bullet \end{cases}$$

Der Schaltvorgang wird in folgender Schreibweise dargestellt:  $m \xrightarrow{\tau, p_\tau} m'$ .

□

Zur Beschreibung des Schaltens einer Transition  $\tau$  im Modus  $p_\tau$  wird die Belegung der Variablen in den Kantenbeschriftungen in der Form

$$Variable\ 1 \leftarrow Wert\ 1, \dots, Variable\ n \leftarrow Wert\ m$$

angegeben.

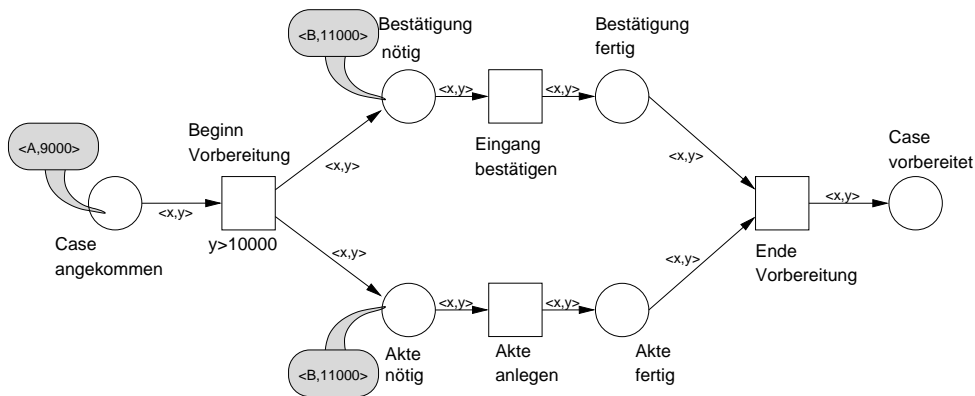


Abbildung 4.6: Schalten und Folgemarkierung in Prädikate/Transitionen-Netzen.

**Beispiel**

Abbildung 4.6 zeigt das Geschäftsprozeßmodell aus Abbildung 4.5 als markiertes Prädikate/Transitionen-Netz nach dem Schalten der Transition **Beginn Vorbereitung** im Modus  $x \leftarrow B, y \leftarrow 11000$ .

□

Schaltet eine Transition eines Prädikate/Transitionen-Netzes, so werden dem Schaltmodus entsprechend alle erforderlichen Marken von den Vorbereichsstellen der Transition entfernt, die entsprechend beim Schalten produzierten Marken werden auf den Nachbereichsstellen abgelegt. Die Menge aller Markierungen des Netzes, die durch wiederholtes

Schalten ausgehend von einer Markierung  $m$  erreichbar sind, wird als die *Menge der (von  $m$  aus) erreichbaren Markierungen*  $[m)$  bezeichnet. Die Ausgangsmarkierung  $m$  ist dabei selbst in dieser Menge enthalten. Die hier vorgestellte Schaltregel stellt die Basis für die Dynamik von markierten Prädikate/Transitionen-Netzen dar. Deren vollständiges dynamisches Verhalten bzw. die Beschreibung dieses Verhaltens wird in Kapitel 6 behandelt.

### 4.3 *VIPmodel*-Netze

Die bisher eingeführten Petrinetz-Konzepte bilden die wesentliche Grundlage für die Einführung der *VIPbusiness*-Erweiterungen im nächsten Kapitel. Da wir für die Umsetzung der in dieser Arbeit vorgestellten Konzepte und Methoden auf den Ergebnissen des *VIP*-Projekts und dem daraus resultierenden Werkzeug *VIPtool* aufbauen, wird in diesem Abschnitt kurz auf die Klasse der *VIPmodel*-Netze eingegangen, die ihrerseits eine leichte Modifikation der oben eingeführten Petrinetz-Konzepte darstellen.

Ein Ziel des *VIP*-Projekts ist die Validierung von Software in frühen Phasen des Entwurfs von Informationssystemen [Fre01, DO95]. Wegen der eingeschränkten Modellierungsmöglichkeiten scheidet dabei einfache Petrinetze aus, so daß nur höhere Petrinetze in Frage kommen. Allgemeine Prädikate/Transitionsnetze [GL81, Smi98] und die meisten ihrer Varianten wie z. B. *algebraische Petrinetze* [Rei98] basieren auf einem abstrakt-mathematischen Formalismus und verfügen über einen hohen Grad an Universalität und Modellierungsmächtigkeit. Die abstrakte Notation und die fehlende Endlichkeit der Datenstrukturen führen jedoch zu großen Implementierungsproblemen. Die von Jensen [Jen96] eingeführten *Coloured Petri Nets* vermeiden diese Nachteile durch eine streng programmiersprachliche Notation und sind somit unmittelbar für die Verwendung beim Einsatz entsprechender Werkzeuge wie beispielsweise *Design-CPN* [Jen95] geeignet, wobei allerdings durch den großen Sprachumfang ein enormer technischer Aufwand beispielsweise bei der Realisierung des Netzsyntax-Parsers entsteht.

Aus diesem Grund wurde im Hinblick auf die im Software-Werkzeug *VIPtool* zu realisierenden *Simulations-Komponenten* (vgl. die Ausführungen zum *VIPsim*-Konzept in Abschnitt 6.2) ein Kompromiß zwischen den Vor- und Nachteilen der oben genannten Netzklassen geschlossen. Die Klasse der *VIPmodel*-Netze besitzt eine überschaubare Syntax und verbindet die kompakte und elegante Notation allgemeiner Prädikate/Transitionen-Netze mit der leichten Implementierbarkeit gefärbter Petrinetze, ohne auf die zur Modellierung praxisrelevanter Systeme benötigte Flexibilität zu verzichten. Die vorgenommenen syntaktischen Vereinfachungen beziehen sich dabei auf folgende Aspekte [Fre01]:

- Vordefinierte Datentypen und Operatoren.

Domänen für Stellen sind kartesische Produkte aus den Basisdatentypen BOOLEAN, INTEGER und STRING oder explizit in einer Datentypdeklaration vereinbarten endlichen Untermengen davon. In Schaltformeln dürfen nur vordefinierte Standardoperatoren wie z. B. '+' , '-' oder '<' verwendet werden. Allgemeine Prädikate/Transitionsnetze erlauben hier die Angabe beliebiger Mengen und Funktionen, gefärbte Petrinetze die programmiersprachliche Definition von beliebigen Datentypen und durch das Schalten von Transitionen ausgeführten Berechnungen.

- Explizit aufzählbare Markierungen.

Markierungen sind Multimengen aus zu den Domänen der Stellen passenden Datentupeln und müssen explizit aufgezählt werden. Angaben generischer Markierungen in Form von Mengenspezifikationen - wie sie z. B. bei algebraischen Petrinetzen möglich sind - sind unzulässig.

- Keine Operatoren in Kantenbeschriftungen.

Kantenbeschriftungen sind Multimengen aus zu den Domänen der Stellen passenden Variablen-tupeln ohne Operatoren. Diese Einschränkung verringert die Ausdrucksmächtigkeit der Netzklasse nicht, da sich jedes Netz mit Operatoren in Kantenbeschriftungen auf einfache Weise in ein äquivalentes Netz ohne diese transformieren läßt, indem die Operatoren in die Schaltformel aufgenommen und gegebenenfalls zusätzliche Variablen eingefügt werden.

- Variablen mit lokalem Gültigkeitsbereich.

Variablen besitzen einen durch ihre Position im Variablen-tupel gegebenen festen Datentyp, dessen Gültigkeitsbereich durch die Umgebung der Transition bestimmt ist. Variablen müssen - im Gegensatz zu Variablen in gefärbten Petrinetzen - nicht vorher deklariert werden, sondern dürfen aus dem Verwendungskontext heraus implizit eingeführt werden.

- Einfacher Variablenbindungsmechanismus.

Jede im Kontext einer Transition vorkommende Variable muss entweder durch eine Vorbereichskantenbeschriftung oder durch die Auswertung der Schaltformel an einen Wert gebunden werden. Die in allgemeinen Prädikate/Transitionsnetzen erlaubte Verwendung von ungebundenen und somit mit einem „Zufallswert“ belegten Variablen an Nachbereichskanten ist unzulässig.

Darüber hinaus können *VIPmodel*-Netze noch zusätzliche Arten von Transitionen enthalten:

- externe Transitionen
- Anfrage-Transitionen
  - Fakt-Transitionen
  - Kausalketten-Transitionen
  - Ziel-Transitionen

*Anfrage-Transitionen* werden in Abschnitt 5.6.3 zur Formulierung von *Business Rules* benutzt, daher wird an dieser Stelle auf eine weitere Erläuterung verzichtet. *Externe Transitionen* sollen im folgenden kurz vorgestellt werden.

Eine Schwierigkeit bei der Simulation des Verhaltens von Geschäftsprozessen stellt die Integration von Einflüssen aus der nicht-modellierten *Umwelt* des Geschäftsprozesses (beispielsweise Kunden oder Lieferanten) dar. Zu Problemen kommt es, wenn der Geschäftsprozeß Zyklen enthält, bei denen die Zahl der Wiederholungen von diesen Umwelteinflüssen abhängt. Um das Verhalten solcher Geschäftsprozesse auch am Modell untersuchen zu können, werden externe Transitionen [Des00, Fre01] benutzt. Dabei wird für Transitionen,

durch deren Schalten die Wiederholung eines Zyklus ausgelöst wird, angegeben, wie oft sie maximal schalten dürfen. Graphisch werden externe Transitionen durch das übliche Transitions-Symbol, ergänzt durch ein stilisiertes  $X$ , dargestellt.

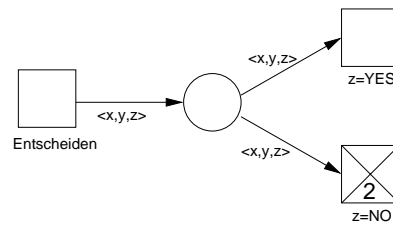


Abbildung 4.7: Externe Transition.

### *Beispiel*

In Abbildung 4.3 ist eine externe Transition im Nachbereich einer Entscheidung verwendet worden. Für den Fall, daß die Entscheidung bei der Bearbeitung eines Case mehrmals getroffen werden muß, ist durch die Verwendung der externen Transition sichergestellt, daß die Bearbeitung bei NO-Entscheidungen höchstens zweimal fortgesetzt wird.

□



## Kapitel 5

# Modellierung mit *VIPbusiness*-Netzen

Im vorangegangenen Kapitel sind einige grundlegende Konzepte zur Modellierung mit Petrinetzen erläutert worden. Anhand von Beispielen wurde gezeigt, daß sich mit den dabei eingeführten Modellierungs-Bausteinen bereits einfache Modelle von Geschäftsprozessen erstellen lassen. Die ausschließliche Verwendung „klassischer“ Petrinetz-Konzepte ist jedoch mit einer Reihe von Nachteilen verbunden.

- Die Modellierung wichtiger Bestandteile von Geschäftsprozessen wird nicht explizit unterstützt.

Zwar lassen sich Dinge wie *Aktivitäten*, *Routing-Entscheidungen* oder *Ressourcen* auch mit den Bausteinen aus dem vorangegangenen Kapitel modellieren, die Erstellung der Modelle gestaltet sich jedoch erheblich schwieriger. Darüber hinaus werden die resultierenden Modelle bereits für einfache Geschäftsprozesse unübersichtlich.

- Zeit- und Kostenmerkmale können nicht modelliert werden.

Da für Petrinetze ursprünglich weder ein Zeit- noch ein Kostenkonzept vorgesehen war, können entsprechende Merkmale von Geschäftsprozessen nicht modelliert werden. Dies stellt insbesondere im Hinblick auf die geplante Analyse der erstellten Geschäftsprozeßmodelle bezüglich wichtiger Leistungsparameter wie der Durchlaufzeit in Kapitel 7 ein Problem dar.

- Nicht-strukturierte Teile von Geschäftsprozessen sind nicht darstellbar.

Die Modellierung von Systemen mit nicht-strukturierten Bestandteilen war beim ursprünglichen Entwurf von Petrinetzen ebenfalls nicht vorgesehen. Viele Geschäfts-

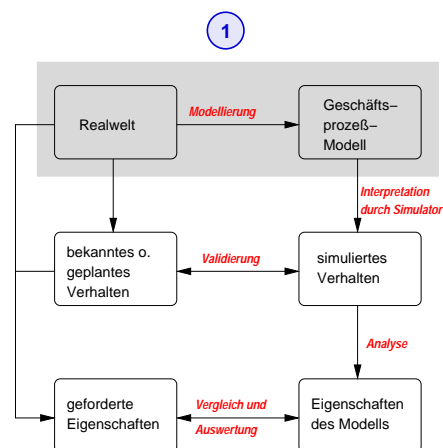


Abbildung 5.1: Schritt 1 (II): *VIPbusiness*-Netze.

prozesse enthalten jedoch sowohl strukturierte als auch nicht-strukturierte Teile, die Modellierbarkeit nicht-strukturierter Teile ist daher eine wichtige Anforderung an die Modellierungs-Konzepte.

In diesem Kapitel werden daher Konzepte und Bausteine zur besonderen Unterstützung der Modellierung von Geschäftsprozessen mit Hilfe von Petrinetzen eingeführt. Allen vorgestellten Konzepten und Bausteinen ist gemeinsam, daß sie sich auf „klassische“ Petrinetz-Konzepte zurückführen lassen. Die im vorangegangenen Kapitel beschriebenen grundlegenden Eigenschaften insbesondere höherer Petrinetze bleiben also erhalten.

Ein besonderes Augenmerk wurde beim Entwurf der Bausteine auf eine Unterstützung des Erstellers der Geschäftsprozeßmodelle gelegt. Daher werden in diesem Kapitel eine Reihe von Bausteinen eingeführt, die als *Kapselung* technischer Details dienen. So werden in Abschnitt 5.1 *Aktivitäts-Transitionen* als Baustein zur Modellierung von Aktivitäten vorgestellt. Durch die Verwendung von Aktivitäts-Transitionen werden die zur Ausführung einer Aktivität notwendigen Teil-Aktivitäten automatisch generiert, dem Modellierer bleibt diese Detailarbeit erspart. Durch die Möglichkeit, die Details einer Aktivitäts-Transition im Modell ein- und ausblenden zu können, kann die Übersichtlichkeit der Geschäftsprozeßmodelle wesentlich erhöht werden, ohne daß auf Detail-Informationen verzichtet werden muß.

In Abschnitt 5.2 werden eine Reihe von Routing-Konstrukten eingeführt, mit deren Hilfe die Reihenfolge von Aktivitäten modelliert werden kann. Für die Modellierung komplizierterer Routing-Konstrukte wird dabei ebenfalls auf das Prinzip der Kapselung zurückgegriffen. Die Arbeit des Modellierers beschränkt sich bei der Verwendung dieser Konstrukte auf die Angabe von Werten für Parameter, beispielsweise Werte für Entscheidungs-Funktionen bei Auswahlentscheidungen.

Die Ausführung von Aktivitäten erfordert in der Regel den Einsatz (beschränkter) Ressourcen. In Abschnitt 5.3 werden daher zwei Möglichkeiten zur Modellierung des Einsatzes von Ressourcen vorgestellt.

*Zeit* spielt in den klassischen Arbeiten zu Petrinetzen keine Rolle, da Carl Adam Petri das Verhalten eines Systems allein durch seine kausale Struktur - unabhängig vom Einfluß zeitlicher Größen - beschreiben wollte [ABC<sup>+</sup>95]. Nichtsdestotrotz sind *Zeitgrößen* bei der Beantwortung von Effizienzfragen für reale Systeme von großer Bedeutung, so daß inzwischen eine Reihe von Ansätzen zur Integration von Zeit in Petrinetze existieren [ABC<sup>+</sup>95]. Auch für die Bewertung von Geschäftsprozessen spielen Kenngrößen wie *Durchlaufzeit* eine zentrale Rolle (vgl. Kapitel 2). In den Abschnitten 5.4.1 und 5.4.2 werden zunächst einige Zeitkonzepte für Petrinetze erläutert, bevor in Abschnitt 5.4.3 ein Konzept für die Integration von Zeitaspekten in Geschäftsprozeßmodelle vorgeschlagen wird.

Auf diesem Zeitkonzept aufbauend werden in Abschnitt 5.5 die in Abschnitt 2.2.3 als relevant herausgefilterten *Kostengrößen* in Geschäftsprozeßmodelle integriert.

Mit der Einführung von Bausteinen zur Modellierung *nicht-strukturierter Teile* von Geschäftsprozessen in Abschnitt 5.6 wird der Tatsache Rechnung getragen, daß viele in der Realität vorkommende Geschäftsprozesse nicht vollständig strukturiert, sondern vielmehr



eine Zusammensetzung aus strukturierten und nicht-strukturierten Teilen sind. Nicht-strukturierte und strukturierte Teile werden zu *hybriden Spezifikationen* nicht vollständig strukturierter Geschäftsprozesse kombiniert.

In Abschnitt 5.7 werden alle in diesem Kapitel eingeführten Bausteine zu einer neuen Klasse von Petrinetzen, den *VIPbusiness*-Netzen, zusammengefaßt.

## 5.1 Aktivitäts-Transitionen

Reale Geschäftsprozesse können sehr komplexe Systeme sein [AHKB00a]. Soll eine Modellierung über eine eher konzeptuelle Nachbildung eines Geschäftsprozesses hinausgehen, ist das Ergebnis schnell ein unübersichtliches Modell. Petrinetze bilden hier keine Ausnahme [Mül95].

In diesem Abschnitt werden *Aktivitäts-Transitionen* zur Modellierung von Aktivitäten eingeführt. Dieses Konzept ermöglicht die *Kapselung* technischer Details von Aktivitäten innerhalb einer Aktivitäts-Transition. Durch die Möglichkeit, die Details von Aktivitäts-Transitionen im Geschäftsprozeßmodell ein- oder ausblenden zu können, gewinnen selbst kleine Modelle an Übersichtlichkeit.

Die Ausführung einer Aktivität erfordert in der Regel die Ausführung einer Reihe von Teil-Aktivitäten. Mit den im vorangegangenen Kapitel eingeführten Modellierungs-Bausteinen ist der Modellierer gezwungen, sämtliche Teil-Aktivitäten selber zu modellieren. In den meisten Fällen handelt es sich bei den Details der Ausführung einer Aktivität um eine Sequenz von Teil-Aktivitäten (vgl. Abschnitt 2.2.3). Da diese Sequenz für viele Aktivitäten zudem aus denselben Teil-Aktivitäten besteht, bietet sich die Einführung eines Bausteins an, durch dessen Verwendung die Sequenz der Teil-Aktivitäten automatisch generiert wird. Dadurch wird die Arbeit des Modellierers erheblich erleichtert.

Zur Kapselung der Teil-Aktivitäten wird das bekannte Konzept der *Vergrößerung* von Petrinetzen verwendet. Vergrößerungen erlauben es, Teilnetze durch einzelne Knoten zu ersetzen. Da Vergrößerungen im weiteren Verlauf dieser Arbeit noch mehrfach zur Einführung neuer Bausteine benutzt werden, sollen sie im folgenden Abschnitt etwas ausführlicher erläutert werden.

### 5.1.1 Vergrößerungen

Zur Einführung des Konzepts der Vergrößerungen werden zunächst die Begriffe *Teilnetz* und *Rand* benötigt. Ein *Teilnetz* eines Petrinetzes ist folgendermaßen definiert<sup>1</sup>:

**Definition 12 (Teilnetz)**

Ein Netz  $N' = (S', T', F')$  heißt *Teilnetz* des Netzes  $N = (S, T, F)$ , wenn  $S' \subseteq S$  und  $T' \subseteq T$  und  $F' = F \cap ((S' \times T') \cup (T' \times S'))$ .

□

---

<sup>1</sup>Die in diesem Kapitel verwendeten Definitionen bekannter Petrinetz-Konzepte basieren wie schon in Kapitel 4 im wesentlichen auf [Bau96, DE95, Des00, Obe96, Rei86].

Der *Rand* eines Teilnetzes ist wie folgt festgelegt:

**Definition 13 (Rand)**

Der *Rand* eines Teilnetzes  $N'$  bezüglich des Gesamtnetzes  $N$  sind diejenigen seiner Knoten, die über Kanten mit dem Restnetz verbunden sind:

$$Rand(N', N) = \{x \in S' \cup T' \mid ((x \bullet \cup \bullet x) \setminus (S' \cup T')) \neq \emptyset\}$$

Vor- und Nachbereich von  $x$  sind hierbei bezüglich  $N$  zu verstehen.

Ein Teilnetz  $N'$  heißt

- *stellenberandet*, wenn sein Rand nur Stellen enthält,
- *transitionsberandet*, wenn sein Rand nur Transitionen enthält.

□

**Beispiel**

In Abbildung 5.2 stellt  $N'$  ein transitionsberandetes Teilnetz des gesamten Geschäftsprozeßmodells dar.

□

Bei der Vergrößerung eines transitionsberandeten Teilnetzes wird das Teilnetz auf eine einzelne Transition abgebildet:

**Definition 14 (Vergrößerung transitionsberandeter Teilnetze)**

Sei  $N = (S, T, F)$  ein Netz und  $N' = (S', T', F')$  ein transitionsberandetes Teilnetz von  $N$ . Ferner sei  $\tau \notin S \cup T$ ; dies wird die Transition, die  $N'$  ersetzt. Für alle  $(x, y) \in F \setminus F'$  wird mit  $\varphi$  eine Operation ausgeführt, die alle Kanten zwischen  $N'$  und seiner Umgebung auf  $\tau$  „zusammenzieht“.

$$\varphi(x, y) = \begin{cases} (x, y) & \text{falls } x, y \in (S \cup T) \setminus (S' \cup T') \\ (x, \tau) & \text{falls } y \in Rand(N', N) \\ (\tau, y) & \text{falls } x \in Rand(N', N) \end{cases}$$

Dann ist das Netz  $\hat{N} = (\hat{S}, \hat{T}, \hat{F})$  mit

$$\begin{aligned} \hat{S} &= (S \setminus S') \\ \hat{T} &= (T \setminus T') \cup \{\tau\} \\ \hat{F} &= \varphi(F \setminus F') \end{aligned}$$

eine *Vergrößerung* von  $N$ .

□

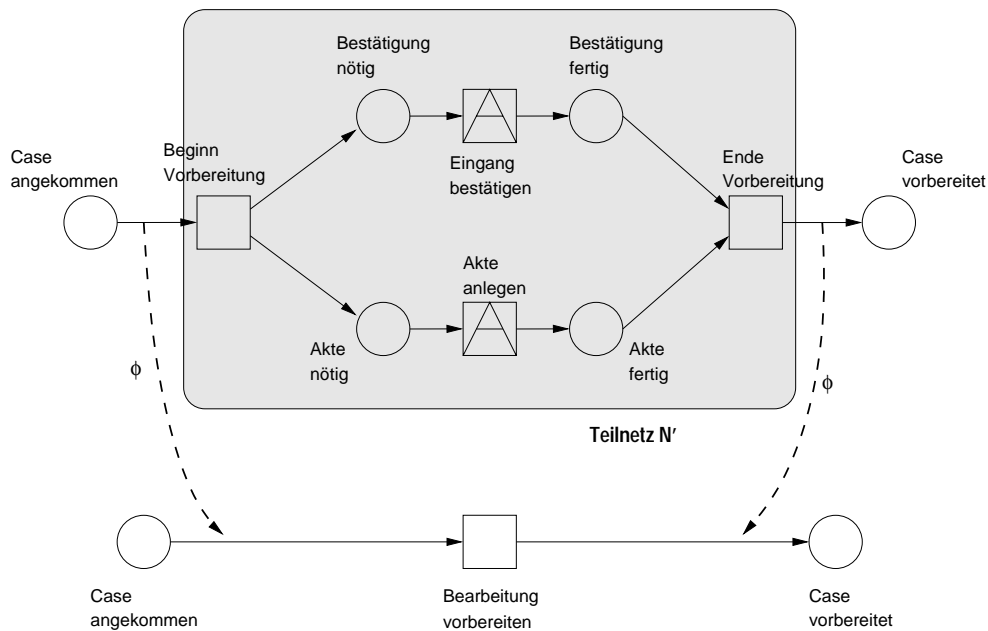


Abbildung 5.2: Teilnetz und Vergrößerung.

### Beispiel

In Abbildung 5.2 wird das transitionsberandete Teilnetz  $N'$  durch die Transition **Bearbeitung vorbereiten** ersetzt.

□

Im folgenden Abschnitt verwenden wir Vergrößerungen zur Einführung eines neuen Modellierungs-Bausteins: Aktivitäts-Transitionen.

### 5.1.2 Aktivitäts-Transitionen als Vergrößerungen

Aktivitäts-Transitionen dienen der Modellierung von Aktivitäten in Geschäftsprozessen. Wie bereits erwähnt, ist zur Ausführung einer Aktivität in der Regel die Durchführung einer Reihe von Teil-Aktivitäten nötig. Die Einführung von Aktivitäts-Transitionen als Baustein zur Modellierung der Teil-Aktivitäten ist im wesentlichen mit zwei Vorteilen verbunden:

- Die Teil-Aktivitäten werden automatisch generiert.

Durch die Verwendung von Aktivitäts-Transitionen wird es dem Modellierer ermöglicht, sich beim Entwurf auf die logische Struktur eines Geschäftsprozesses zu konzentrieren, ohne auf die beispielsweise für Analysezwecke unverzichtbaren Details von Aktivitäten verzichten zu müssen.

- Die Teil-Aktivitäten können ein- oder ausgeblendet werden.

Für die Darstellung der logischen Struktur eines Geschäftsprozesses spielt es keine Rolle, ob eine Sequenz von Teil-Aktivitäten explizit modelliert wird oder nicht.

Im Gegenteil, je mehr Aktivitäten durch Sequenzen von Teil-Aktivitäten dargestellt werden, desto unübersichtlicher wird das Geschäftsprozeßmodell. Ist das Ziel der Modellierung dagegen in erster Linie die möglichst detailgetreue Darstellung eines Geschäftsprozesses im Petrinetzmodell, können alle Teil-Aktivitäten problemlos explizit dargestellt werden. So läßt sich in einer eher konzeptuellen Sicht über die grundsätzliche Struktur eines Geschäftsprozesses oder in einer sehr detaillierten Sicht über Anpassungen an technische Realitäten diskutieren, ohne daß Modifikationen des zugrundeliegenden Petrinetzmodells nötig sind.

Formal kann die Kapselung der Teil-Aktivitäten durch Aktivitäts-Transitionen mit Hilfe des im vorangegangenen Abschnitt eingeführten Konzepts der Vergrößerungen beschrieben werden. Dabei wird ein zusammenhängendes, transitionsberandetes Teilnetz mit Teil-Aktivitäten durch eine Transition, die Aktivitäts-Transition, ersetzt. Bei der Verwendung von Aktivitäts-Transitionen im Kontext von Modellierungs-Werkzeugen sieht es natürlich so aus, daß durch die Verwendung einer Aktivitäts-Transition das transitionsberandete Teilnetz mit Teil-Aktivitäten automatisch generiert wird. Dabei erhält das Teilnetz genau eine *Eingangs*-Transition (diese stellt den *Beginn* der Aktivität dar) sowie genau eine *Ausgangs*-Transition (diese stellt das *Ende* der Aktivität dar). Diese beiden Transitionen stellen den Rand des Teilnetzes dar. Ist die Ausführung der Aktivität an eine Bedingung geknüpft, so wird dies mittels einer entsprechenden Schaltbedingung für die Eingangs-Transition modelliert. Außer für die Eingangs-Transition werden im Teilnetz keine Schaltbedingungen verwendet. Das Teilnetz enthält keine Aktivitäts-Transitionen.

Da für die Formulierung von Reihenfolgebeziehungen von Aktivitäten in dieser Arbeit ausschließlich (die in Abschnitt 5.2 noch einzuführenden) Routing-Konstrukte benutzt werden, sind Aktivitäts-Transitionen grundsätzlich weder vorwärts- noch rückwärtsverzweigt. Damit haben sowohl die Eingangs- als auch die Ausgangs-Transition des Teilnetzes genau eine Eingangs- bzw. Ausgangskante. Die Kantenbeschriftungen innerhalb des Teilnetzes entsprechen der Kantenbeschriftung der Eingangskante.

Mit diesen Erläuterungen definieren wir nun Aktivitäts-Transitionen:

**Definition 15 (Aktivitäts-Transition)**

Sei mit  $N_{Pr} = (S, T, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz gegeben. Ferner sei  $N' = (S', T', F')$  ein transitionsberandetes Teilnetz von  $N_{Pr}$  mit einer Menge von Teil-Aktivitäten  $\tau_{a_1} \dots \tau_{a_n}$ .

Sei  $\tau_{a_1}$  die Eingangs-Transition von  $N'$  sowie  $\tau_{a_n}$  die Ausgangs-Transition von  $N'$ .

Die Kantenbeschriftungen in  $N'$  entsprechen der Kantenbeschriftung der Eingangskante von  $\tau_{a_1}$ .

Eine *Aktivitäts-Transition*  $\tau_A \notin S \cup T$  ist die vergrößerte Darstellung von  $N'$  gemäß Definition 14.

Die Stellen  $s \in S'$  werden als *interne* Stellen bezeichnet. Alle anderen Stellen werden als *externe* Stellen bezeichnet. Gleiches gilt für Transitionen.

Graphisch werden Aktivitäts-Transitionen durch das übliche Transitionssymbol für Petri-netze, ergänzt durch ein großes  $A$  als Beschriftung, dargestellt.

Ist die Ausführung einer Aktivität mit einer Schaltbedingung  $guard(\tau_{a1})$  an der Eingangs-Transition, verknüpft, so erhält die Aktivitäts-Transition diese Schaltbedingung als Anschrift.

Die Menge der Aktivitäts-Transitionen eines Netzes wird mit  $T_A$  bezeichnet.

Ein Prädikate/Transitionen-Netz  $N_{T_A}$  mit Aktivitäts-Transitionen wird durch das Tupel  $(S, T, T_A, F, D, m_0, \mathcal{P}, \Phi)$  beschrieben.

□

Es wird an dieser Stelle die Annahme getroffen, daß es sich bei der durch Aktivitäts-Transitionen modellierten Menge von Teilaktivitäten um eine untrennbare Einheit handelt, d.h., die Aktivität wird stets als ganzes ausgeführt. In dieser Hinsicht entspricht eine Aktivitäts-Transition einer *Transaktion* im Sinne der Datenbank-Theorie [HR83, GR93].

Im Prinzip lassen sich mit Hilfe von Aktivitäts-Transitionen beliebige Mengen von Teil-Aktivitäten kapseln. Auf diese Weise können Teil-Aktivitäten, die stets in einer Einheit ausgeführt werden müssen, durch eine Aktivitäts-Transition zusammengefaßt und auf einfache Weise bei der Modellierung wiederverwendet werden. Im Rahmen dieser Arbeit beschränken wir uns jedoch auf *Sequenzen* von Teil-Aktivitäten. Im Zuge der Diskussion von Zeitmerkmalen in Abschnitt 2.2.3 wurde bereits erläutert, wie eine solche Sequenz aussehen kann. Ausgehend von verschiedenen in Literatur und Praxis vorgeschlagenen Möglichkeiten (vgl. beispielsweise [SO95], [FHS83] oder [Bes86]) wird in dieser Arbeit für Aktivitäten aus dem Bereich der Fertigung folgende Sequenz von Teil-Aktivitäten verwendet:

- Transport,
- Rüsten,
- Bearbeiten,
- fertigungsbedingtes Warten.

Da bei diesen Aktivitäten der Fokus auf der Erzeugung physischer Produkte liegt, ist diese Sequenz für Aktivitäten zur Erbringung einer Dienstleistung ungeeignet (beispielsweise entfällt in der Regel fertigungsbedingtes Warten). Für diese Art von Aktivitäten wird daher folgende Sequenz von Teil-Aktivitäten festgelegt:

- Vorbereiten,
- Bearbeiten.

Durch diese Sequenzen werden die folgenden transitionsberandeten Teilnetze  $N_{seq}(\tau_A) = (S_{seq}, T_{seq}, F_{seq})$  generiert<sup>2</sup>:

---

<sup>2</sup>Der Übersichtlichkeit zuliebe wurden folgende Abkürzungen verwendet:

$$\begin{aligned}
S_{seq} &= \{s_T, s_R, s_B, s_{W_f}\} \\
T_{seq} &= \{B_T, B_R, B_B, B_{W_f}, E\} \\
F_{seq} &= \{(B_T, s_T), (s_T, B_R), (B_R, s_R), (s_R, B_B), (B_B, s_B) \\
&\quad (s_B, B_{W_f}), (B_{W_f}, s_{W_f}), (s_{W_f}, E)\}
\end{aligned}$$

bzw.

$$\begin{aligned}
S_{seq} &= \{s_V, s_B\} \\
T_{seq} &= \{B_V, B_B, E\} \\
F_{seq} &= \{(B_V, s_V), (s_V, B_B), (B_B, s_B), (s_B, E)\}
\end{aligned}$$

Bei der Modellierung einer Aktivität durch eine Aktivitäts-Transition  $\tau_A$  können die internen Stellen wie folgt referenziert werden:

- *Transport*:  $s_T(\tau_A)$ ,
- *Rüsten*:  $s_R(\tau_A)$ ,
- *Bearbeiten*:  $s_B(\tau_A)$ ,
- *fertigungsbedingtes Warten*:  $s_{W_f}(\tau_A)$ ,
- *Vorbereiten*:  $s_V(\tau_A)$ .

Die Transitionen innerhalb einer Aktivitäts-Transition können analog referenziert werden, beispielsweise  $B_V(\tau_A)$  als Beginn der Vorbereitung der Ausführung von  $\tau_A$ .

### **Beispiel**

In Abbildung 5.3 ist das Geschäftsprozeßmodell aus Abbildung 4.5 um Aktivitäts-Transitionen erweitert worden. Dabei ist die Aktivitäts-Transition **Akte anlegen** sowohl in vergrößerter als auch in verfeinerter Sichtweise dargestellt.

□

## **5.2 Routing**

Bei der Bearbeitung von Cases werden Aktivitäten in einer bestimmten Reihenfolge ausgeführt. Die Beschreibung der entsprechenden Reihenfolgen wird auch als *Kontrollfluß-Perspektive* [AHKB00a] von Geschäftsprozeßmodellen bezeichnet<sup>3</sup>. Da die Modellierung des Kontrollflusses beim Erstellen von Geschäftsprozeßmodellen eine zentrale Bedeutung

- 
- *Stellen*: Transport ( $s_T$ ), Rüsten ( $s_R$ ), Bearbeiten ( $s_B$ ), fertigungsbedingtes Warten ( $s_{W_f}$ ), Vorbereiten ( $s_V$ )
  - *Transitionen*: Beginn Transport ( $B_T$ ), Beginn Rüsten ( $B_R$ ), Beginn Bearbeiten ( $B_B$ ), Beginn fertigungsbedingtes Warten ( $B_{W_f}$ ), Beginn Vorbereiten ( $B_V$ ), Ende ( $E$ ).

<sup>3</sup>vgl. dazu auch die Ausführungen zu verschiedenen Dimensionen von Geschäftsprozessen in Abschnitt 2.3.

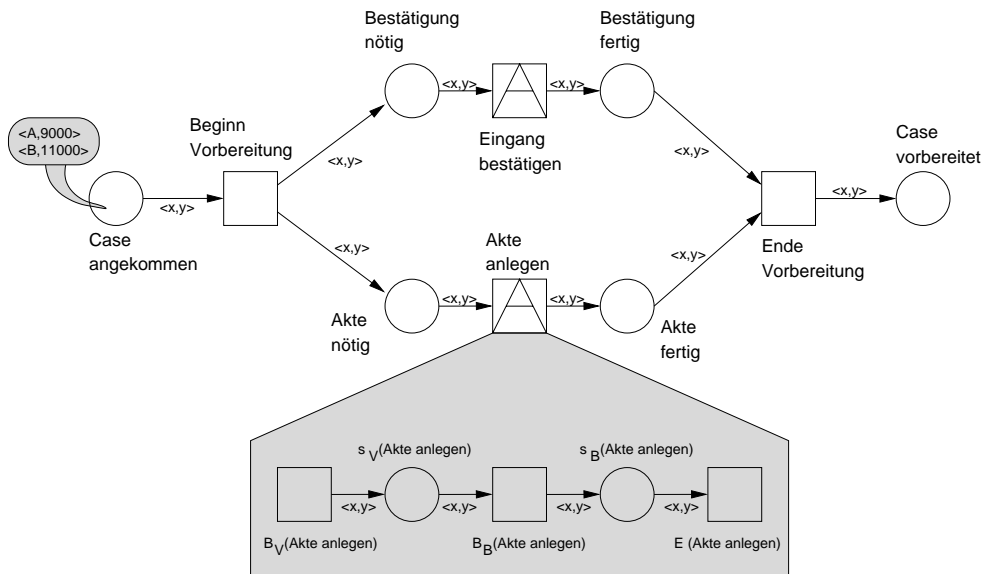


Abbildung 5.3: Das Geschäftsprozeßmodell aus Abbildung 4.5 mit Aktivitäts-Transitionen.

einnimmt, beinhalten jeder Modellierungs-Ansatz und insbesondere jede Modellierungs-Sprache für Werkzeuge eine Reihe entsprechender Bausteine (vgl. dazu die Abschnitte 2.3.2 und 3.2.2). Auch für Petrinetze ist die Verwendung spezieller *Routing-Konstrukte* bereits vorgeschlagen worden (vgl. [Aal98d] oder [AHKB00b]). Eine Definition entsprechender Bausteine, die eine Verwendung im Rahmen der Modellierung mit höheren Petrinetzen erlauben würde, existiert bisher jedoch noch nicht.

Daher werden in diesem Abschnitt eine Reihe von Bausteinen eingeführt, mit deren Hilfe Reihenfolgebeziehungen zwischen zwei Aktivitäten in höheren Petrinetzen dargestellt werden können. Da sich die entworfenen Bausteine auf bekannte Petrinetz-Konzepte zurückführen lassen, können sie problemlos mit den bisher eingeführten Bausteinen kombiniert werden.

Bei der Reihenfolge von Aktivitäten lassen sich die folgenden Fälle unterscheiden (vgl. dazu auch [Sch91], [Aal98d], [AHKB00b] oder [AHKB00a]):

- Die Aktivitäten müssen *sequentiell* ausgeführt werden.
- Die Aktivitäten können *unabhängig voneinander* (auch: *nebenläufig* oder *parallel*<sup>4</sup>) ausgeführt werden. Dabei wird der Geschäftsprozess bei der Ausführung durch *AND-Splits* in nebenläufige Teil-Prozesse unterteilt, die nach ihrer Bearbeitung mittels *AND-Joins* wieder synchronisiert werden können.
- Bezüglich der Ausführung von Aktivitäten muß eine *Auswahlentscheidung* getroffen werden. Hier können wiederum die folgenden Fälle unterschieden werden:

<sup>4</sup>Der Begriff *parallel* ist allerdings an dieser Stelle nicht ganz korrekt, da er oft eine *gleichzeitige* Ausführung bedeutet, die nicht unbedingt gegeben sein muß (vgl. dazu auch die Ausführungen zu *AND-Konstrukten*).

- Die Auswahlentscheidung wird *implizit*<sup>5</sup> getroffen (*impliziter OR-Split*).
- Die Auswahlentscheidung wird *explizit* und *exklusiv* getroffen, es wird also genau eine mögliche Aktivität ausgewählt (*expliziter OR-Split*).
- Die Auswahlentscheidung wird *explizit* und *inklusiv* getroffen, es können also  $n$  aus  $m$  Aktivitäten ausgewählt werden (*inklusive OR*).

Die Zusammenführung der alternativ ausführbaren Teil-Prozesse kann ebenfalls implizit (*impliziter OR-Join*), explizit und exklusiv (*expliziter OR-Join*) oder explizit und inklusiv erfolgen (*inklusive OR-Join*).

Darüber hinaus kann es wichtig sein, bei der Bearbeitung von Cases einzelne Aktivitäten oder Teil-Prozesse wiederholen zu können. Entsprechend muß im Geschäftsprozeßmodell die Möglichkeit der *Iteration* modelliert werden können. In Abschnitt 5.2.4 wird gezeigt, wie dies mit Hilfe expliziter und exklusiver OR-Splits und impliziten OR-Joins getan werden kann.

Prinzipiell können alle Reihenfolgebeziehungen sowie die Iteration mit den klassischen und im Rahmen dieser Arbeit bereits eingeführten Petrinetz-Konzepten modelliert werden. Wir folgen jedoch den Vorschlägen aus [Aal98d] und führen im folgenden spezielle Bausteine für Routing-Konstrukte ein, deren Semantik auf Petrinetze zurückgeführt werden kann. Wie wir noch sehen werden, stellen die eingeführten Routing-Konstrukte lediglich eine *Kapselung* von Petrinetz-Bausteinen dar. Durch dieses Vorgehen wird die Modellierung der Geschäftsprozesse erleichtert, ohne daß die formale Semantik der Petrinetze verloren geht. Eine klare optische Trennung zwischen Aktivitäten und Routing-Konstrukten trägt wesentlich zur Übersichtlichkeit der Geschäftsprozeßmodelle bei.

### 5.2.1 Sequenz

Aktivitäten, die sequentiell auszuführen sind, werden als *kausal geordnet* (auch: *kausal abhängig*) bezeichnet [DFO97]. Dabei sind *Nachbedingungen* der Ausführung einer Aktivität *Vorbedingungen* der Ausführung einer anderen Aktivität<sup>6</sup>. Bei der Modellierung mit Petrinetzen wird für Sequenzen kein spezielles Routing-Konstrukt benötigt. Die kausale Abhängigkeit der Aktivitäten wird bereits durch die Flußrelation  $F$  ausgedrückt (vgl. dazu auch die Ausführungen in Abschnitt 6.1.2 sowie Abbildung 5.4).

---

<sup>5</sup>Dabei handelt es sich immer um eine *exklusive* OR-Verzweigung (vgl. dazu auch die Ausführungen zum *impliziten OR*).

<sup>6</sup>In Abschnitt 2.1.1 wurde bereits erläutert, daß die Ausführung einer Aktivität Vorbedingungen in Nachbedingungen überführt.



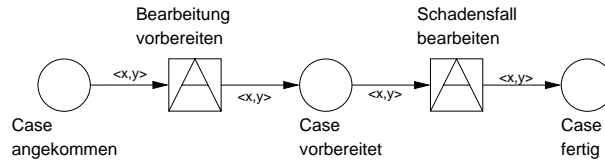


Abbildung 5.4: Sequenz von Aktivitäten.

### Beispiel

In Abbildung 5.4 ist eine Sequenz mit zwei Aktivitäten dargestellt. Dabei muß die Aktivität *Bearbeitung vorbereiten* nach Eintreffen eines Cases stets vor der Aktivität *Schadensfall bearbeiten* ausgeführt werden<sup>7</sup>. □

### 5.2.2 AND-Konstrukte

AND-Konstrukte werden benötigt, um die Bearbeitung eines Cases in zwei oder mehr parallel laufende Teil-Prozesse aufzuteilen bzw. solche Teil-Prozesse wieder zusammenzuführen (auch: zu *synchronisieren*). Die Teil-Prozesse bzw. die Aktivitäten der jeweiligen Teil-Prozesse können unabhängig voneinander ausgeführt werden. Zur Darstellung von AND-Splits und AND-Joins verwenden wir vorwärts- bzw. rückwärtsverzweigte Transitionen, die wir mit einem stilisierten logischen UND versehen. Wir folgen damit der Notation aus [Aal98d], wo Routing-Konstrukte für *Workflow Nets* eingeführt werden. Die Semantik der AND-Konstrukte entspricht dabei der üblichen Semantik vorwärts- bzw. rückwärtsverzweigter Transitionen. Streng genommen könnten wir also auf die Einführung spezieller Transitionen verzichten (da de facto keine Petrinetz-Bausteine „gekapselt“ werden). Um die Bausteine und damit die Geschäftsprozeßmodelle jedoch möglichst einheitlich und übersichtlich zu gestalten, führen wir für alle oben eingeführten Routing-Konzepte (mit Ausnahme der Sequenz) spezielle Routing-Konstrukte ein.

#### Definition 16 (AND-Split, AND-Join)

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen.

Ein *AND-Split* ist eine vorwärtsverzweigte Transition, die keine Vergrößerung eines Teilnetzes ist. Die Menge der AND-Splits wird mit  $T_{AS}$  bezeichnet. Ein *AND-Join* ist eine rückwärtsverzweigte Transition, die keine Vergrößerung eines Teilnetzes ist. Die Menge der AND-Joins wird mit  $T_{AJ}$  bezeichnet.

Graphisch werden AND-Splits und AND-Joins durch das übliche Transitionssymbol für Petrinetze, ergänzt durch ein stilisiertes logisches UND als Beschriftung, dargestellt. □

<sup>7</sup>Wir werden auf diese Art der Reihenfolgebeziehung noch bei der Diskussion von *Kausalitäten* für Abläufe von Geschäftsprozeßmodellen zurückkommen (vgl. Abschnitt 6.2).

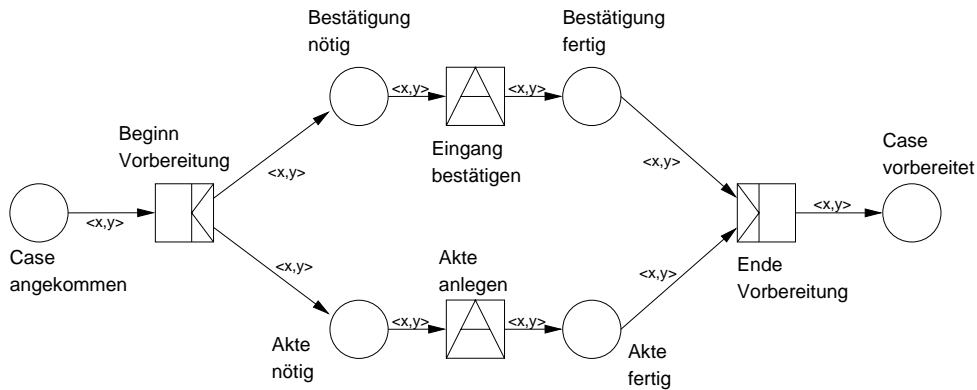


Abbildung 5.5: AND-Split und AND-Join.

### Beispiel

Abbildung 5.5 zeigt den Geschäftsprozeß aus Abbildung 5.3 mit AND-Routing-Konstrukten. Dabei wird der Geschäftsprozeß durch **Beginn Vorbereitung** in zwei Teil-Prozesse aufgespalten. Das Bestätigen des Eingangs der Schadensmeldung sowie das Anlegen der Akte können dabei unabhängig voneinander stattfinden. Sobald die beiden Teil-Prozesse abgearbeitet sind, wird durch **Ende Vorbereitung** wieder synchronisiert.

□

### 5.2.3 OR-Konstrukte

OR-Konstrukte werden benötigt, um Auswahlentscheidungen bei der Ausführung von Geschäftsprozessen zu modellieren. Üblicherweise werden bei der Modellierung von Geschäftsprozessen ausschließlich *exklusive* OR-Konstrukte verwendet. Zwar ist es prinzipiell möglich, *inklusive* OR-Konstrukte als Modellierungs-Bausteine bereitzustellen, die Erfahrung mit diesen Bausteinen hat jedoch gezeigt, daß ihre Verwendung eine häufige Fehlerquelle darstellt. Daher werden im Rahmen dieser Arbeit lediglich exklusive OR-Konstrukte verwendet.

Die Auswahlentscheidung zwischen alternativ auszuführenden Teil-Prozessen wird durch *OR-Splits* modelliert, die Zusammenführung der Teil-Prozesse geschieht mit Hilfe von *OR-Joins*. Sowohl *OR-Splits* als auch *OR-Joins* werden in ihrer impliziten sowie ihrer expliziten Form eingeführt.

Der *implizite* OR-Split zeichnet sich dadurch aus, daß die eigentliche Entscheidung *nicht* explizit modelliert wird. Vielmehr wird eine Entscheidung dadurch getroffen, daß in einem Zustand eine von zwei oder mehr Aktivitäten als erste durchgeführt wird. Der implizite OR-Split wird nicht allzu häufig verwendet. Für den Fall, daß Entscheidungen nicht aktiv herbeigeführt werden können (die Entscheidung hängt beispielsweise von verschiedenen Dingen ab, auf die gewartet werden muß), ist ein impliziter OR-Split jedoch die angebrachtere Form der Modellierung.

Üblicher ist jedoch die Verwendung eines *expliziten* OR-Splits. Dabei wird die Auswahl-

entscheidung explizit mit Hilfe einer (Auswahl-)Aktivität modelliert. Dies entspricht der Tatsache, daß in Geschäftsprozessen Auswahlentscheidungen in der Regel aktiv getroffen werden, sei es automatisch durch Auswertung von Daten oder durch einen erfahrenen Mitarbeiter, der situativ entscheiden muß. Diese Art der Auswahlentscheidung hat zur Folge, daß die Entscheidung *a priori*, d.h., vor der Ausführung der alternativ auszuführenden Aktivitäten oder Teil-Prozesse, getroffen wird. Der entscheidende Unterschied bei der Verwendung eines impliziten oder eines expliziten OR-Splits ist also der *Moment* der Auswahlentscheidung.

Ähnlich ist der Unterschied bei der Zusammenführung der Teil-Prozesse durch OR-Joins. Beim *impliziten* OR-Join ist zur Zusammenführung der Teil-Prozesse keine Aktivität auszuführen. Der implizite OR-Join ist damit lediglich ein strukturelles Hilfsmittel, der *explizite* OR-Join ist dagegen mit der Ausführung einer Aktivität verknüpft. Da bei der Verwendung von OR-Konstrukten die Auswahlentscheidung (und damit der OR-Split) oft eine wichtigere Rolle als die Zusammenführung der alternativen Teil-Prozesse spielt (diese muß lediglich „möglich“ sein), wird der implizite OR-Join weitaus häufiger als seine explizite Form verwendet. Die Teile von Geschäftsprozessen, die durch alternativ auszuführende Teil-Prozesse gekennzeichnet sind, beginnen also häufig mit einer durch einen expliziten OR-Split modellierten Auswahlentscheidung und enden durch eine Zusammenführung der Teil-Prozesse durch einen impliziten OR-Join.

### Impliziter OR-Split, impliziter OR-Join

Ein impliziter OR-Split wird in Petrinetzen mittels eines *Konflikts* (auch: *Konflikt-Situation*) modelliert:

#### Definition 17 (Konflikt)

Sei  $(N, M)$  ein markiertes Petrinetz. Zwei aktivierte Transitionen sind genau dann in der Markierung  $M$  miteinander im Konflikt, wenn sie eine gemeinsame Eingangsstelle haben, die genau eine Marke trägt.

□

Entscheidendes strukturelles Merkmal bei der Modellierung ist also eine vorwärtsverzweigte Stelle. Sind mehrere Transitionen im Nachbereich dieser Stelle aktiviert, so konkurrieren diese Transitionen um die Markierung der Stelle. Die Auswahlentscheidung wird dabei durch die Transition getroffen, die als erste schaltet. Analog finden beim impliziten OR-Join rückwärtsverzweigte Stellen Verwendung. Bei den impliziten OR-Konstrukten handelt es sich also nicht um Routing-Konstrukte im eigentlichen Sinne, da sie nicht aus einzelnen, sondern vielmehr aus einer Kombination verschiedener Bausteine bestehen. Eine Kapselung dieser Kombinationen (wie etwa bei den noch einzuführenden expliziten OR-Konstrukten) macht in diesem Fall keinen Sinn.

#### Definition 18 (Impliziter OR-Split, impliziter OR-Join)

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen.

Ein *impliziter OR-Split* ist eine vorwärtsverzweigte Stelle<sup>8</sup>.

<sup>8</sup>Es sei an dieser Stelle daran erinnert, daß wir in Abschnitt 5.1 gefordert haben, daß Aktivitäten je

Ein *impliziter OR-Join* ist eine rückwärtsverzweigte Stelle.

□

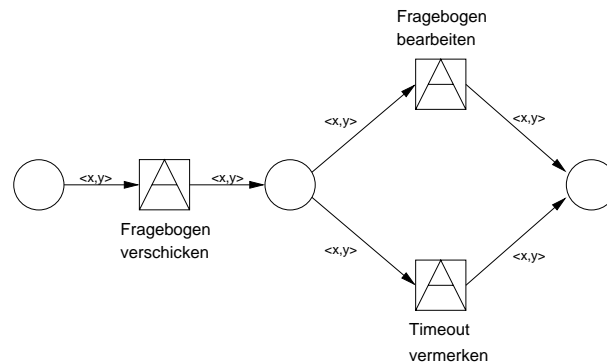


Abbildung 5.6: Impliziter OR-Split, impliziter OR-Join.

### **Beispiel**

Abbildung 5.6 zeigt die Verwendung impliziter OR-Konstrukte. Durch den impliziten OR-Split wird modelliert, daß nach Verschicken eines Fragebogens dieser entweder bearbeitet wird oder ein Time-Out ausgelöst wird (und der Fragebogen bei der weiteren Bearbeitung des Case nicht berücksichtigt wird).

□

Wie eingangs dieses Abschnitts bereits erwähnt, ist die Verwendung des impliziten OR-Splits bei der Modellierung von Geschäftsprozessen eher selten und wird von den wenigsten Formalismen und Werkzeugen unterstützt. Einer der Hauptgründe ist der Umstand, daß die Verwendung von impliziten OR-Splits die explizite Modellierbarkeit von Zuständen voraussetzt. Diese Voraussetzung wird nur von wenigen Modellierungs-Ansätzen und -Werkzeugen erfüllt (vgl. Abschnitte 2.3.2 und 3.2.2). Darüber hinaus ist die Verwendung eines expliziten OR-Splits und der damit verbundenen (expliziten) Auswahlentscheidung in den meisten Fällen die angebrachtere Form der Modellierung. Jedoch stellt der implizite OR-Split die Voraussetzung für die nachfolgende Einführung des expliziten OR-Splits dar.

### **Expliziter OR-Split, expliziter OR-Join**

Analog zu Aktivitäts-Transitionen in Abschnitt 5.1 handelt es sich bei den in dieser Arbeit eingeführten Bausteinen zur Modellierung von expliziten OR-Splits und expliziten OR-Joins um Vergrößerungen von Teilnetzen, die auf den gerade eingeführten Konzepten der impliziten OR-Konstrukte basieren. Das einen expliziten OR-Split modellierende Teilnetz besteht dabei im wesentlichen aus einer Aktivitäts-Transition  $\tau_A$  zur Modellierung der eigentlichen Entscheidung sowie Transitionen  $\tau_1, \dots, \tau_n$  (mit  $n \geq 2$ ) zur Modellierung

---

genau eine Stelle im Vor- und Nachbereich haben. Die verzweigten Stellen sind also die einzigen Stellen im Vor- bzw. Nachbereich der entsprechenden Transitionen.

der Verzweigung, die mit entsprechenden Schaltformeln versehen werden. Mittels einer internen Stelle, die sich im Nachbereich von  $\tau_A$  und im Vorbereich von  $\tau_1, \dots, \tau_n$  befindet, wird der explizite OR-Split auf den bereits bekannten impliziten OR-Split zurückgeführt. Das einen expliziten OR-Join modellierende Teilnetz besteht entsprechend aus einem impliziten OR-Join, ergänzt durch eine Aktivitäts-Transition  $\tau'_A$  im Nachbereich der internen Stelle zur Modellierung der die Zusammenführung herbeiführenden Aktivität.

**Definition 19 (Expliziter OR-Split, expliziter OR-Join)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen.

Sei  $N_{OS} = (S_{OS}, T_{OS}, T_{A_{OS}}, F_{OS})$  ein Teilnetz mit Aktivitäts-Transitionen von  $N_{TA}$  mit

- $S_{OS} = \{s_{OS}\}$ ,  $T_{OS} = \{\tau_1, \dots, \tau_n\}$  mit  $n \geq 2$ ,  $T_{A_{OS}} = \{\tau_A\}$ ,
- alle Transitionen  $\tau \in T_{OS}$  besitzen eine Schaltbedingung  $guard(\tau)$ ,
- $\bullet s_{OS} = \{\tau_A\}$ ,  $\tau_A \bullet = \{s_{OS}\}$ ,
- $\bullet T_{OS} = \{s_{OS}\}$ ,  $s_{OS} \bullet = \{T_{OS}\}$ .

Ein *exklusiver OR-Split*  $\tau_{OS} \notin T \cup T_A$  ist die vergrößerte Darstellung von  $N_{OS}$  gemäß Definition 14.

Die Stelle  $s_{OS}$  ist eine interne Stelle bezüglich  $\tau_{OS}$  und kann über  $s_{OS}(\tau_{OS})$  referenziert werden.

Sei  $N_{OJ} = (S_{OJ}, T_{OJ}, T_{A_{OJ}}, F_{OJ})$  ein Teilnetz mit Aktivitäts-Transitionen von  $N_{TA}$  mit

- $S_{OJ} = \{s_{OJ}\}$ ,  $T_{OJ} = \{\tau_1, \dots, \tau_n\} \cup \{\tau'\}$  mit  $n \geq 2$ ,
- $\bullet s_{OJ} = \{\tau_1, \dots, \tau_n\}$ , für alle  $\tau \in T_{OJ}$  gilt:  $\{\tau\} \bullet = s_{OJ}$ ,
- $\bullet \tau'_A = s_{OJ}$ ,  $s_{OJ} \bullet = \tau'$ .

Ein *exklusiver OR-Join*  $\tau_{OJ} \notin T \cup T_A$  ist die vergrößerte Darstellung von  $N_{OJ}$  gemäß Definition 14.

Die Stelle  $s_{OJ}$  ist eine interne Stelle bezüglich  $\tau_{OJ}$  und kann über  $s_{OJ}(\tau_{OJ})$  referenziert werden.

Graphisch werden ein exklusiver OR-Split und ein exklusiver OR-Join durch das übliche Transitionssymbol für Petrinetze, ergänzt durch ein stilisiertes logisches OR als Beschriftung, dargestellt.

Die Menge der exklusiven OR-Splits eines Netzes wird mit  $T_{OS}$  bezeichnet, die Menge der exklusiven OR-Joins mit  $T_{OJ}$ .

□

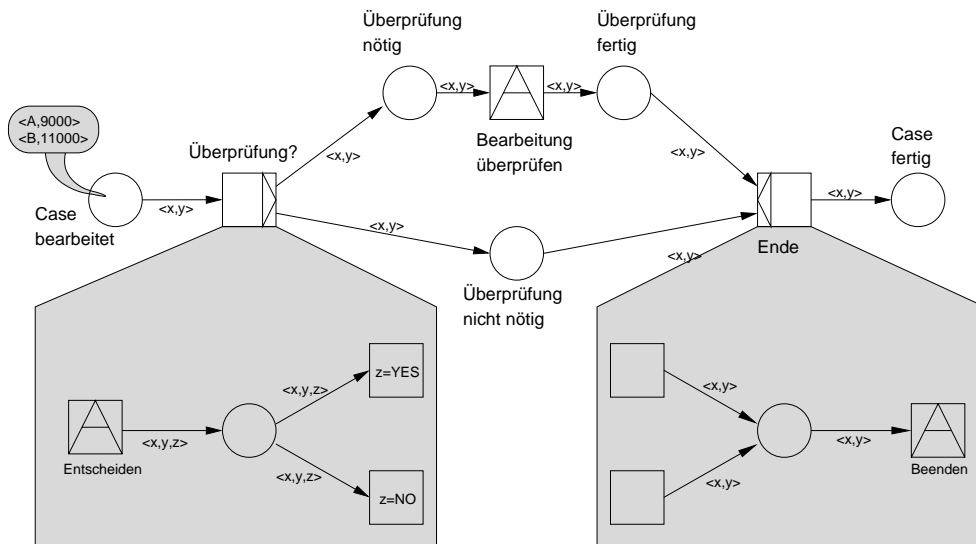


Abbildung 5.7: Expliziter OR-Split, expliziter OR-Join.

### Beispiel

Abbildung 5.7 zeigt die Verwendung der expliziten OR-Konstrukte in einem Geschäftsprozessmodell. Analog zu Aktivitäts-Transitionen können OR-Konstrukte ebenfalls in vergrößerter oder verfeinerter Sichtweise dargestellt werden.

□

Eine spezielle Anwendung des expliziten OR-Joins ist die Modellierung von Aktivitäten, die zu einem sofortigen Beenden der Ausführung des Geschäftsprozesses führen, beispielsweise die Stornierung eines Auftrags durch den Kunden. Solche Aktivitäten können für verschiedene Zustände bei der Ausführung des Geschäftsprozesses stattfinden, alle diese Zustände stellen damit einen möglichen Vorbereitungsbereich der die Aktivität modellierenden Transition dar. Bei der Verwendung herkömmlicher Transitionen stellt sich das Problem, daß gemäß der Definition von Aktiviertheit als Voraussetzung für mögliches Schalten alle Vorbereitungen erfüllt sein müssen. Die einzige Möglichkeit, die Aktivität mit herkömmlichen Transitionen zu modellieren, besteht darin, für jeden möglichen Zustand genau eine Transition zu verwenden. Dieses Vorgehen hat jedoch den Nachteil, daß die gleiche Aktivität durch verschiedene Transitionen modelliert wird. Zudem kann das Geschäftsprozessmodell sehr schnell unübersichtlich werden, insbesondere für den Fall, daß die entsprechende Aktivität für jeden möglichen Zustand bei der Ausführung aktiviert sein muß. Die Verwendung eines exklusiven OR-Joins hat den Vorteil, daß statt multipler Transitionen lediglich die entsprechenden Kanten im Modell eingezeichnet werden müssen. Für jede Markierung, in dem die Aktivität stattfinden können soll, wird eine interne Transition erzeugt, die als Vorbereitung die gegebenenfalls markierten Stellen erhält.

### Beispiel

In Abbildung 5.8 wird die mögliche Stornierung eines Auftrags durch den Kunden mittels eines expliziten OR-Joins **Auftrag stornieren** modelliert.

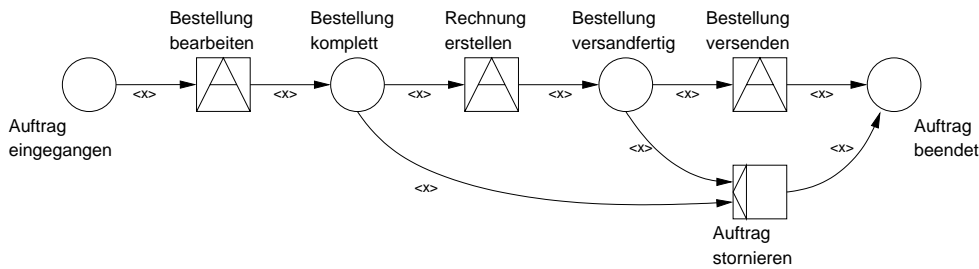


Abbildung 5.8: Spezielle Anwendung für den expliziten OR-Join.

□

## Inklusives OR

Durch die Verwendung eines inklusiven OR-Splits läßt sich eine Auswahlentscheidung modellieren, bei der aus einer Menge alternativer Teil-Prozesse eine beliebige Teilmenge ausgewählt werden kann. Die Bereitstellung entsprechender Routing-Konstrukte ist jedoch mit einem schwerwiegenden Problem verbunden. Zwar läßt sich der inklusive OR-Split problemlos (einem expliziten OR-Split ähnlich) modellieren, Schwierigkeiten bereitet jedoch die Synchronisation der alternativen Teil-Prozesse. Da im Prinzip eine beliebige Teilmenge der alternativen Teil-Prozesse bei der Bearbeitung eines Cases zur Ausführung kommen kann, ist bei der Synchronisation dieser Teil-Prozesse durch einen inklusiven OR-Join nicht klar, wieviele und vor allem welche Teil-Prozesse zu synchronisieren sind. In der Literatur existieren zu diesem Problem eine Reihe von Lösungsvorschlägen, die jedoch mit einigen Nachteilen verbunden sind.

Grundsätzlich lassen sich bei der Synchronisation eines inklusiven OR-Splits zwei Strategien unterscheiden:

- Die weitere Bearbeitung des Cases beginnt erst, nachdem alle ausgewählten Teil-Prozesse beendet worden sind.

Diese Art der Synchronisation ist mit dem Problem verbunden, daß die Information über die Menge der ausgewählten Teil-Prozesse vom inklusiven OR-Split zum inklusiven OR-Join gelangen muß. Dies kann beispielsweise durch die Verwendung zusätzlicher Stellen geschehen. Diese Stellen dienen jedoch ausschließlich dem Informationsaustausch zwischen Split und Join, haben also mit dem eigentlichen Zweck der Modellierung, der Darstellung der Bearbeitung von Cases durch Aktivitäten, nichts zu tun und machen das Geschäftsprozeßmodell damit unübersichtlicher.

- Sobald einer der Teil-Prozesse beendet ist, wird mit der weiteren Bearbeitung des Cases begonnen. Die Beendigung der übrigen Teil-Prozesse findet keine Berücksichtigung mehr. Im Fall der von uns verwendeten Petrinetze stellt sich hier die Frage, was mit den Marken geschehen soll, die nach der Beendigung des ersten Teil-Prozesses im Vorbereich des OR-Joins eintreffen. Da „übrig gebliebene“ Marken in Abschnitt 7.1

als nicht wünschenswertes Phänomen identifiziert werden<sup>9</sup>, sollten sie entfernt werden. Die Modellierung dieses Entfernens ist jedoch alles andere als einfach und trägt ähnlich wie die oben beschriebenen Informations-Stellen zur Unübersichtlichkeit des Geschäftsprozeßmodells bei.

Reihenfolgebeziehungen zwischen Aktivitäten, die sich vermeintlich nur durch die Verwendung inklusiver OR-Konstrukte realisieren lassen, können mit etwas Übung seitens des Modellierers leicht mit Hilfe der übrigen im Rahmen dieser Arbeit eingeführten Routing-Konstrukte modelliert werden. Angesichts der mit der Verwendung von inklusiven OR-Konstrukten verbundenen Nachteile verzichten wir daher auf die Einführung entsprechender spezieller Bausteine.

#### 5.2.4 Zyklen

Bausteine zur Modellierung von *Zyklen* (auch: *Schleifen*, *Iterationen*, *Loops*) werden benötigt, um die Wiederholung von Aktivitäten oder von Teil-Prozessen spezifizieren zu können. Kernstück der Modellierung von Zyklen ist ein expliziter-OR-Split, wobei mindestens eine der Verzweigungs-Transitionen eine externe Stelle aus dem Vorbereich des expliziten OR-Splits als Nachbereich hat. Die Verbindung der entsprechenden Verzweigungs-Transitionen mit den externen Stellen aus dem Vorbereich erfolgt durch einen impliziten OR-Join. Die auf diese Weise modellierten Zyklen haben damit genau einen *Eintrittspunkt* (die externe Stelle) und genau einen *Austrittspunkt* (den expliziten OR-Split)<sup>10</sup>.

##### Definition 20 (Zyklus)

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen. Sei  $\tau_{OS}$  ein expliziter OR-Split. Durch  $\tau_{OS}$  wird ein *Zyklus* realisiert, falls für mindestens eine Transition  $\tau' \in T_{OS}$  gilt:  $\tau' \bullet = \{s'\}$ , wobei  $s'$  eine externe Stelle ist und ein Pfad von  $s'$  nach  $\tau_{OS}$  bezüglich  $F$  existiert.

□

##### Beispiel

In Abbildung 5.9 ist mit Hilfe des exklusiven OR-Splits **Überprüfung OK?** ein Zyklus modelliert worden. Falls die Entscheidung ein *NO* erzeugt, muß die Bearbeitung wiederholt werden.

□

Bei der Verwendung von Zyklen ist die Frage der *Endlichkeit* der Anzahl der Wiederholungen von großem Interesse. Für einen Geschäftsprozeß-Entwurf sollte ausgeschlossen werden können, daß es bei der Bearbeitung von Cases zu *Endlosschleifen* kommt, der Zyklus also unendlich oft durchlaufen wird. Prinzipiell ist es mit den oben eingeführten Bausteinen möglich und eventuell kaum zu vermeiden, Endlosschleifen zu modellieren. Im Rahmen der Simulation des Geschäftsprozeßmodells (vgl. Abschnitt 6.2) lassen sich Endlosschleifen (und damit die Gründe für ihr Auftreten) jedoch identifizieren. Durch entsprechende

<sup>9</sup>Beispielsweise ist mit ihrem Auftreten die Gefahr von Mehrfach-Ausführungen von Aktivitäten verbunden (vgl. Abschnitt 7.1).

<sup>10</sup>vgl. dazu auch die Ausführungen zu verschiedenen Arten von Zyklen, insbesondere *strukturierten Zyklen* in [AHKB00a].



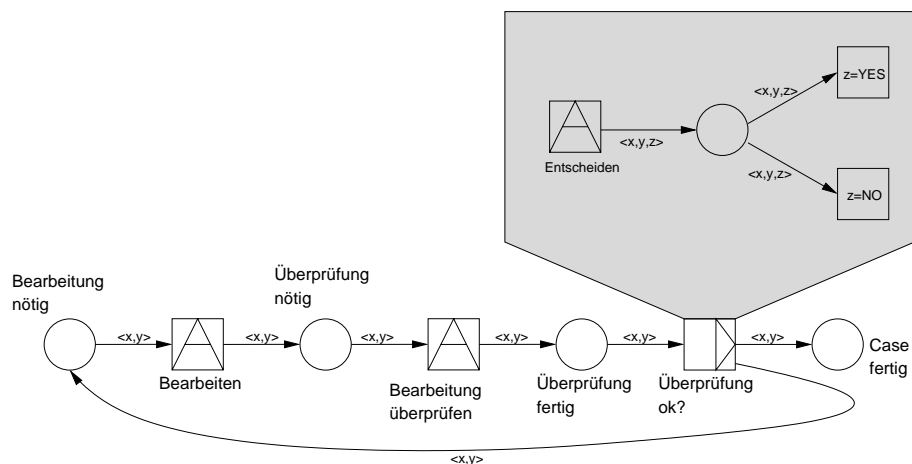


Abbildung 5.9: Ein Zyklus.

Korrektur der Modelle kann für die meisten Geschäftsprozeß-Entwürfe auf diese Weise das Auftreten von Endlosschleifen ausgeschlossen werden.

Selbst wenn Endlosschleifen in dem endgültigen Entwurf eines Geschäftsprozesses prinzipiell nicht explizit ausgeschlossen sind, muß dies für die Handhabung echter Cases zur Laufzeit nicht unbedingt ein Problem darstellen. In [AE01] wird argumentiert, daß sich die Wahrscheinlichkeit einer Wiederholung mit jedem Durchlaufen des Zyklus verringert. Beispielsweise wird eine Überprüfung der Bearbeitung (vgl. Abbildung 5.9) in unserem Beispiel in der Realität sicherlich nicht unendlich oft eine unkorrekte Bearbeitung und damit eine notwendige Wiederholung zur Folge haben, auch wenn dies prinzipiell möglich ist. Endlosschleifen stellen somit weder während der Entwurfs- noch in der Einsatzphase des Geschäftsprozeßmodells ein ernsthaftes Problem dar.

### 5.2.5 Zusammengesetzte Routing-Konstrukte

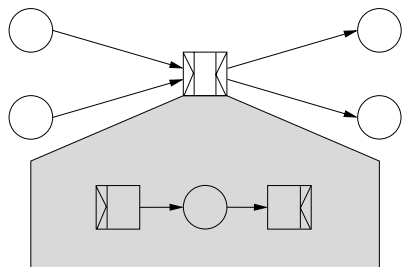
Da bei der Modellierung der Geschäftsprozesse die Aktivitäten im Vordergrund stehen sollten, ist es wünschenswert, daß Routing-Information so kompakt wie möglich modelliert werden kann. Zu diesem Zweck werden in diesem Abschnitt *zusammengesetzte* Routing-Konstrukte eingeführt. Diese bieten zwar keine neue Möglichkeiten zur Modellierung von Routing-Informationen, stellen jedoch eine Möglichkeit zur kompakten Darstellung typischer Routing-Szenarien dar, die in anderen Ansätzen zur Modellierung von Geschäftsprozessen in dieser Form nicht zu finden sind. Wir führen 4 Kombinationen ein:

#### AND-Join und AND-Split

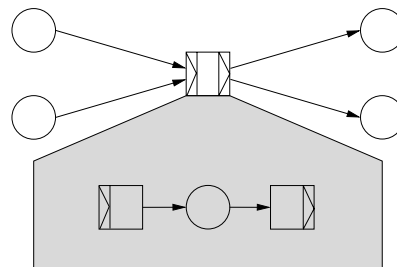
Durch dieses Routing-Konstrukt wird ein in eine Menge von Teil-Prozessen aufgesplitteter Geschäftsprozeß synchronisiert und sofort wieder in eine neue Menge von Teil-Prozessen aufgesplittet (vgl. Abbildung 5.10a).

### AND-Join und expliziter OR-Split

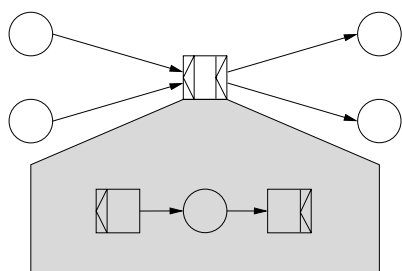
Mit Hilfe dieses Routing-Konstrukts wird ein in eine Menge von Teil-Prozessen aufgesplitteter Geschäftsprozeß synchronisiert und anschließend sofort eine Auswahlentscheidung bezüglich mehrerer möglicher Bearbeitungs-Alternativen getroffen (vgl. Abbildung 5.10b).



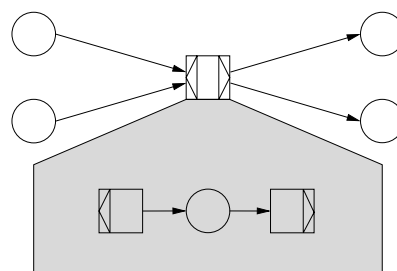
(a) Kombination aus AND-Join und AND-Split.



(b) Kombination aus AND-Join und OR-Split.



(c) Kombination aus OR-Join und AND-Split.



(d) Kombination aus OR-Join und OR-Split.

Abbildung 5.10: Die zusammengesetzten Routing-Konstrukte.

### Expliziter OR-Join und AND-Split

Bei Verwendung dieses Routing-Konstrukts wird zunächst eine während der Bearbeitung des Cases getroffene Auswahlentscheidung synchronisiert, anschließend wird die Bearbeitung des Cases in mehrere Teil-Prozesse aufgesplittet (vgl. Abbildung 5.10c).

### Expliziter OR-Join und expliziter OR-Split

Durch dieses Routing-Konstrukt wird eine während der Bearbeitung des Cases getroffene Auswahlentscheidung synchronisiert und anschließend eine Auswahlentscheidung bezüglich mehrerer möglicher Bearbeitungs-Alternativen getroffen (vgl. Abbildung 5.10d).

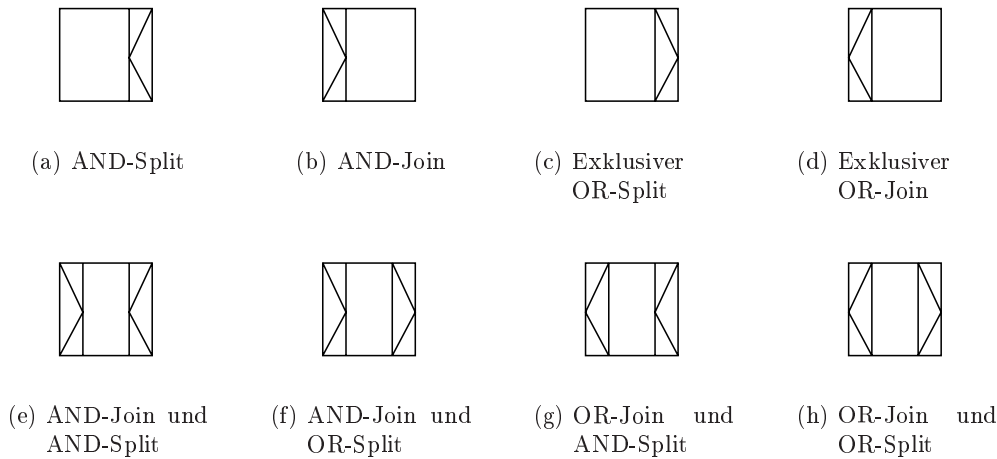


Abbildung 5.11: Alle Routing-Konstrukte.

### 5.2.6 Zusammenfassung

Mit den in diesem Abschnitt eingeführten Routing-Konstrukten lassen sich alle relevanten Reihenfolgebeziehungen zwischen Aktivitäten eines Geschäftsprozesse auf einfache Art und Weise modellieren. Trotzdem kann nicht ausgeschlossen werden, daß durch eine unsachgemäße Verwendung der Routing-Konstrukte Geschäftsprozeßmodelle erstellt werden, bei deren Ausführung es zu Problemen kommen kann. Wie man die entsprechenden Fehler in Geschäftsprozeßmodellen findet und wie sie bei der Modellierung vermieden werden können, wird in Abschnitt 7.1 behandelt.

In Abbildung 5.11 sind noch einmal die graphischen Symbole für alle Routing-Konstrukte in einer Übersicht angegeben. Einige dieser Routing-Konstrukte verwenden wir nun für unser Beispiel-Geschäftsprozeßmodell.

#### *Beispiel*

In Abbildung 5.12 ist das Geschäftsprozeßmodell aus Abbildung 5.3 um Routing-Konstrukte erweitert worden. Die bereits bekannte Aufteilung der Bearbeitungs-Vorbereitung in zwei Teil-Prozesse wurde mit Hilfe der beiden AND-Konstrukte modelliert. Zudem wurde die Bearbeitung des Cases als zusätzliche Aktivität eingeführt. Die Entscheidung darüber, ob die Bearbeitung eines Cases zu überprüfen ist, wurde mit Hilfe eines expliziten OR-Split modelliert. Die Synchronisation des expliziten OR-Splits erfolgt durch einen impliziten OR-Join.

□

## 5.3 Nutzung von Ressourcen

Die Ausführung einer Aktivität zur Bearbeitung eines Cases benötigt in aller Regel Ressourcen. Da Ressourcen im allgemeinen nicht in unbeschränkter Menge vorhanden sind, ist es wichtig, beim Entwurf eines Geschäftsprozesses so früh wie möglich die Effekte des Einsatzes beschränkter Ressourcen zu berücksichtigen.

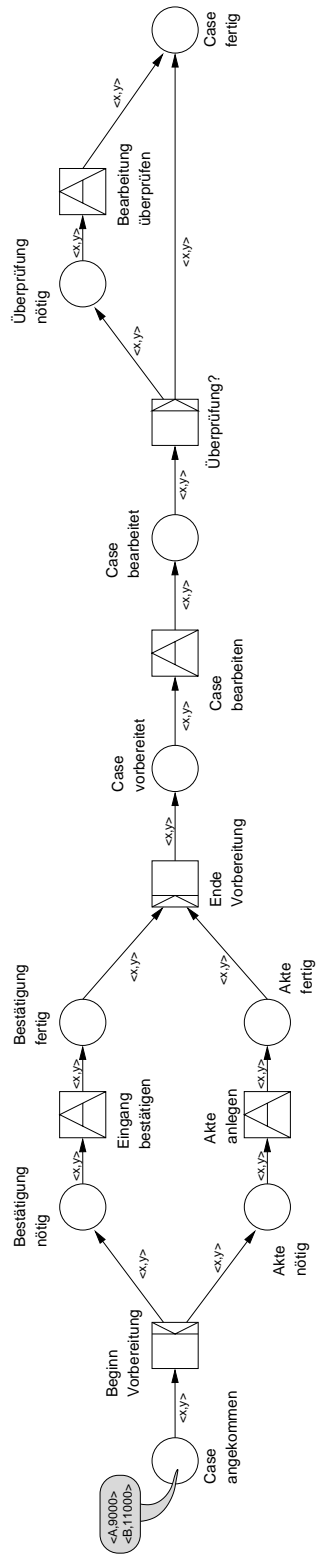


Abbildung 5.12: Geschäftsprozessmodell mit Routing-Konstrukten.

Zur Erweiterung der bisher eingeführten *VIPbusiness*-Konzepte um die Nutzung von Ressourcen müssen die folgenden zwei Punkte behandelt werden:

- Ressourcen-Klassen und Instanzen der Ressourcen-Klassen<sup>11</sup> müssen modelliert werden.
- Die Ressourcen-Klassen müssen mit Aktivitäten verknüpft werden.

In der traditionellen Petrinetz-Theorie werden Ressourcen vor allem im Zusammenhang mit der Modellierung verteilter Systeme betrachtet. Von besonderem Interesse ist dabei vor allem der Zugriff verteilter Komponenten auf gemeinsam genutzte Ressourcen und die daraus resultierende Beeinflussung des Verhaltens des modellierten verteilten Systems. Üblicherweise werden Ressourcen dort mittels Stellen modelliert, diese werden jedoch in der Regel nicht von den übrigen Stellen des Modells unterschieden. Für die Simulations- und Analyse-Konzepte in den Kapiteln 6 und 7 wird eine solche Unterscheidbarkeit der Stellen jedoch benötigt.

Zur Modellierung von Ressourcen-Klassen führen wir daher zunächst *Ressourcen-Stellen* ein:

**Definition 21 (Ressourcen-Stelle)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen.

Eine *Ressourcen-Stelle* ist eine Stelle  $r \in S$  zur Modellierung einer Ressourcen-Klasse.

Die Menge der Ressourcen-Stellen wird mit  $R$  bezeichnet, wobei  $R$  eine echte Teilmenge von  $S$  ist.

Die jeweils verfügbaren Ressourcen einer Ressourcen-Klasse sind durch die Markierung  $m(r)$  der Ressourcen-Stelle gegeben. Insbesondere sind die anfangs verfügbaren Ressourcen durch  $m_0(r)$  gegeben.

$F_R \subseteq (R \times T_A) \cup (T_A \times R)$  bezeichnet die Flußrelation zwischen Ressourcen-Stellen und Aktivitäts-Transitionen. Ressourcen-Stellen werden ausschließlich mit Aktivitäts-Transitionen verbunden.

□

Bezüglich Aktivitäts-Transitionen stellen Ressourcen-Stellen externe Stellen dar.

Eine Ressource wird einer Aktivitäts-Transition für die Ausführung der kompletten Aktivität zugeordnet. Eine Zuordnung von Ressourcen zu Teil-Aktivitäten ist nicht vorgesehen. Dies entspricht der Annahme, daß es sich bei der durch eine Aktivitäts-Transition modellierten Aktivität um eine nicht-trennbare Einheit handelt (vgl. Abschnitt 5.1). Damit sind die Kanten in  $F_R$  Verbindungen zwischen Ressourcen-Stelle und Eingangs- bzw. Ausgangs-Transition der Aktivitäts-Transition.

---

<sup>11</sup>Wie in Abschnitt 2.1.1 bereits erläutert, werden im folgenden die Begriffe *Instanz der Ressourcen-Klasse* und *Ressource* synonym verwendet.

Bei den Marken aus der Domäne  $D_r$  für eine Ressourcen-Stelle kann es sich sowohl um konkrete Ressourcen als auch um verschiedene der Ressourcen-Klasse zurechenbare Typen von Ressourcen handeln. So kann eine Ressourcen-Stelle **Mitarbeiter** beispielsweise Marken wie **Schmidt** oder **Bauer** erhalten, um ganz konkrete Ressourcen zu modellieren (die in den Markierungen dann jeweils maximal einmal vorkommen, z.B.  $\langle \text{Schmidt} \rangle + \langle \text{Bauer} \rangle$ ). Es können aber auch mittels Marken wie **Typ 1** oder **Typ 2** verschiedene Typen von Mitarbeitern unterschieden werden, von denen jeweils mehrere Exemplare vorhanden sein können (z.B.  $2 \langle \text{Typ 1} \rangle + 3 \langle \text{Typ 2} \rangle$ ). Diese Möglichkeiten werden bei der Analyse ressourcenbezogener Kenngrößen in Kapitel 7 genutzt.

Durch einfaches Weglassen der durch obige Definition gegebenen Ressourcen-spezifischen Erweiterungen läßt sich jedes Geschäftsprozeßmodell mit Ressourcen in ein Geschäftsprozeßmodell ohne Ressourcen umwandeln, Ressourcen lassen sich also im Geschäftsprozeßmodell ein- bzw. ausblenden. Diese Eigenschaft wird bei der Simulation sowie der Analyse der Geschäftsprozeßmodelle in den Kapiteln 6 und 7 von großem Vorteil sein.

Zwar haben wir mit  $F_R$  bereits eine Verknüpfung zwischen Ressourcen-Stellen und Aktivitäts-Transitionen, die eigentliche Nutzung von Ressourcen durch Aktivitäten ist damit jedoch noch nicht spezifiziert. Insbesondere muß die Ressourcen-Stelle mit dem „Innenleben“ der Aktivitäts-Transition verknüpft werden. Für die Modellierung der Nutzung von Ressourcen durch Aktivitäts-Transitionen werden zwei Möglichkeiten vorgestellt:

- Modellierung mittels Erweiterungen der Kantenbeschriftungen,
- Modellierung mittels einer zusätzlichen Stelle innerhalb der Aktivitäts-Transition.

Wir betrachten zunächst die Modellierung mittels Erweiterungen der Kantenbeschriftung.

### 5.3.1 Modellierung als Erweiterung der Kantenbeschriftung

Bei der Modellierung der Nutzung von Ressourcen mittels Erweiterungen der Kantenbeschriftung wird das  $n$ -stellige Variablen-tupel der Kantenbeschriftung der Kante zwischen externer Stelle im Vorbereich der Aktivitäts-Transition und der Aktivitäts-Transition um das  $m$ -stellige Variablen-tupel der Kante zwischen Ressourcen-Stelle und Aktivitäts-Transition erweitert (wobei  $m$  in der Regel 1 sein wird). Das nunmehr  $n + m$ -stellige Variablen-tupel wird als Kantenbeschriftung der Kanten innerhalb der Aktivitäts-Transition verwendet. Für den Fall, daß die Aktivitäts-Transition Ressourcen verschiedener Ressourcen-Klassen bzw. mehrere Ressourcen einer Ressourcen-Klasse benötigt, müssen die Kantenbeschriftungen der internen Kanten entsprechend erweitert werden.

#### *Beispiel*

In Abbildung 5.13 ist die Aktivität **Akte anlegen** mit der Ressourcen-Klasse für Mitarbeiter **MA** verknüpft worden. Dabei wurde die Nutzung der Ressource über die Erweiterung der Beschriftungen der Kanten innerhalb der Aktivitäts-Transition modelliert. Man beachte, daß die Eingangs- und Ausgangskanten der Aktivitäts-Transition in der verfeinerten Sicht weggelassen wurden (wie in allen anderen Abbildungen mit verfeinerten Sichten von Aktivitäts-Transitionen auch).

□

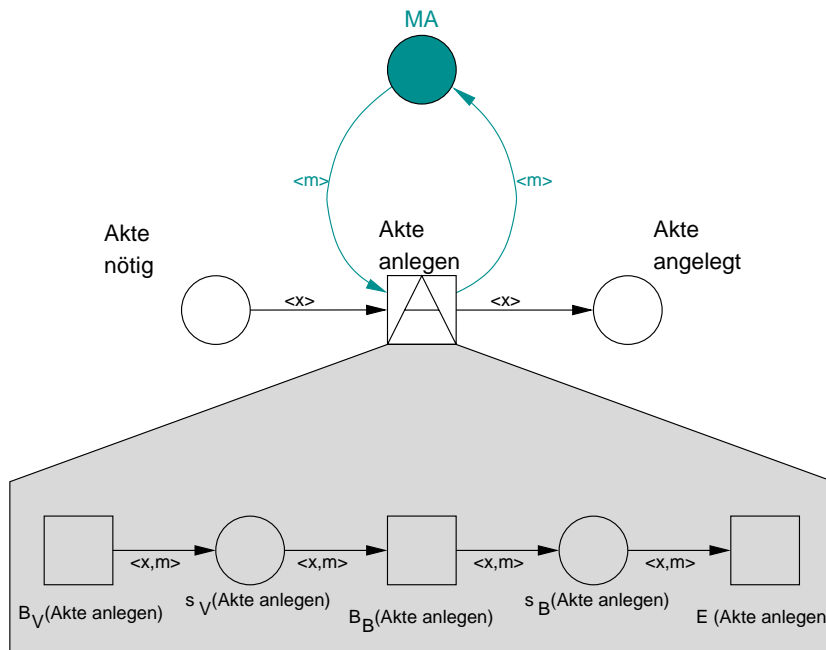


Abbildung 5.13: Ressourcen-Nutzung via Kantenbeschriftung.

Der Vorteil dieser Art der Modellierung liegt in ihrer Kompaktheit. Es werden keine zusätzlichen Netzelemente benötigt. Ein großer Nachteil besteht jedoch darin, daß durch die Verknüpfung der Kantenbeschriftungen keine einfache Trennung zwischen Geschäftsprozeßmodell ohne und Geschäftsprozeßmodell mit Nutzung von Ressourcen möglich ist. Da diese Trennbarkeit für die Simulation sowie die Analyse der Geschäftsprozeßmodelle jedoch einen großen Vorteil darstellt, modellieren wir im Rahmen dieser Arbeit die Nutzung von Ressourcen mittels interner Stellen.

### 5.3.2 Modellierung als interne Stelle

Bei der Modellierung der Nutzung von Ressourcen mittels interner Stellen wird das durch eine Aktivitäts-Transition  $\tau_A$  vergrößerte Teilnetz  $N_{seq}(\tau_A)$  durch eine zusätzliche interne Stelle  $s_{Res}(\tau_A)$  sowie zwei Kanten  $(B_T(\tau_A), s_{Res}(\tau_A))$  (bzw.  $(B_V(\tau_A), s_{Res}(\tau_A))$  für Dienstleistungs-Aktivitäten) und  $(s_{Res}(\tau_A), E(\tau_A))$  erweitert. Diese Kanten erhalten die Beschriftung der Kanten zwischen Ressourcenstelle und  $\tau_A$ . Für jede Ressourcen-Klasse, aus der  $\tau_A$  bei der Ausführung Ressourcen benötigt, wird eine interne Stelle sowie die entsprechenden Kanten erzeugt. Um die Notationen möglichst einfach zu halten, wird für den weiteren Verlauf der Arbeit vereinfachend angenommen, daß für die Ausführung einer Aktivität lediglich Ressourcen *einer* Ressourcen-Klasse benötigt werden. Auf diese Weise wird eine Indizierung der zusätzlichen internen Stelle  $s_{Res}(\tau_A)$  vermieden. Insbesondere die in Kapitel 7 vorgestellten Verfahren zur Analyse von Geschäftsprozeß-Entwürfen können im Bedarfsfall problemlos für die Verwendung mehrerer Ressourcen-Klassen durch eine Aktivität angepaßt werden.

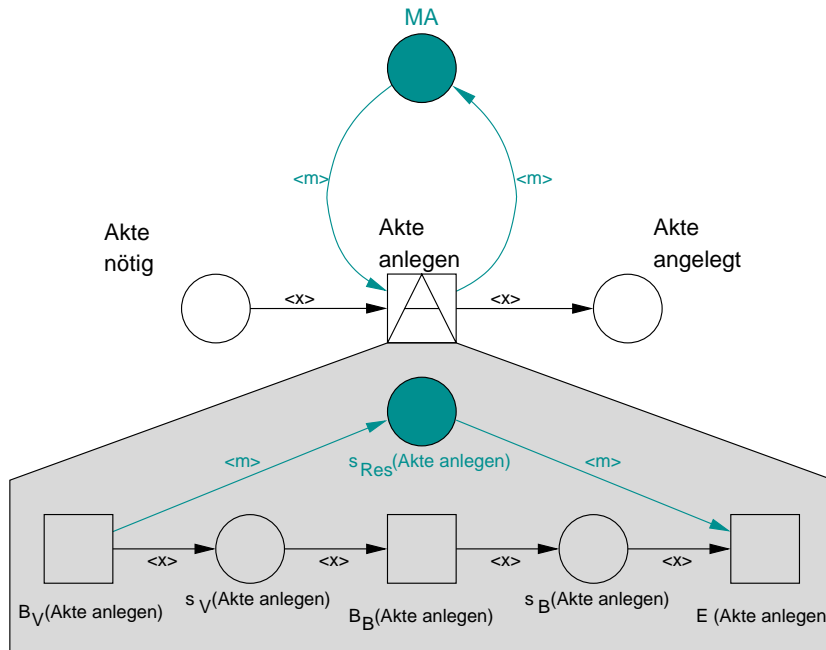


Abbildung 5.14: Ressourcen-Nutzung via interne Stelle.

### Beispiel

In Abbildung 5.13 ist die Aktivität **Akte anlegen** mit der Ressourcen-Klasse für Mitarbeiter **MA** verknüpft worden. Dabei wurde die Nutzung der Ressource mittels einer neuen internen Stelle  $s_{Res}(\text{Akte Anlegen})$  modelliert.

□

Der Nachteil zusätzlicher Netzelemente in Form der zusätzlichen internen Stellen wird durch die dadurch erreichte Erhaltung der Trennbarkeit von Geschäftsprozeßmodellen mit und Geschäftsprozeßmodellen ohne Nutzung von Ressourcen mehr als ausgeglichen, weshalb wir im weiteren Verlauf dieser Arbeit die Nutzung von Ressourcen mittels interner Stellen realisieren.

### Beispiel

In Abbildung 5.15 ist das Geschäftsprozeßmodell aus Abbildung 5.12 um Ressourcen erweitert worden. Die beiden Aktivitäten **Eingang bestätigen** und **Akte anlegen** sind jeweils mit der Ressourcen-Klasse für Mitarbeiter **MA** verknüpft worden. Es werden zwei Typen von Mitarbeitern 1 und 2 unterschieden, von denen jeweils einer anfangs verfügbar ist. Dies wird durch die Anfangsmarkierung von **MA** modelliert.

□

## 5.4 Zeit

Wie zu Beginn dieses Kapitels bereits erwähnt, sind Petrinetze ursprünglich unter ausdrücklicher Auslassung von Zeit konzipiert worden. Mit der Verwendung von Petrinetzen



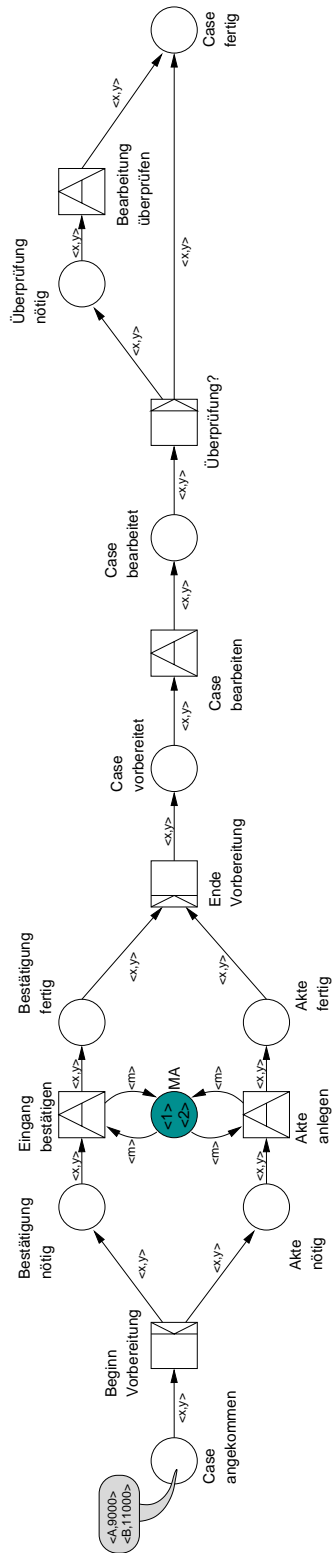


Abbildung 5.15: Geschäftsprozeßmodell mit Ressourcen.

für die Modellierung und Analyse auch quantitativer Aspekte realer Systeme ist jedoch ein Bedarf an Methoden zur Integration von Zeit in klassische Petrinetzmodelle entstanden.

Ein Problem stellt die mögliche Beeinflussung des Verhaltens von Petrinetzen durch die Berücksichtigung von Zeitaspekten dar. In der Tat ist es so, daß die meisten Zeitkonzepte für Petrinetze beispielsweise die Möglichkeiten für das Schalten von Transitionen einschränken oder verhindern [ABC<sup>+</sup>95]. Auf diese Weise wird der Anspruch klassischer Petrinetze, das gesamte mögliche Verhalten eines Systems zu repräsentieren, zunichte gemacht. Bei der Einführung eines Zeitkonzepts für Petrinetze sollte also darauf geachtet werden, daß das Verhalten des zugrundeliegenden Modells ohne Zeitkonzept nicht beeinflußt wird. Auf diese Weise bleiben alle Eigenschaften des Modells ohne Zeitkonzept erhalten.

Bevor wir in Abschnitt 5.4.3 ein Zeitkonzept einführen, das die oben formulierte Anforderung erfüllt, werfen wir zunächst einen Blick auf grundsätzliche Möglichkeiten der Integration von Zeit in Petrinetze sowie einige bekannte Zeitkonzepte mit ihren Vor- und Nachteilen.

### 5.4.1 Grundsätzliche Modellierungsmöglichkeiten

Im Prinzip gibt es vier grundsätzlich verschiedene Möglichkeiten, Zeit in Petrinetze zu integrieren [Ajm95, ABC<sup>+</sup>95]:

1. Zeit wird Stellen zugeordnet.
2. Zeit wird Marken zugeordnet.
3. Zeit wird Kanten zugeordnet.
4. Zeit wird Transitionen zugeordnet.

Bei der Zuordnung von Zeit zu *Stellen* stehen die Marken, die eine Transition  $\tau_1$  beim Schalten in ihrem Nachbereich erzeugt, erst nach Ablauf einer Wartezeit  $t_W$  zur Aktivierung einer Transition  $\tau_2$  zur Verfügung (vgl. Abbildung 5.16).

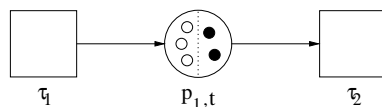


Abbildung 5.16: Zeit als Attribut von Stellen.

Über die Einführung einer *Reiseverzögerung* (original: *traveling delay*) können Zeitaspekte den *Kanten* eines Netzes zugeordnet werden.

Dabei stehen Marken für das Schalten einer Transition erst nach Ablauf einer gewissen Zeitspanne zur Verfügung, die für das Zurücklegen des Wegs zwischen der Vorbereichsstelle und der schaltenden Transition benötigt wird (vgl. Abbildung 5.17).

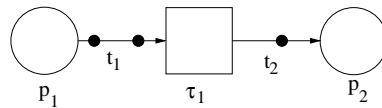


Abbildung 5.17: Zeit als Attribut von Kanten.

Die Zuordnung von Zeit zu *Transitionen* wird als *Schaltdauer* interpretiert. Dabei werden zwei Ansätze unterschieden:

1. Das Schalten der Transitionen wird als *atomarer Vorgang* angesehen.
2. Das Schalten einer Transition wird in *drei Schritte* untergliedert.

Im ersten Fall wird die Schaltdauer durch einen *Timer* dargestellt, mit dessen Herunterzählen begonnen wird, sobald die Transition aktiviert ist.

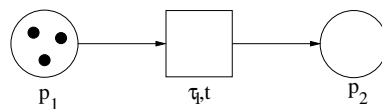


Abbildung 5.18: Zeit als Attribut von Transitionen - atomares Schalten.

Bis zum Stattfinden des eigentlichen Schaltvorgangs (wenn der Timer auf *Null* heruntergezählt wurde) bleiben die Marken im Vorbereich der Transition, um nach dem Schaltvorgang sofort in den Nachbereichsstellen der Transition erzeugt zu werden (vgl. Abbildung 5.18). Durch den Schaltvorgang selbst wird also keine Zeit verbraucht.

Soll Zeitverbrauch durch Schalten modelliert werden, muß der Schaltvorgang in drei Schritte untergliedert werden (vgl. Abbildung 5.19 sowie [Bau96]).

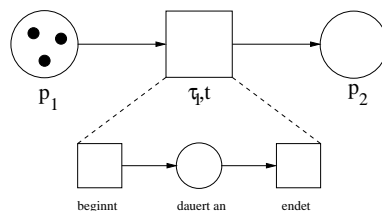


Abbildung 5.19: Zeit als Attribut von Transitionen - Schalten in drei Schritten.

Sobald die Transition aktiviert ist, werden die Marken aus dem Vorbereich der Transi-

tion entfernt (*Schritt 1*). Die Schaltdauer verstreicht (*Schritt 2*). Die Marken werden im Nachbereich der Transition generiert (*Schritt 3*).

Die oben geschilderten grundsätzlichen Möglichkeiten, Zeit in Petrinetze zu integrieren, können natürlich auch kombiniert werden [Bau96]. Die Probleme, die im allgemeinen mit der Einführung von Zeitkonzepten verbunden sind, werden im nächsten Abschnitt anhand einiger bekannter Ansätze aus der Literatur erläutert.

## 5.4.2 Bekannte Ansätze

Eines der bekanntesten Zeitkonzepte wird durch die *Timed Coloured Petri Nets (TCPN)* realisiert (vgl. dazu [Jen95, Mül95, Reu95]). Coloured Petri Nets sind eine High-Level-Netzvariante, bei der die Modellierung von Marken als Objekte mit Attributen unterstützt wird<sup>12</sup>.

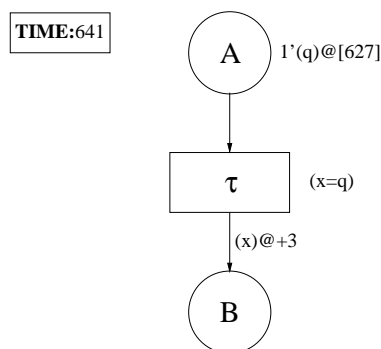


Abbildung 5.20: Modellierung von Ereignissen mit *Timed Coloured Petri Nets*.

Zur Erläuterung des Zeitkonzepts von TCPNs soll das folgende Beispiel (vgl. Abbildung 5.20) genutzt werden:

### Beispiel

Damit die Transition  $\tau$  schalten kann, müssen zwei Bedingungen erfüllt sein:

- $\tau$  benötigt im Vorbereich eine Marke  $m$  des Typs  $q$ .
- Für das Zeitattribut  $t_m$  der Marke  $m$  muß gelten:  $t_m \leq t_{System}$ . Falls dies für keine Marke der Vorbereichsstelle erfüllt ist, wird die Systemzeit  $t_{System}$  hochgezählt.

Die Marken in TCPNs tragen *time stamps*  $t_m$  als Attribute. Für die Entscheidung, ob eine Marke bereits zum Schalten zur Verfügung steht, wird mit der Systemzeit  $t_{System}$  eine globale Systemuhr verwendet. Bezüglich des Schaltvorgangs werden folgende Annahmen getroffen:

<sup>12</sup>[Jen95] verwendet in diesem Zusammenhang den Begriff der *gefärbten* Marken (daher auch der Name der Netzklasse).

- Der Schaltvorgang ist atomar.
- Die Marken werden im Nachbereich der Transition sofort wieder generiert.
- Bevor sie dort für die Aktivierung einer weiteren Transition berücksichtigt werden können, muß die der Transition zugeordnete Bearbeitungszeit verstreichen.

Für das Beispiel aus Abbildung 5.20 wird also durch den Schaltvorgang

$$(q)@[627] \xrightarrow{\tau} (q)@[630]$$

folgendes Verhalten modelliert:

1. Für die Zeitbewertung der Marke in  $A$  gilt:  $t_q = 627 \leq t_{system} = 641$ . Der Typ der Marke  $(q)$  entspricht dem von  $\tau$  zum Schalten benötigten Typ.
2.  $(q)@[627]$  wird zum Schalten von  $\tau$  benutzt.
3. Die Marke  $(q)@[630]$  wird in  $B$  sofort nach dem Schalten von  $\tau$  generiert, muß dort jedoch mindestens 3  $ZE$  warten, bis sie zur Aktivierung weiterer Transitionen genutzt werden kann.

□

Dieses Zeitkonzept ist mit einer Reihe von Problemen verbunden:

- Obwohl die Marken im Nachbereich der Stelle nach dem Schaltvorgang direkt erzeugt werden, sind sie nicht zur Aktivierung von Transitionen nutzbar. Aus der Sicht der Transition, die geschaltet hat, sind die Nachbedingungen also bereits erfüllt, während die Aktivierungsbedingung für eventuell nachfolgende Transitionen erst nach Ablauf der Wartezeit gelten, obwohl es sich um die gleiche Stelle mit der gleichen Markierung handelt. Auch wenn diese Sichtweise formal keine Schwierigkeiten bereitet, ist eine Interpretation für reale Systeme schwierig. Während des Ablaufs der Wartezeit kann nicht eindeutig festgestellt werden, ob Vor- bzw. Nachbedingungen von Aktivitäten erfüllt sind oder nicht. In diesem Zusammenhang wird auch vom Auftreten von *Phantommarken* gesprochen.
- Echte Wartezeit kann nicht modelliert werden. Abhilfe bringt hier auch keine Einführung von Transitionen, durch deren Schalten der Wartevorgang simuliert werden soll, da der Zeitverbrauch für den Schaltvorgang im voraus festzulegen ist. Bei echter Wartezeit handelt es sich jedoch um eine variable Zeitgröße, die vom jeweiligen Verhalten des Systems beeinflusst wird und daher beispielsweise mit unterschiedlichen Lösungen von Konfliktsituationen variieren kann.
- Eventuell können „irgendwann“ aktivierte Transitionen zum Zeitpunkt ihrer Aktivierung nicht mehr schalten, das Verhalten des Systems wird durch die Einführung des Zeitkonzepts also beeinflusst.

Ein weiteres bekanntes Zeitkonzept ist das der *Generalized Stochastic Petri Nets (GSPN)*. Dabei wird Zeit durch Transitionen mit Interpretation des Schaltvorgangs als atomaren Vorgang modelliert (vgl. [ABC<sup>+</sup>95] und Abschnitt 5.4.1).

Dieses Zeitkonzept ist ebenfalls mit einigen Problemen verbunden:

- Zwar wird durch die Einführung eines Timers im Vorbereitungsbereich der Transition die oben beschriebene Problematik der Phantommarken umgangen, das Schaltverhalten des Systems wird durch das Zeitkonzept dennoch beeinflusst (z.B. bei der Auflösung von Konfliktsituationen).
- Echte Wartezeit läßt sich auch hier nicht modellieren, da der durch den Timer modellierte Zeitverbrauch der Transition im voraus festzulegen ist und somit nur nicht-variable Zeitgrößen modellierbar sind.

Den beiden oben geschilderten Zeitkonzepten ist gemeinsam, daß Zeit ausschließlich mit der Ausführung von Aktivitäten verknüpft wird<sup>13</sup>. *Echte Wartezeit* (vgl. dazu auch Abschnitt 2.2.3) ist mit derartigen Zeitkonzepten grundsätzlich nicht darstellbar. Um das Auftreten echter Wartezeit in einem solchen Fall überhaupt einem Objekt im System zuordnen zu können, muß der Wartevorgang als Aktivität modelliert werden. Dies widerspricht jedoch der intuitiven Interpretation des Begriffs *Aktivität*. Da potentiell zwischen je zwei beliebigen Aktivitäten Wartezeit auftreten kann, müßte folglich zwischen je zwei Aktivitäten eine zusätzliche „Warte“-Aktivität modelliert werden, wodurch die Übersichtlichkeit und die Verständlichkeit des modellierten Systems erheblich verschlechtert würde. Gewonnen wäre dadurch allerdings immer noch nichts, da auch den „Warte-Aktivitäten“ eine feste oder variable Zeitbewertung zugeordnet werden muß. Damit ist der Umstand, daß echte Wartezeit ein Ergebnis des Zeitverbrauchs der Aktivitäten in Abhängigkeit von der jeweiligen Ausführung des Geschäftsprozesses ist, nicht modellierbar.

Soll echte Wartezeit modellierbar sein, muß Zeit also den Zuständen eines Systems zugeordnet werden<sup>14</sup>. Für die Aktivitäten (und deren Teil-Aktivitäten) hat dies zur Folge, daß das Schalten der entsprechenden Transitionen im Prinzip der Interpretation eines Schaltvorgangs durch drei Schritte (**beginnt**, **dauert\_an**, **endet**) entspricht (vgl. Abschnitt 5.4.1), wobei der Zeitverbrauch durch die Aktivität der entsprechenden Stelle **dauert\_an** zugeordnet wird (vgl. auch Abschnitt 5.4.3). Tritt echte Wartezeit auf, so kann dies durch den Verbleib einer Marke auf einer Stelle ausgedrückt werden. Die entsprechende Wartezeit wird der Stelle als Attribut zugeordnet.

Probleme wie das Auftreten von *Phantommarken* werden durch die Modellierung von Zeit in Stellen ebenfalls vermieden. Für jede Stelle kann zu jedem Zeitpunkt entschieden werden, ob sie markiert ist (oder nicht markiert ist) und die dadurch modellierten Bedingungen erfüllt sind (oder nicht erfüllt sind). Dadurch ist der Zustand des modellierten Systems zu jedem Zeitpunkt eindeutig festgelegt.

Im folgenden Abschnitt wird ein auf der Zuordnung von Zeit zu Stellen basierendes Zeitkonzept für die Modellierung von Geschäftsprozessen vorgestellt.

### 5.4.3 Zeitmodellierung für Geschäftsprozesse

Aufgrund der Betrachtungen in den vorangegangenen Abschnitten wird folgendes Zeitkonzept für die Modellierung von Geschäftsprozessen vorgeschlagen:

- Zeit wird in Stellen modelliert.

<sup>13</sup>Ein solches Zeitkonzept wird auch als *event-based* bezeichnet [Aal98d].

<sup>14</sup>Ein solches Zeitkonzept wird auch als *state-based* bezeichnet [Aal98d].

Es wird kein globaler Timer benutzt, Marken erhalten keine *time stamps*.

- Zeitbewertungen werden beim Schalten von Transitionen nicht berücksichtigt.

Das *Vipsim*-Simulationskonzept<sup>15</sup> wird durch dieses Zeitkonzept in keiner Weise beeinflusst. Die Generierung von Abläufen erfolgt also unter Auslassung von Zeitaspekten. Die Zeitbewertungen werden anhand des Systemnetzes vorgenommen und anschließend auf die einzelnen Abläufe abgebildet (vgl. Abschnitt 6.3.4). Auf die Generierung der halbgeordneten Abläufe hat dies keine Auswirkungen. Damit ist die Forderung erfüllt, daß ein Zeitkonzept das Verhalten des zugrundeliegenden Systems ohne Zeitkonzept nicht beeinflussen soll.

In Abschnitt 2.2.3 wurden folgende Zeitgrößen als für die Analyse von Geschäftsprozessen relevant herausgearbeitet:

- Aktivitätszeit  $t_A$ 
  - Transportzeit  $t_T$
  - Rüstzeit  $t_R$
  - Bearbeitungszeit  $t_B$
  - fertigungsbedingte Wartezeit  $t_{W_f}$

bzw. (für Dienstleistungs-Aktivitäten)

- Vorbereitungszeit  $t_V$
- Bearbeitungszeit  $t_B$
- echte Wartezeit  $t_{W_e}$

Den unter dem Begriff *Aktivitätszeit* zusammengefaßten Zeitgrößen ist dabei gemein, daß sie allesamt nicht-variable, bekannte Kenngrößen von Geschäftsprozessen sind. Echte Wartezeit dagegen kann mit unterschiedlichen Ausführungen eines Geschäftsprozesses variieren.

In Anlehnung an die Definition von Aktivitäts-Transitionen sowie die Festlegung der durch eine Aktivitäts-Transition modellierte Sequenz der Teil-Aktivitäten in Abschnitt 5.1 lassen sich Zeitgrößen einem Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen wie folgt zuordnen:

- *Aktivitätszeit* (bzw. die unter diesem Begriff zusammengefaßten Zeitgrößen, s.o.) wird den Aktivitäts-Transitionen bzw. den entsprechenden internen Stellen zugeordnet. Bezogen auf eine Aktivitäts-Transition  $\tau_A \in T_A$  ergeben sich folgende Zeitgrößen:
  - $t_T(\tau_A)$  als Transportzeit, die zur Ausführung der durch  $\tau_A$  modellierten Aktivität benötigt wird,

---

<sup>15</sup>Dieses wird zwar erst in Abschnitt 6.2 behandelt, die Vorteile des gewählten Zeitkonzepts in Bezug auf die Simulation der Geschäftsprozeßmodelle soll jedoch an dieser Stelle nicht unerwähnt bleiben.

- $t_R(\tau_A)$  als entsprechende Rüstzeit,
- $t_B(\tau_A)$  als Bearbeitungszeit,
- $t_{W_f}(\tau_A)$  als fertigungsbedingte Wartezeit.

bzw. (für Dienstleistungs-Aktivitäten)

- $t_V(\tau_A)$  als Vorbereitungszeit,
- $t_B(\tau_A)$  als Bearbeitungszeit.

- *Echte Wartezeit*  $t_{W_e}$  wird externen Stellen zugeordnet.

Da eine Ressource stets für die gesamte Dauer der Ausführung einer Aktivität der entsprechenden Aktivitäts-Transition  $\tau_A$  zugeordnet wird, läßt sich die Dauer der Nutzung einer Ressource durch  $\tau_A$  als Summe der Zeitbewertungen der übrigen internen Stellen von  $\tau_A$  ermitteln. Diese Summe wird der internen Stelle  $s_{Res}(\tau_A)$  zugeordnet<sup>16</sup>.

Formal läßt sich dieser Sachverhalt folgendermaßen ausdrücken:

**Definition 22 (Zeitbewertung)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen. Die *Zeitbewertung* von  $N_{TA}$  ist eine Abbildung  $t : S \rightarrow \mathbb{N}^{17}$  mit:

$$t(s) = \begin{cases} t_T(\tau_A) & \text{falls } s = s_T(\tau_A) \\ t_R(\tau_A) & \text{falls } s = s_R(\tau_A) \\ t_B(\tau_A) & \text{falls } s = s_B(\tau_A) \\ t_{W_f}(\tau_A) & \text{falls } s = s_{W_f}(\tau_A) \\ t_V(\tau_A) & \text{falls } s = s_V(\tau_A) \\ t_{Res}(\tau_A) & \text{falls } s = s_{Res}(\tau_A) \\ t_{W_e} & \text{falls } s \text{ externe Stelle ist} \end{cases}$$

Die Stellen erhalten die entsprechende Zeitbewertung als Beschriftung.

Aktivitäts-Transitionen, die in der vergrößerten Sicht betrachtet werden, erhalten die Summe der Zeitbewertungen der Teil-Aktivitäten als Beschriftung. Formal ist diese Zeit jedoch den entsprechenden internen Stellen der Aktivitäts-Transitionen zugeordnet.

□

Da es sich bei den Komponenten der Aktivitätszeit um nicht-variable Zeitgrößen handelt, deren Wert bei der Modellierung bereits bekannt ist oder mit einem Sollwert angesetzt wird (vgl. Abschnitt 2.2.3), können die entsprechenden Werte direkt modelliert bzw. den entsprechenden Stellen des Geschäftsprozeßmodells zugeordnet werden. Echte Wartezeit kann dagegen erst nach der Ausführung des Geschäftsprozeßmodells bestimmt werden. Im Geschäftsprozeßmodell selbst werden sie daher als Variablen modelliert.

<sup>16</sup>An dieser Stelle sei daran erinnert, daß wir uns für die Modellierung der Nutzung von Ressourcen mittels interner Stellen entschieden haben (vgl. Abschnitt 5.3).

<sup>17</sup>Die Verwendung der natürlichen Zahlen an dieser Stelle ist nicht zwingend.



Falls es sich bei einer externen Stelle um eine Ressourcen-Stelle handelt, wird durch die (variable) Zeitbewertung die Dauer modelliert, in der die Ressource ungenutzt bleibt.

Kann bei der Ausführung einer Aktivität auf eine oder mehrere Teil-Aktivitäten des Schemas verzichtet werden (es fällt beispielsweise keine fertigungsbedingte Wartezeit an), so werden die entsprechenden internen Stellen der Aktivitäts-Transition mit  $t(s) = 0$  bewertet.

### *Beispiel*

Abbildung 5.21 zeigt ein Geschäftsprozeßmodell mit Zeitbewertungen. □

## 5.5 Kosten

In Abschnitt 2.2.3 wurden in Analogie zu den relevanten Zeitmerkmalen folgende im Rahmen dieser Arbeit interessierende Kenngrößen für die bei der Ausführung eines Geschäftsprozesses anfallenden Kosten eingeführt:

- Aktivitätskosten  $k_A$ 
  - Transportkosten  $k_T$
  - Rüstkosten  $k_R$
  - Bearbeitungskosten  $k_B$
  - fertigungsbedingte Wartekosten  $k_{W_f}$

bzw. (für Dienstleistungs-Aktivitäten)

- Vorbereitungskosten  $k_V$
  - Bearbeitungskosten  $k_B$
- echte Wartekosten  $k_{W_e}$

Jede dieser Kostengrößen setzt sich dabei aus einem fixen und einem (in Abhängigkeit entsprechender Zeitgrößen) variablen Teil zusammen und ist damit insgesamt eine zeitabhängige Größe:

$$k(t) = k_{fix} + k_{var}(t)$$

Im weiteren Verlauf dieser Arbeit wird von dem genauen Zusammenhang zwischen Zeit und Kosten (z.B. fixer Anteil + linear zeitabhängiger variabler Anteil) abstrahiert. Kosten werden als eine Funktion von Zeit angesehen (vgl. Abschnitt 2.2.3).

Die Ausführung von Aktivitäten und der damit verbundene Zeitverbrauch sind in Abschnitt 5.4 durch Aktivitäts-Transitionen bzw. die internen Stellen des dazugehörigen

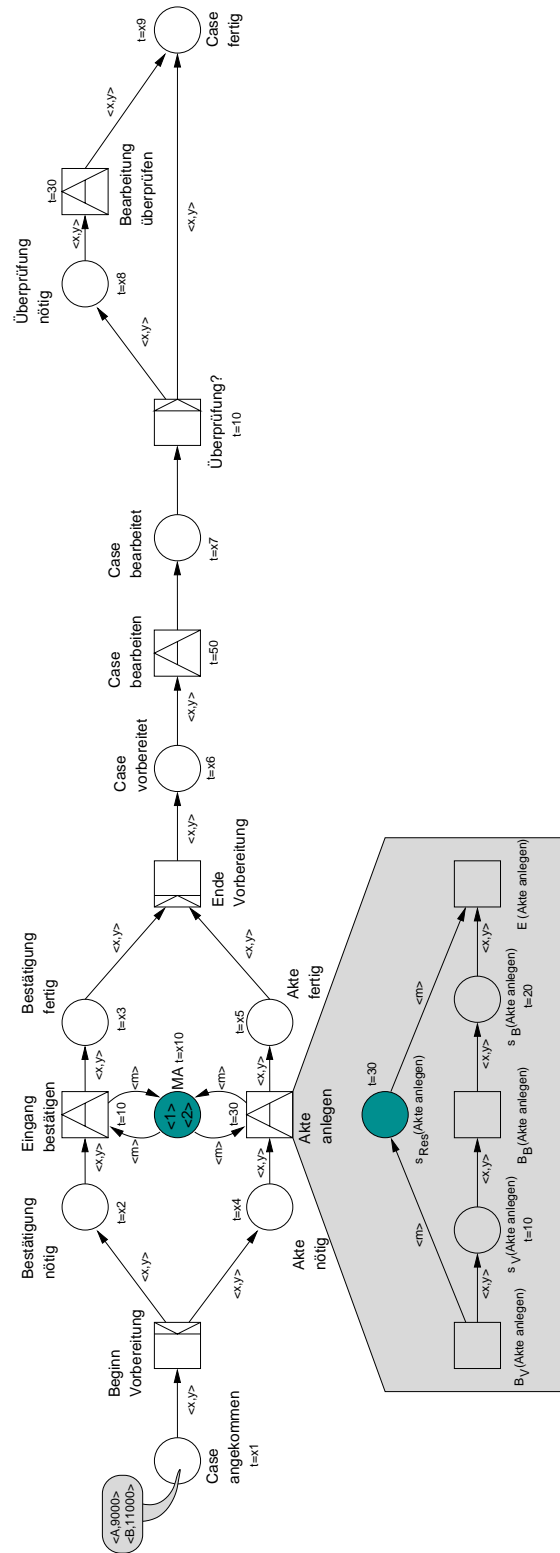


Abbildung 5.21: Geschäftsprozeßmodell mit Zeit.

Teilnetzes der Sequenz der Teil-Aktivitäten modelliert worden. Dementsprechend können Aktivitätskosten Aktivitäts-Transitionen bzw. den entsprechenden internen Stellen zugeordnet werden. Bezogen auf eine Aktivitäts-Transition  $\tau_A$  ergeben sich die folgenden Kostengrößen (in Abhängigkeit vom Zeitverbrauch der jeweiligen Teil-Aktivität):

- $k_T(t_T(\tau_A))$  als Transportkosten, die bei Ausführung der durch  $t_A$  modellierten Aktivität entstehen,
- $k_R(t_R(\tau_A))$  als entsprechende Rüstkosten,
- $k_B(t_B(\tau_A))$  als Bearbeitungskosten,
- $k_{W_f}(t_{W_f}(\tau_A))$  als fertigungsbedingte Wartekosten.

bzw. (für Dienstleistungs-Aktivitäten)

- $k_V(t_V(\tau_A))$  als Vorbereitungskosten,
- $k_B(t_B(\tau_A))$  als Bearbeitungskosten.

Analog zur echten Wartezeit treten echte Wartekosten  $k_{W_e}(t(s))$  ebenfalls nur an externen Stellen auf.

Die Kosten für die Nutzung von Ressourcen stellen eine von der Zeitdauer  $t_{Res}(\tau_A)$  abhängige Größe dar. Als Grundlage für die Kostenfunktion  $k_{Res}(t_{Res}(\tau_A))$  werden dabei die Kostensätze der entsprechenden Ressourcen verwendet (vgl. Abschnitt 2.2.3).

Formal lassen sich Kosten den Stellen eines Netzes wie folgt zuordnen<sup>18</sup>:

**Definition 23 (Kostenbewertung)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen. Die *Kostenbewertung* von  $N_{TA}$  ist eine Abbildung  $k : S \rightarrow \mathbb{N}$  mit:

$$k(s) = \begin{cases} k_T(t(s)) & \text{falls } s = s_T(\tau_A) \\ k_R(t(s)) & \text{falls } s = s_R(\tau_A) \\ k_B(t(s)) & \text{falls } s = s_B(\tau_A) \\ k_{W_f}(t(s)) & \text{falls } s = s_{W_f}(\tau_A) \\ k_V(t(s)) & \text{falls } s = s_V(\tau_A) \\ k_{Res}(t(s)) & \text{falls } s = s_{Res}(\tau_A) \\ k_{W_e}(t(s)) & \text{falls } s \text{ externe Stelle ist} \end{cases}$$

Die Stellen erhalten die entsprechenden Kostenbewertung als Beschriftung.

□

Für Kostenbewertungen von Aktivitäts-Transitionen in der vergrößerten Sicht gelten die bereits für Zeitbewertungen gemachten Aussagen. So erhalten Aktivitäts-Transitionen,

---

<sup>18</sup>Aus Gründen der Übersichtlichkeit wurde für die Zeitbewertung der Stellen  $t(s)$  verwendet. Durch Anwendung von Definition 22 ergibt sich die Zuordnung der entsprechenden Zeitgrößen (z.B.  $t(s) = t_T(\tau_A)$ , falls  $s$  interne Stelle einer Aktivitäts-Transition  $\tau_A$ ).

die in der vergrößerten Sicht betrachtet werden, die Summe der Kostenbewertungen der Teil-Aktivitäten als Beschriftung. Formal sind diese Kosten jedoch den entsprechenden internen Stellen der Aktivitäts-Transitionen zugeordnet.

Falls es sich bei der externen Stelle um eine Ressourcen-Stelle handelt, entspricht die Kostenbewertung der Stelle den Kosten, die durch nicht genutzte Ressourcen entstehen.

Wird bei der Ausführung einer Aktivität auf eine oder mehrere Teil-Aktivitäten des Schemas aus Abschnitt 5.1 verzichtet, so werden die entsprechenden internen Stellen der Aktivitäts-Transition mit  $k(s) = 0$  bewertet.

### **Beispiel**

In Abbildung 5.22 ist das Geschäftsprozeßmodell aus Abbildung 5.21 um Kostenbewertungen erweitert worden. □

Die Einführung von Kostenattributen für Stellen hat keinerlei Auswirkungen auf das mögliche Verhalten des Geschäftsprozeßmodells. Wie die in Abschnitt 5.4 eingeführten Zeitattribute für Stellen werden die Kostenattribute bei der Generierung der halbgeordneten Abläufe eines Systems ignoriert (vgl. Abschnitte 3.1 und 6.2). Das Verhalten eines Systems mit Kostenbewertungen unterscheidet sich also zunächst einmal nicht vom Verhalten eines Systems ohne Kostenkonzept.

Nichtsdestotrotz können Kostengrößen zur ex-post-Analyse der generierten Abläufe herangezogen werden (vgl. Abschnitt 3.1). In Abschnitt 7.3.2 wird dazu ein Verfahren zur kostenoptimalen Verteilung von echter Wartezeit vorgestellt. Dabei kann es vorkommen, daß das *frühestmögliche Schalten*<sup>19</sup> einer Transition nicht in jedem Fall kostenoptimal sein muß. Wird das Schalten der Transition entsprechend hinausgezögert, kommt es in diesem Fall im nachhinein zwar durchaus zu einer Beeinflussung des Schaltverhaltens durch die Einführung eines Kostenkonzepts. Da die möglichen Abläufe des Systems zu diesem Zeitpunkt jedoch bereits generiert worden sind, gehen keine Verhaltensaspekte des zugrundeliegenden Systems ohne Kostenaspekte verloren.

## **5.6 Nicht-strukturierte Teile und hybride Spezifikationen**

Bei der Betrachtung unterschiedlicher Arten von Geschäftsprozessen in Abschnitt 2.1.2 wurde bereits festgestellt, daß nur wenige Geschäftsprozesse vollständig strukturiert sind. Viel häufiger sind Geschäftsprozesse eine Kombination aus strukturierten und nicht-strukturierten Teilen. Geschäftsprozesse diesen Typs werden als *hybrid strukturiert*, die dazugehörigen Modelle als *hybride Spezifikationen* bezeichnet. Die Untersuchung gängiger Ansätze und Werkzeuge zur Geschäftsprozeßmodellierung in den Abschnitten 2.3 und 3.2.2 hat gezeigt, daß der Entwurf von hybrid strukturierten Geschäftsprozessen bislang gar nicht oder nur ansatzweise unterstützt wird. Insbesondere existiert bisher kein Ansatz, der die Erstellung von ausführbaren Modellen von hybrid strukturierten Geschäftsprozessen und die damit verbundenen Analysemethoden im Rahmen von Simulationen ermöglicht.

---

<sup>19</sup>Frühestmögliches Schalten: eine Transition  $\tau$  schaltet, sobald sie aktiviert ist.

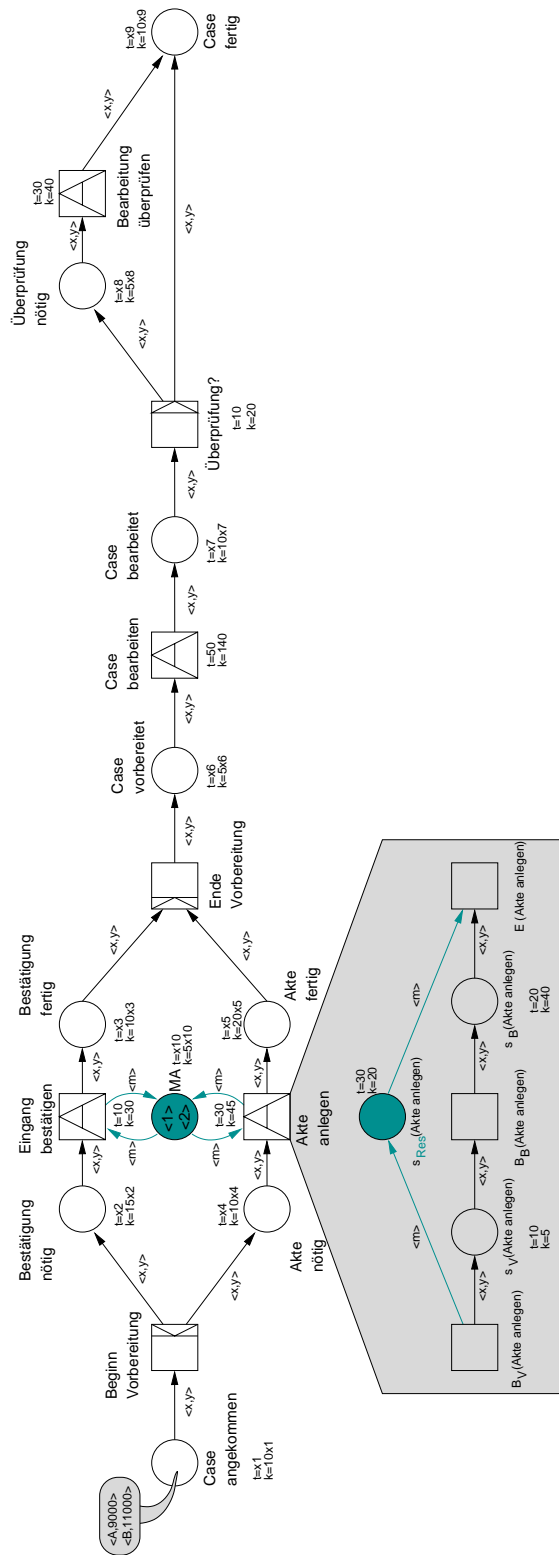


Abbildung 5.22: Geschäftsprozeßmodell mit Kosten.

In den beiden folgenden Abschnitten werden daher Bausteine zur Modellierung nicht-strukturierter Teile von Geschäftsprozessen vorgestellt. Wie in Abschnitt 3.1.1 bereits erläutert, wird dabei im Gegensatz zu herkömmlichen Modellierungsansätzen nicht versucht, auf *konstruktive* Weise das gesamte mögliche und gewünschte Verhalten nicht-strukturierter Teile von Geschäftsprozessen im Modell zu spezifizieren. Vielmehr werden die einzelnen Aktivitäten nicht-strukturierter Teile zunächst ohne irgendwelche Festlegungen bezüglich ihrer Ausführung modelliert. Gegebenenfalls werden die nicht gewünschten oder unerlaubten Variationen bei der Ausführung von Aktivitäten mit Hilfe von Regeln ausgeschlossen. Dieses Vorgehen bei der Modellierung - die Spezifikation des *nicht* erlaubten oder gewünschten Verhaltens auf *deklarative* Weise in Form von Regeln - führt bei nicht-strukturierten Teilen von Geschäftsprozessen zu wesentlich kompakteren Modellen (vgl. dazu das Beispiel in Abschnitt 5.6.2).

In Abschnitt 5.6.4 wird gezeigt, wie die Modelle der nicht-strukturierten Teile eines Geschäftsprozesses mit den Modellen der strukturierten Teile zu einem einzigen Modell, der *hybriden Spezifikation* eines Geschäftsprozesses kombiniert werden können.

Die Modellierung der nicht-strukturierten Teiles eines Geschäftsprozesses erfolgt durch Modellierung einer entsprechenden Menge *nicht-strukturierter Segmente*. Für die Modellierung eines nicht-strukturierter Segments werden 5 Bausteine benötigt:

- ein Startzustand,
- ein Zielzustand,
- eine Menge von Netzfragmenten zur Modellierung der Aktivitäten,
- eine Menge von Ressourcen-Stellen,
- eine Menge von Business Rules.

Abbildung 5.23 zeigt das Schema für die graphische Darstellung eines nicht-strukturierter Segments (vgl. auch Abbildungen 5.24 und 5.26).

Der *Start-* und der *Zielzustand* des nicht-strukturierter Segments bilden die *Schnittstellen* zwischen nicht-strukturierter Teil und dem Rest des Geschäftsprozeßmodells. Die Menge der im nicht-strukturierter Teil auszuführenden Aktivitäten wird mit Hilfe einer Menge von *Netzfragmenten* modelliert, die für die Ausführung der Aktivitäten benötigten Ressourcen durch eine Menge von Ressourcen-Stellen. Um die Ausführung der Aktivitäten im nicht-strukturierter Teil steuern oder einschränken zu können, wird eine Menge von *Geschäftsregeln* oder *Business Rules* angegeben, die bei der Ausführung der Aktivitäten unbedingt einzuhalten sind.

### 5.6.1 Startzustand und Zielzustand

Der *Startzustand* eines nicht-strukturierter Segments besteht aus einer Menge von Stellen. Diese Stellen modellieren die Bedingungen, die erfüllt sein müssen, damit bei Ausführung

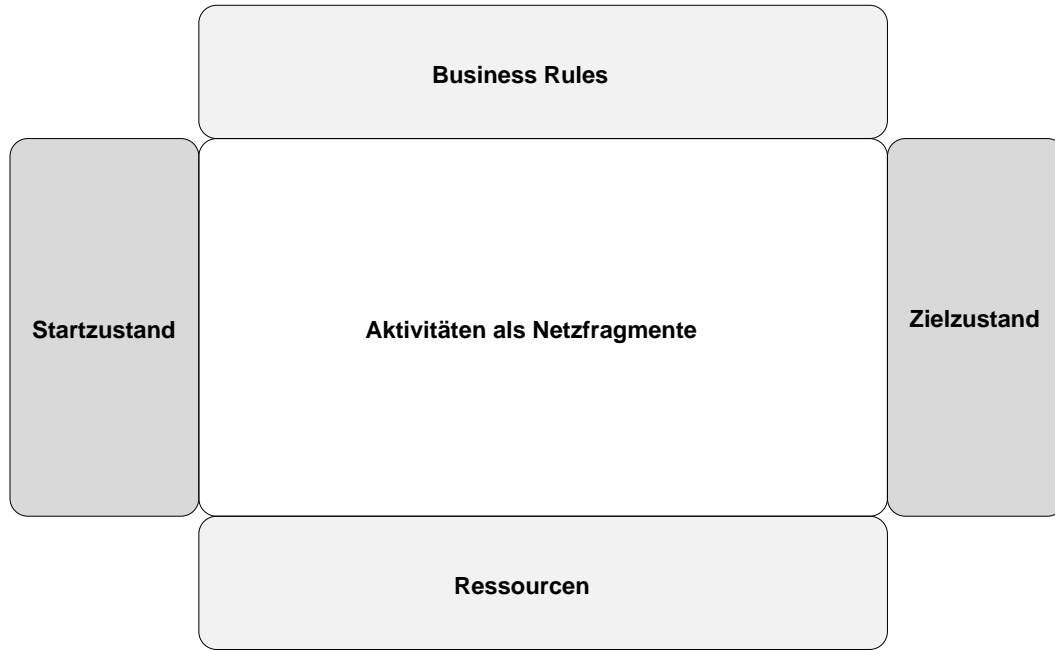


Abbildung 5.23: Schema für die graphische Darstellung nicht-strukturierter Segmente.

eines Geschäftsprozesses der Übergang vom strukturierten Teil zum nicht-strukturierten Segment stattfinden kann. Dabei kann es ausreichen, daß nur ein Teil dieser Bedingungen erfüllt sein muß, damit die Ausführung der Aktivitäten des nicht-strukturierten Segments beginnen kann. In diesem Fall kann es vorkommen, daß die Anzahl der Möglichkeiten für die Ausführung der Aktivitäten (beispielsweise bezüglich der Reihenfolge) eingeschränkt wird, da für einige Aktivitäten die Bedingungen für den Beginn der Ausführung noch nicht erfüllt sind.

**Definition 24 (Startzustand)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen zur Modellierung eines Geschäftsprozesses. Sei  $ns$  ein nicht-strukturiertes Segment dieses Geschäftsprozesses. Der *Startzustand* von  $ns$  wird durch eine Menge  $S_{0_{ns}} \subseteq S$  beschrieben.

□

Die Stellen zur Modellierung der Bedingungen für den Beginn der Ausführung des nicht-strukturierten Segments müssen also in der Menge der Stellen des Modells für den strukturierten Teil enthalten sein. Gleiches gilt für die Stellen zur Modellierung der Bedingungen für den *Zielzustand* eines nicht-strukturierten Segments:

**Definition 25 (Zielzustand)**

Sei  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen zur Modellierung eines Geschäftsprozesses. Sei  $ns$  ein nicht-strukturiertes Segment

dieses Geschäftsprozesses. Der *Zielzustand* von  $ns$  wird durch eine Menge  $S_{z_{ns}} \subseteq S$  beschrieben.

□

Die durch die Stellen des Zielzustands modellierten Bedingungen müssen nach Ausführung der Aktivitäten des nicht-strukturierten Segments gelten. Für den Beginn der Ausführung des auf ein nicht-strukturiertes Segment folgenden Teils kann es bereits ausreichen, daß nur eine Teilmenge der modellierten Bedingungen erfüllt ist (die entsprechenden Stellen markiert sind). Auch hier gilt: unter Umständen werden die Möglichkeiten der weiteren Ausführung eingeschränkt, beispielsweise für den Fall, daß eine Auswahlentscheidung dadurch getroffen wird, daß für die Ausführung einer der möglichen Alternativen noch nicht alle Vorbedingungen erfüllt sind.

Start- und Zielzustand eines nicht-strukturierten Segments werden in Abschnitt 5.6.4 benutzt, um nicht-strukturierte Segmente und strukturierte Teile eines Geschäftsprozesse zu hybriden Spezifikationen zu verschmelzen. Vorher werden jedoch die Bausteine zur Modellierung der eigentlichen Inhalte nicht-strukturierter Segmente vorgestellt: Fragmente und Business Rules.

## 5.6.2 Fragmente

Kernstück eines jeden nicht-strukturierten Segments ist die Menge der Aktivitäten, die auszuführen sind, um vom Start- zum Zielzustand des nicht-strukturierten Segments zu gelangen. In Abschnitt 3.1.1 wurde bereits erläutert, daß sich nicht-strukturierte Teile eines Geschäftsprozesses von den strukturierten Teilen durch die *Variabilität* der Ausführung der Aktivitäten unterscheiden. Während die Ausführung der Aktivitäten in strukturierten Teilen stets auf dieselbe Art und Weise erfolgt, kann sie für nicht-strukturierte Teil *reihenfolgevariabel*<sup>20</sup> oder *modusvariabel*<sup>21</sup> sein.

Die Menge der auszuführenden Aktivitäten eines nicht-strukturierten Segments wird durch eine Menge von *Netzfragmenten* modelliert. Für diese Fragmente wird keine explizite Reihenfolgebeziehung angegeben. Damit ermöglicht diese Form der Modellierung zunächst einmal die Ausführung der entsprechenden Aktivitäten in allen möglichen Reihenfolge-Kombinationen<sup>22</sup>, eventuell eingeschränkt durch eine Menge von zu berücksichtigenden Business Rules (vgl. Abschnitt 5.6.3). Auf diese Weise lassen sich Teile von Geschäftsprozessen spezifizieren, deren Ausführung reihenfolgevariabel ist. Für Aktivitäten, deren Ausführung *modusvariabel* ist, wird für jeden möglichen Modus ein eigenes Netzfragment angegeben.

Ein (minimales) Netzfragment besteht aus genau einer Aktivitäts-Transition und einer Menge von Stellen. Den in Kapitel 4 eingeführten grundlegenden Petrinetz-Konzepten entsprechend werden durch die Menge der Stellen im Vorbereich der Aktivitäts-Transition

---

<sup>20</sup>Reihenfolgevariabel: die Ausführung der Aktivitäten kann in unterschiedlicher Reihenfolge erfolgen.

<sup>21</sup>Modusvariabel: die Aktivität kann in verschiedenen Modi ausgeführt werden, beispielsweise kann eine Benachrichtigung per Post oder per Telefon erfolgen.

<sup>22</sup>Wie Netzfragmente bei der Ausführung der modellierten Aktivitäten im Rahmen der Simulation eines Geschäftsprozeßmodells mit nicht-strukturierten Segmenten kombiniert werden können, wird in Abschnitt 6.3.6 erläutert.



die Bedingungen modelliert, die vor Ausführung der durch die Aktivitäts-Transition modellierten Aktivität erfüllt sein müssen. Entsprechend werden durch die Stellen im Nachbereich der Aktivitäts-Transition die Bedingungen modelliert, die nach Ausführung der Aktivität erfüllt sind.

**Definition 26 (Netzfragment, Menge von Netzfragmenten)**

Ein *Netzfragment* ist ein zusammenhängendes Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen und ohne Anfangsmarkierung  $nf = (S_{nf}, T_{nf}, T_{A_{nf}}, F_{nf}, \mathcal{D}_{nf}, \mathcal{P}_{nf}, \Phi_{nf})$  mit den Eigenschaften:

- $|T_{A_{nf}}| = 1$ ,
- Alle Transitionen in  $T_{nf}$  sind interne Transitionen der Transition in  $T_{A_{nf}}$ .
- Die Transition in  $T_{A_{nf}}$  hat weder einen leeren Vor- noch einen leeren Nachbereich:

$$\tau_A \in T_{A_{nf}} : \bullet\tau_A \neq \emptyset \text{ und } \tau_A\bullet \neq \emptyset.$$

Die *Menge von Netzfragmenten* eines nicht-strukturierten Segments  $ns$  wird mit  $NF_{ns}$  bezeichnet und besteht aus Netzfragmenten mit paarweise disjunkten Transitionenmengen: keine Transition eines Netzfragments gehört zu mehreren Netzfragmenten.

□

Die paarweise Disjunktheit der Transitionenmengen von Netzfragmenten in obiger Definition stellt sicher, daß jede Aktivität innerhalb des nicht-strukturierten Segments in genau einem Netzfragment spezifiziert wird.

Es ist grundsätzlich möglich, daß ein Netzfragment auch mehr als eine Aktivität enthält. Die Ausführung dieser Aktivitäten darf jedoch nicht reihenfolgevariabel sein. So gesehen stellen Netzfragmente mit mehreren Aktivitäten strukturierte Teilnetze innerhalb des nicht-strukturierten Segments dar. Im Rahmen dieser Arbeit werden jedoch ausschließlich Netzfragmente mit einer Aktivität verwendet.

Werden zur Ausführung der Aktivitäts-Transition eines Netzfragments Ressourcen benötigt, so werden diese in der bekannten Weise mittels interner Stellen modelliert (vgl. Abschnitt 5.3.2). Dabei werden die entsprechenden Ressourcen-Stellen mit der Aktivitäts-Transition des jeweiligen Netzfragments über Kanten verbunden. Werden die innerhalb eines nicht-strukturierten Segments benötigten Ressourcen auch außerhalb dieses Segments benötigt, wird die entsprechende Ressourcen-Stelle aus Gründen der Übersichtlichkeit reproduziert. Formal handelt es sich dabei jedoch um dieselbe Stelle (vgl. auch Abbildung 5.26).

Auch wenn das Verhalten und die Simulation nicht-strukturierter Segmente erst in Abschnitt 6.3.6 behandelt werden, soll an dieser Stelle auf einen in diesem Zusammenhang wichtigen Aspekt bei der Modellierung hingewiesen werden. Bei der Ausführung der Aktivitäten eines nicht-strukturierten Segments werden die jeweiligen Netzfragmente über ihre Stellen mit anderen Netzfragmenten sowie dem Anfangs- und Endzustand des nicht-strukturierten Segments verbunden. Dabei können nur *gleiche* Stellen verbunden werden: die Vorbedingung(en) zur Ausführung der Aktivität eines Netzfragments müssen also Nachbedingung(en) in anderen Netzfragmenten oder Bedingung(en) des Startzustands

sein, entsprechend die Nachbedingung(en) der Ausführung der Aktivität eines Netzfragments Vorbedingung(en) in anderen Netzfragmenten oder Bedingung(en) des Zielzustands. Grundsätzlich würde es dabei ausreichen, daß die Domänen der zu verbindenden Stellen gleich sind. Dies ist die Voraussetzung dafür, daß eine Marke der einen auch eine Marke der anderen Stelle sein kann. In dieser Arbeit werden Stellen, die bei Ausführung des nicht-strukturierten Segments miteinander verbunden werden können, auch gleich benannt. Formal handelt es sich also um mehrfache Exemplare der gleichen Stelle. Auf diese Weise ist sofort ersichtlich, wie die Netzfragmente miteinander sowie mit den Stellen des Start- und Zielzustands verknüpft werden können. Entscheidend für die Ausführbarkeit eines nicht-strukturierten Segments ist dabei die Möglichkeit, durch Verbindung der Netzfragmente vom Start- zum Zielzustand zu gelangen. Dabei müssen alle erfüllten Bedingungen des Startzustands in Bedingungen des Zielzustands überführt werden. Daraus ergibt sich die Anforderung, daß die Stellenmengen der Netzfragmente nichtleere Schnittmengen besitzen müssen. Dies sollte bei der Modellierung der Fragmente berücksichtigt werden.

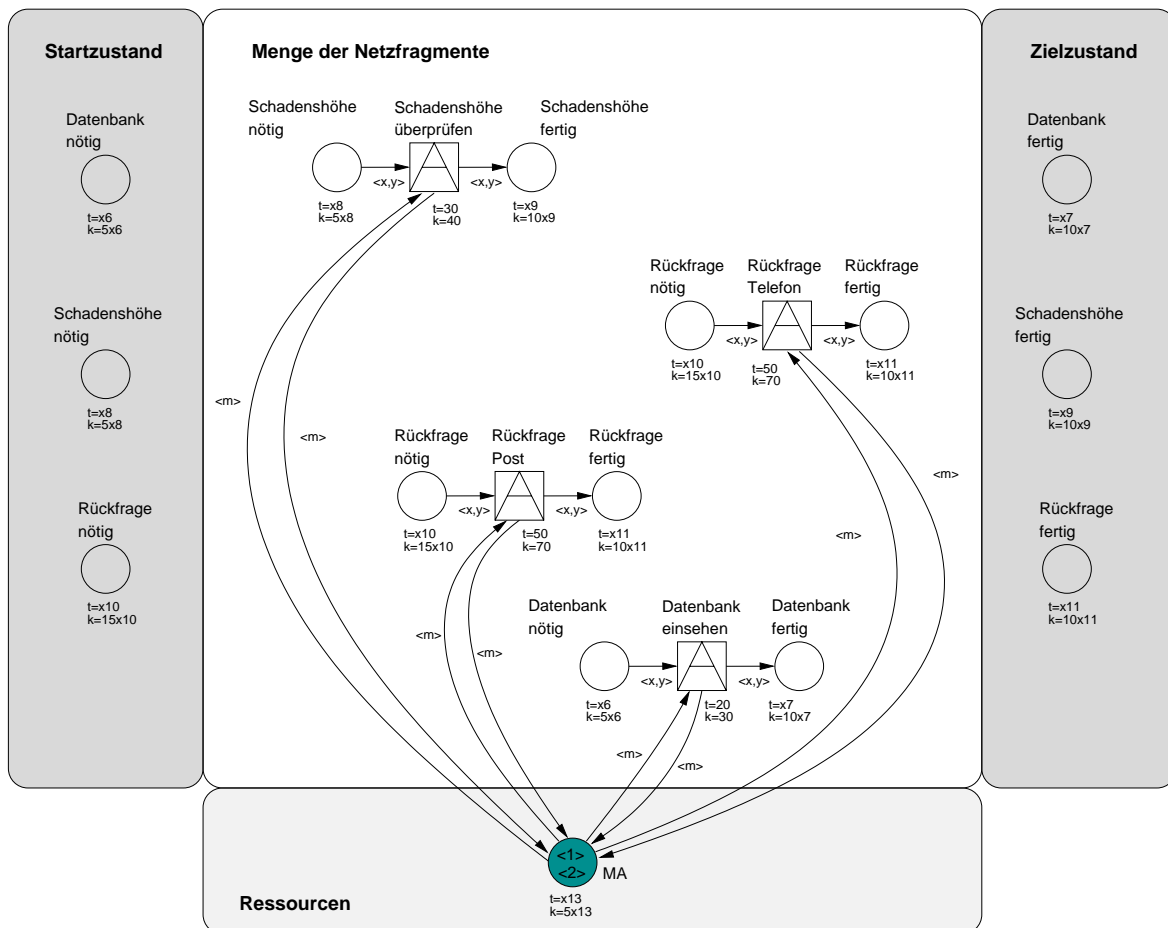


Abbildung 5.24: Der nicht-strukturierte Teil des Geschäftsprozesses aus Abbildung 5.22: Startzustand, Zielzustand, Menge der Netzfragmente.

### *Beispiel*

Die eigentliche Bearbeitung der Schadensmeldung (modelliert durch die Aktivitäts-Transition `Case bearbeiten`, vgl. beispielsweise Abbildung 5.22) ist bisher noch nicht genauer betrachtet worden. Zur Bearbeitung einer Schadensmeldung sind drei Aktivitäten auszuführen:

- Aus der Datenbank der Versicherung werden einige Informationen zum Kunden sowie ähnlichen Cases abgefragt.
- Die angegebene Höhe des Schadens wird überprüft.
- Der Kunde wird noch einmal persönlich kontaktiert.

Um Einarbeitungszeiten zu minimieren, werden alle drei Aktivitäten für reale Cases in der Regel von einem Mitarbeiter ausgeführt, die Aktivitäten werden also sequentiell ausgeführt. Die Reihenfolge, in der die Aktivitäten ausgeführt werden, spielt jedoch keine Rolle, die Ausführung ist damit reihenfolgevariabel. Da der Kunde wahlweise telefonisch oder postalisch kontaktiert werden kann, ist die Ausführung darüber hinaus modusvariabel.

Die geschilderten Aktivitäten werden in Abbildung 5.24 durch vier Netzfragmente mit den Aktivitäts-Transitionen `Datenbank einsehen`, `Schadenshöhe überprüfen` sowie `Rückfrage Telefon` und `Rückfrage Post` modelliert. Der Startzustand ist durch die Stellen `Schadenshöhe nötig`, `Datenbank nötig`, `Rückfrage nötig`, der Zielzustand durch die Stellen `Datenbank fertig`, `Schadenshöhe fertig`, `Rückfrage fertig` gegeben. □

Würde man diesen Teil des Geschäftsprozesse in der herkömmlichen konstruktiven Weise modellieren, müßten alle sechs erlaubten Möglichkeiten, die drei Aktivitäten sequentiell auszuführen, explizit modelliert und als alternativ ausführbare Pfade in das Geschäftsprozeßmodell integriert werden. Zusätzlich müßte in jeden Pfad (beispielsweise mit Hilfe impliziter OR-Konstrukte) die Auswahlentscheidung bezüglich der beiden Modi der Kontaktaufnahme mit dem Kunden integriert werden<sup>23</sup>. Schon dieses kleine Beispiel zeigt damit recht deutlich die Grenzen der konstruktiven Modellierung von nicht-strukturierten Teilen von Geschäftsprozessen auf. Im Hinblick auf die Simulation nicht-strukturierter Teile von Geschäftsprozeßmodellen in Abschnitt 6.3.6 sei an dieser Stelle darauf hingewiesen, daß zu jeder deklarativen Spezifikation mittels nicht-strukturierter Segmente eine konstruktive Spezifikation erzeugt werden kann. Diese enthält sämtliche zulässigen Kombinationsmöglichkeiten der Netzfragmente als alternativ ausführbare Pfade.

### **5.6.3 Business Rules**

Bisher ist die Ausführung der mittels Netzfragmenten modellierten Aktivitäten nicht-strukturierter Teile in keiner Weise eingeschränkt. Für die Bearbeitung der Schadensmeldung in Abbildung 5.24 sind beispielsweise alle 12 Variationen der Ausführung möglich. In vielen Fällen ist es jedoch wünschenswert, gewisse Variationen bei der Ausführung ausschließen zu können. Im Rahmen des *VIPbusiness*-Ansatzes geschieht dies durch Angabe von *Geschäftsregeln* (üblicherweise als *Business Rules* [Her96, HK96] bezeichnet).

<sup>23</sup>Natürlich könnten an dieser Stelle auch 12 alternativ ausführbare Pfade (die sechs möglichen Reihenfolgen mit je einem möglichen Modus) verwendet werden.

Durch die Angabe von Business Rules können Bedingungen formuliert werden, die bei der Ausführung der Geschäftsprozeßmodelle stets erfüllt sein müssen. So kann beispielsweise die Reihenfolge der Ausführung von Aktivitäten beeinflusst werden, kann die Ausführung einer Aktivität als zwingend vorgeschrieben werden oder läßt sich die Ausführung einer Aktivität in gewissen Situationen verbieten.

Die Idee, das mögliche Verhalten eines Geschäftsprozeßmodells durch Angabe einer Menge von Business Rules einzuschränken, entspricht dem Konzept der *Röhre* aus Abschnitt 3.1.1, bei dem die Variabilität der Ausführung der mittels Netzfragmenten modellierten Aktivitäten durch den Rand der Röhre eingeschränkt wird. Die Röhre selbst ist durch die Menge der Business Rules gegeben. Innerhalb dieser Röhre kann die Bearbeitung von Cases durch die Ausführung der mittels Netzfragmenten modellierten Aktivitäten auf beliebige Art und Weise stattfinden.

Allgemein haben Business Rules die folgende Form:

**Definition 27 (Business Rule, Menge von Business Rules)**

Eine *Business Rule* ist ein zusammenhängendes Prädikate/Transitionen-Netz ohne Anfangsmarkierung  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  mit:

- $S_{br}$  ist eine nichtleere Menge von Stellen,
- $T_{br}$  besteht aus genau einem deklarativen Element des Typs Transition.

Die Menge von Business Rules eines nicht-strukturierten Segments  $ns$  wird mit  $BR_{ns}$  bezeichnet. □

Zentraler Bestandteil einer Business Rule ist das deklarative Element. Zur Modellierung deklarativer Elemente werden spezielle Transitionen verwendet. Im Rahmen des *VIP*-Projekts wurden auf Basis verschiedener Arten von speziellen Transitionen 5 Arten von Business Rules entwickelt, mit deren Hilfe das mögliche Verhalten modellierter Systeme [Des00, Fre01] und insbesondere das mögliche Verhalten nicht-strukturierter Teile von Geschäftsprozessen [DE00] eingeschränkt werden kann:

- Regeln auf Basis von Fakt-Transitionen,
- Regeln auf Basis von Ziel-Transitionen,
- Regeln auf Basis von Kausalketten-Transitionen mit Distanzen,
- Regeln auf Basis von Nina-Transitionen,
- Regeln auf Basis von ausgeschlossenen Transitionen.

In den folgenden Abschnitten wird erläutert, welche Bausteine für welche Regel zu benutzen sind und welche Anforderungen bezüglich des Verhaltens eines nicht-strukturierter

Segments mit der jeweiligen Regel formuliert werden können. Wie das mögliche Verhalten eines nicht-strukturierten Segments durch die dazugehörigen Business Rules im Rahmen der Ausführung oder Simulation des entsprechenden Geschäftsprozeßmodells eingeschränkt wird, ist Gegenstand der Ausführungen in den Abschnitten 6.3.6 und 7.1.

### Fakt-Transitionen

*Fakt-Transitionen* [Des00, Fre01, Rei86] werden benutzt, um Eigenschaften eines Geschäftsprozesses zu spezifizieren, die während der gesamten Bearbeitung eines Cases erfüllt sein sollen<sup>24</sup>. Fakt-Transitionen gehören zu der Gruppe der *Anfrage-Transitionen* [Des00, Fre01], sie dienen der Spezifikation gewünschter oder geforderter Eigenschaften eines modellierten Systems und haben keinen direkten Einfluß auf sein mögliches Verhalten. Durch Fakt-Transitionen spezifizierte Eigenschaften eines Geschäftsprozeß-Entwurfs gelten als erfüllt, falls die Fakt-Transitionen im zugehörigen Petrinetzmodell unter keiner erreichbaren Markierung aktiviert sind. Als graphischer Baustein wird für Fakt-Transitionen das übliche Transitions-Symbol, ergänzt durch ein stilisiertes *F* verwendet. Um zu verdeutlichen, daß Fakt-Transitionen und alle übrigen Anfrage-Transitionen das Verhalten des Geschäftsprozeßmodells nicht beeinflussen, werden die entsprechenden Kanten in *gestrichelter* Form dargestellt.

#### *Beispiel*

Durch die Fakt-Transition in Abbildung 5.25 (a) wird gefordert, daß bei nicht vorhandenem Versicherungsschutz (**kein Versicherungsschutz**) niemals eine Begleichung des Schadens (**Schaden begleichen**)vorgenommen werden darf.

□

### Ziel-Transitionen

Mit Hilfe von Zielen läßt sich fordern, daß ausgehend von einem spezifizierten Zustand 1 in jedem Fall irgendwann ein spezifizierter Zustand 2 erreicht wird. So kann beispielsweise gefordert werden, daß jede eingegangene Anfrage eines Kunden irgendwann einmal beantwortet sein muß (eine beliebige Anforderung z.B. bei *Call Centern*). Im Rahmen der Analyse eines Geschäftsprozeßmodells kann untersucht werden, ob eine Verfehlung von Zielen möglich ist (vgl. Abschnitt 7.1). Als graphischer Baustein wird für externe Transitionen das übliche Transitions-Symbol, ergänzt durch ein stilisiertes *Z* verwendet.

#### *Beispiel*

Durch die Ziel-Transition in Abbildung 5.25 (b) wird gefordert, daß nach Eingang eines Schadensfalls in jedem Fall irgendwann der Kunde kontaktiert wird.

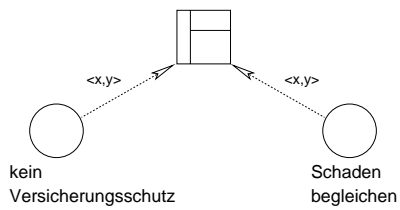
□

### Kausalketten-Transitionen

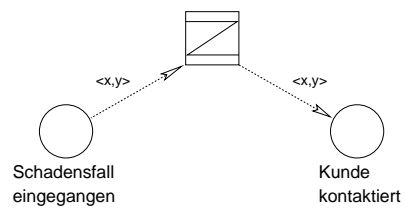
Mit Hilfe von *Kausalketten-Transitionen* lassen sich unmittelbare kausale Abhängigkeiten

---

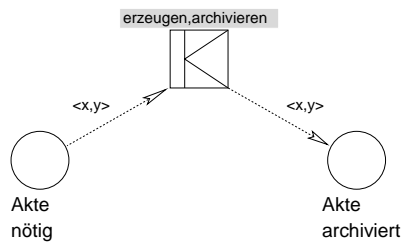
<sup>24</sup>Diese Eigenschaften werden auch als *invariante* Eigenschaften bezeichnet.



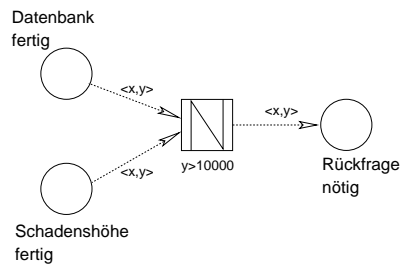
(a) Fakt-Transition.



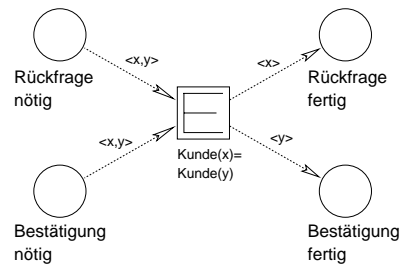
(b) Ziel-Transition.



(c) Kausalketten-Transition.



(d) Nina-Transition.



(e) Ausgeschlossene Transition.

Abbildung 5.25: Spezielle Transitionen zur Modellierung von Business Rules.

zwischen Aktivitäten untersuchen bzw. spezifizieren [Des00, Fre01]. Für zwei Aktivitäten  $a_1$  und  $a_2$  kann auf diese Weise angegeben werden, welche Sequenz von Aktivitäten nach  $a_1$  ausgeführt werden muß, bevor  $a_2$  ausgeführt wird. Alternativ kann durch Angabe einer *Distanz* [Des00] lediglich die Anzahl der Ausführungen von Aktivitäten (ohne explizite Angabe der Aktivitäten selbst) festgelegt werden, die zwischen den Ausführungen von  $a_1$  und  $a_2$  liegen müssen. Oft werden mit Hilfe von Kausalketten unerwünschte Sequenzen von Aktivitäten spezifiziert, beispielsweise die Vernichtung eines Dokuments unmittelbar nach seiner Erzeugung. Als graphischer Baustein wird für Kausalketten-Transitionen das übliche Transitions-Symbol, ergänzt durch ein stilisiertes  $K$  verwendet.

### **Beispiel**

Abbildung 5.25 (c) zeigt eine Kausalketten-Transition, durch die eine (unerwünschte) Sequenz von Aktivitäten (das Erzeugen und unmittelbare Archivieren einer Akte) spezifiziert wird.

□

### **Business Rules auf Basis von Nina-Transitionen**

Mit Hilfe von Nina-Transitionen kann die Anforderung formuliert werden, daß bei der Bearbeitung eines Case bestimmte Aktivitäten *nicht nach* anderen Aktivitäten ausgeführt werden dürfen. So kann beispielsweise gefordert werden, daß für einen Schadensfall mit großer Schadenshöhe zuerst der Kunde kontaktiert wird, bevor andere Aktivitäten der Bearbeitung begonnen werden. Als graphischer Baustein wird für Nina-Transitionen das übliche Transitions-Symbol, ergänzt durch ein stilisiertes  $N$ , verwendet. Die mit Hilfe von Nina-Transitionen formulierten Bedingungen beziehen sich ausschließlich auf die Bearbeitung einzelner Fälle.

### **Beispiel**

Durch die Nina-Transition in Abbildung 5.25 (d) wird gefordert, daß für Fälle mit einer angegebenen Schadenshöhe über 10.000 die Rücksprache mit dem Kunden stets als erste Aktivität zu erfolgen hat.

□

### **Business Rules auf Basis von ausgeschlossenen Transitionen**

Mit Hilfe von *ausgeschlossenen Transitionen* [Obe90, Vos87] lassen sich ebenfalls verbotene Reihenfolgen von Aktivitäten formulieren. Im Gegensatz zu Regeln auf Basis von Nina-Transitionen beziehen sich die formulierten Anforderungen dabei jedoch nicht nur auf die Bearbeitung einzelner Fälle sondern auf sämtliche im Rahmen der Ausführung des Geschäftsprozeßmodells ausgeführte Aktivitäten. Auf diese Weise kann beispielsweise sichergestellt werden, daß die Bearbeitung zweier Schadensmeldungen desselben Kunden stets in der Reihenfolge des Eingangs der Schadensmeldungen erfolgt. So wird verhindert, daß mit einem Kunde Rücksprache bezüglich der Bearbeitung der später eingegangenen Schadensmeldung genommen wird, bevor dies für die erste Schadensmeldung geschehen ist. Als graphischer Baustein wird für Kausalketten-Transitionen das übliche Transitions-Symbol, ergänzt durch ein stilisiertes  $E$  verwendet.

### **Beispiel**

Durch die ausgeschlossene Transition in Abbildung 5.25 (e) wird sichergestellt, daß für den Fall zweier Schadensmeldungen durch denselben Kunden zuerst bezüglich der Schadensmeldung Rücksprache genommen wird, deren Eingang als erstes bestätigt wurde. □

Die graphische Darstellung der beschriebenen Business Rules für ein nicht-strukturiertes Segment erfolgt durch Eintrag der entsprechenden Regeln in das Schema aus Abbildung 5.23.

### **Beispiel**

In Abbildung 5.26 ist das nicht-strukturierte Segment aus Abbildung 5.24 um eine Business Rule auf Basis einer Nina-Transition erweitert worden. Durch diese Regel wird sichergestellt, daß für Fälle mit einer Schadenshöhe über 10.000 als erste Aktivität eine Rücksprache mit dem Kunden erfolgt. □

Mit der Einführung von Business Rules sind alle Bausteine zur Modellierung von nicht-strukturierten Teilen von Geschäftsprozessen vorhanden (vgl. Abbildung 5.23). Damit kann der Begriff des *nicht-strukturierten Segments* folgendermaßen festgelegt werden:

#### **Definition 28 (Nicht-strukturiertes Segment)**

Ein *nicht-strukturiertes Segment*  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  eines Geschäftsprozesses setzt sich zusammen aus

- einem Startzustand  $S_{0_{ns}}$ ,
- einer Menge  $NF_{ns}$  von Netzfragmenten,
- einer Menge  $R_{ns}$  von Ressourcen-Stellen,
- einer Menge  $BR_{ns}$  von Business Rules,
- einem Zielzustand  $S_{Z_{ns}}$ .

### **Beispiel**

Abbildung 5.26 enthält die Bearbeitung der Schadensmeldung als nicht-strukturiertes Segment. □

## **5.6.4 Hybride Spezifikationen**

Mit den in den vorangegangenen Abschnitten eingeführten Bausteinen lassen sich nicht-strukturierte Teile von Geschäftsprozessen auf übersichtliche Weise modellieren. Die meisten realen Geschäftsprozesse beinhalten jedoch sowohl strukturierte als auch nicht-strukturierte Teile (vgl. Abschnitt 3.1.1). Entsprechend müssen auch in Modellen von Geschäftsprozessen nicht-strukturierte mit strukturierten Teilen kombiniert werden. Dies geschieht in Form von *hybriden Spezifikationen* [DE00]:



### Definition 29 (Hybride Spezifikation)

Eine *hybride Spezifikation* eines Geschäftsprozesses setzt sich zusammen aus

- einem Prädikate/Transitionen-Netz mit Aktivitäts-Transitionen  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  zur Modellierung der strukturierten Teile des Geschäftsprozesses,
- einer Menge  $NS$  von nicht-strukturierten Segmenten  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$ .

Dabei sind alle Stellen des Start- und Zielzustands eines jeden nicht-strukturierten Segments auch in der Menge der Stellen von  $N_{TA}$  enthalten:

Für jedes nicht-strukturierte Segment  $ns \in NS$  gilt :  $S_{0_{ns}} \subseteq S$  und  $S_{Z_{ns}} \subseteq S$ .

Gleiches gilt für die Ressourcen-Stellen.

□

### Beispiel

Bei dem Beispiels-Geschäftsprozeßmodell zur Definition der *VIPbusiness*-Netzklasse in Abbildung 5.26 handelt es sich um eine hybride Spezifikation. Sehr deutlich wird in diesem Modell die Schnittstellen-Funktion der Stellen der Start- und Zielzustände von nicht-strukturierten Teilen.

Aus Gründen der Übersichtlichkeit wurde die Ressourcen-Stelle für Mitarbeiter MA, die sowohl von Aktivitäten innerhalb als auch außerhalb des nicht-strukturierten Teils benötigt wird, für den nicht-strukturierten Teil dupliziert. Es handelt sich formal jedoch um die gleiche Stelle. Durch dieses Vorgehen kann vermieden werden, daß die Kanten zwischen Aktivitäten und Ressourcen-Stellen durch das gesamte Modell gezogen werden müssen.

□

## 5.7 *VIPbusiness*-Netze

Als Ergebnis der Kapitel 4 und 5 sollen an dieser Stelle noch einmal alle eingeführten Bausteine und Konzepte in einer Definition der Klasse der *VIPbusiness*-Netze zusammengefaßt werden:

### Definition 30 (*VIPbusiness*-Netz)

Ein *VIPbusiness-Netz* ist ein Prädikate/Transitionen-Netz  $N = (S, T, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  mit den folgenden Erweiterungen:

- eine Menge von Aktivitäts-Transitionen  $T_A$ ,
- eine Menge von Routing-Konstrukten  $\{T_{AS}, T_{AJ}, T_{OS}, T_{OJ}\}$ ,
- eine Menge von Ressourcen-Stellen  $R \subset S$ ,
- ein Zeitkonzept gemäß Definition 22,
- ein Kostenkonzept gemäß Definition 23,

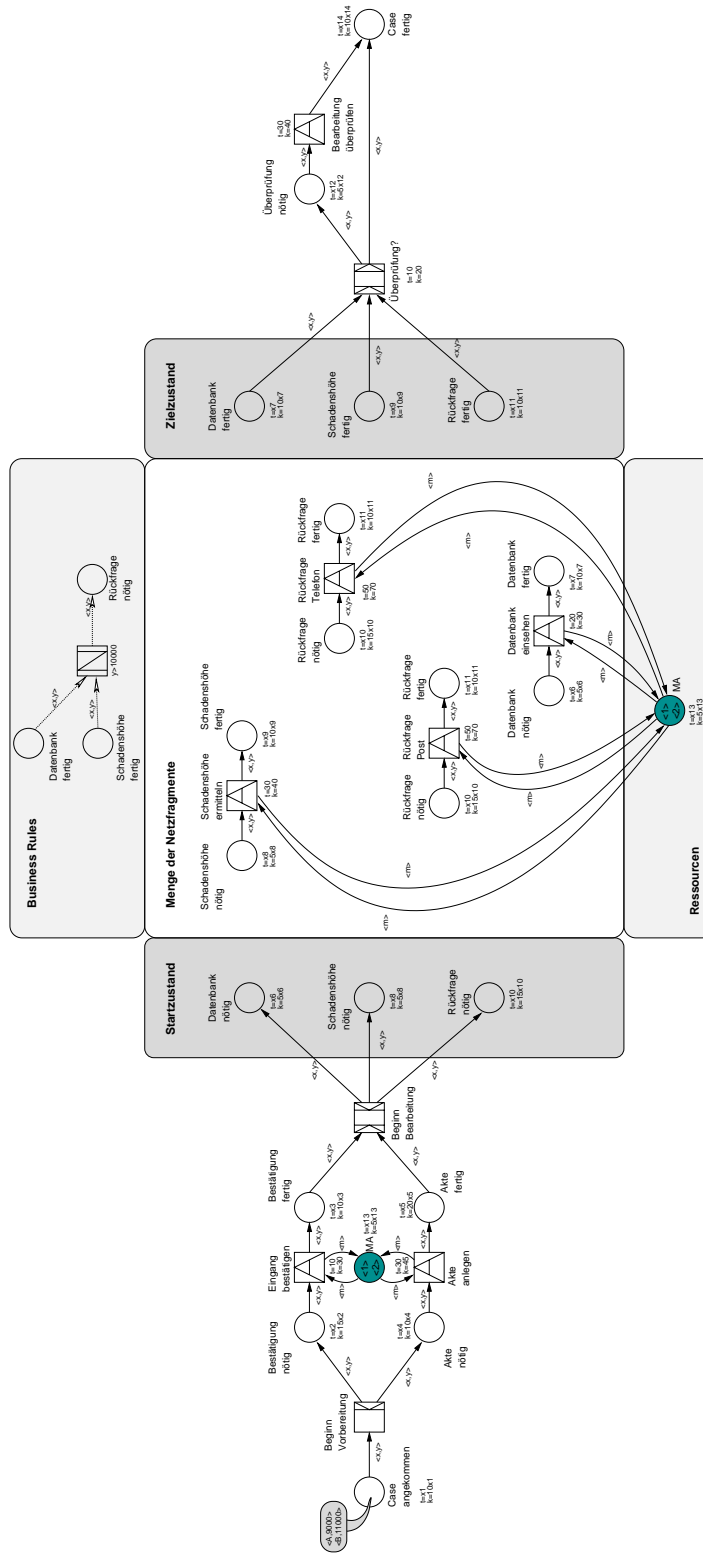


Abbildung 5.26: Geschäftsprozessmodell als *VIPbusiness*-Netz mit allen Erweiterungen.

- eine Menge  $NS$  von nicht-strukturierten Segmenten.

□

Sofern sie eine Menge nicht-strukturierter Segmente besitzen, handelt es sich bei *VIPbusiness*-Netzen damit um hybride Spezifikationen gemäß Definition 29.

### ***Beispiel***

Abbildung 5.26 zeigt ein Geschäftsprozeßmodell mit allen Bausteinen, die im Rahmen der *VIPbusiness*-Erweiterungen eingeführt wurden.

□

Mit der Erstellung von Geschäftsprozeßmodellen mit Hilfe von *VIPbusiness*-Netzen ist der erste Schritt des 3-stufigen *VIPbusiness*-Ansatzes abgeschlossen. Im nächsten Kapitel beschäftigen wir uns mit dem möglichen Verhalten der erstellten Geschäftsprozeßmodelle im Rahmen von *Simulation*.



# Kapitel 6

## Simulation

Nachdem mit Hilfe der in den beiden vorangegangenen Kapitel eingeführten Bausteine Modelle von Geschäftsprozessen erstellt werden können, wird in diesem Kapitel die Beschreibung des Verhaltens dieser Geschäftsprozesse im Rahmen einer Simulation der erstellten Modelle behandelt. Damit werden in diesem Kapitel die Konzepte eingeführt, die zur Umsetzung von *Schritt 2* des 3-stufigen *VIPbusiness*-Ansatzes notwendig sind (vgl. Abbildung 6.1).

In Abschnitt 6.1 wird zunächst allgemein auf Möglichkeiten zur Beschreibung des Verhaltens von Petrinetzmodellen eingegangen. Dabei wird zwischen der Darstellung der *sequentiellen* Semantik durch Schaltfolgen und der Darstellung der *kausalen* Semantik durch halbgeordnete Abläufe unterschieden. Beide Sichtweisen werden in den Abschnitten 6.1.1 und 6.1.2 erläutert. Die Vorteile der Verwendung halbgeordneter Abläufe zur Beschreibung des Verhaltens der erstellten Geschäftsprozeßmodelle werden geschildert.

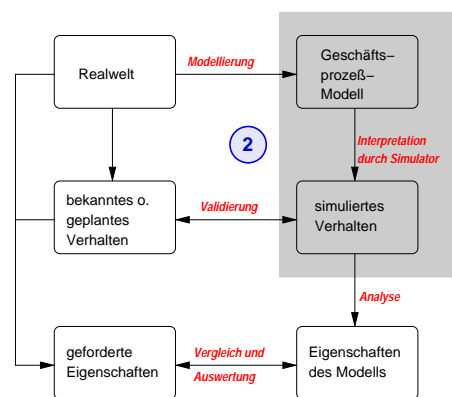


Abbildung 6.1: Schritt 2: Simulation.

Für Geschäftsprozesse aus der Praxis ist es in der Regel nicht möglich, das gesamte mögliche Verhalten des entsprechenden Geschäftsprozeßmodells zu beschreiben. Aus diesem Grund wird in Abschnitt 6.2 auf das Simulationskonzept *VIPsim* des *VIP*-Projekts und die dabei entwickelten Methoden zur Generierung einer (Teil-)Menge aller halbgeordneten Abläufe eines Petrinetzmodells eingegangen.

In Analogie zu der Modifikation der *VIPmodel*-Netzklasse durch die *VIPbusiness*-Erweiterungen in Kapitel 5 wird in Abschnitt 6.3 das *VIPsim*-Konzept mit entsprechenden *VIPbusiness*-Erweiterungen versehen.

## 6.1 Beschreibung des Verhaltens von Petrinetzen

Ein großer Vorteil bei der Verwendung von Petrinetzen zur Geschäftsprozeßmodellierung liegt in der *Ausführbarkeit* der erstellten Modelle (vgl. [Aal98d] oder [Zel96]). In den Abschnitten 4.1 und 4.2 wurde bereits erläutert, wie Zustände von Petrinetzen durch Markierungen beschrieben werden können, die sich durch das Schalten von Transitionen verändern lassen. In den folgenden Abschnitten werden darauf aufbauend Konzepte zur Beschreibung des dynamischen Verhaltens von Petrinetzen vorgestellt. Traditionell wird dabei zwischen zwei Sichtweisen unterschieden (vgl. [DFO97] oder [Des00]): Abläufe können als Sequenzen zeitlich nacheinander geschehender Ereignisse bzw. Ereignismengen in Form von *Schalt-* oder *Schrittfolgen* oder als kausal halbgeordnete Mengen von Ereignissen in Form von *halbgeordneten Abläufen* dargestellt werden.

### 6.1.1 Schaltfolgen und sequentielle Semantik

Im Bereich der Anwendungen von Petrinetzen werden fast ausschließlich *Schaltfolgen* zur Beschreibung des dynamischen Verhaltens von Petrinetzen verwendet [DFO97]:

#### Definition 31 (Schaltfolge)

Sei  $N = (S, T, F, D, m, \mathcal{P}, \Phi)$  ein Prädikate/Transitionen-Netz mit einer Markierung  $m$ .

Falls  $m \xrightarrow{\tau_1, p_{\tau_1}} m_1, m_1 \xrightarrow{\tau_2, p_{\tau_2}} m_2, \dots, m_{n-1} \xrightarrow{\tau_n, p_{\tau_n}} m_n$  Schaltvorgänge sind, ist

$$\omega = (\tau_1, p_{\tau_1}) (\tau_2, p_{\tau_2}) \dots (\tau_n, p_{\tau_n})$$

eine *Schaltfolge* von  $m$  nach  $m_n$  (Schreibweise:  $m \xrightarrow{\omega} m_n$ ). Dies gilt auch für die leere Sequenz  $\varepsilon : m \xrightarrow{\varepsilon} m$  für jede Markierung  $m$ .

Eine Markierung  $m'$  ist von  $m$  *erreichbar*, wenn  $m \xrightarrow{\omega} m'$  für irgendeine Schaltfolge  $\omega$  gilt.

□

Das Verhalten eines markierten Petrinetzes kann durch die Menge aller möglichen Schaltfolgen beschrieben werden. Die Menge der Markierungen, die durch Ausführung der Schaltfolgen erreichbar ist, wird auch als *Erreichbarkeitsmenge* eines Systems bezeichnet [Bau96]. Weder die Menge der Schaltfolgen noch die Menge der erreichbaren Markierungen ist notwendigerweise endlich (vgl. [Bau96] oder [Rei86]). Durch diese Tatsache wird der Nachweis von Systemeigenschaften, die mit dem Eintritt oder der Unerreichbarkeit eines Zustandes zusammenhängen, erheblich erschwert [DO95].

Im Rahmen dieser Arbeit wird davon ausgegangen, daß jeder Geschäftsprozeß entweder selbst terminiert oder ohne Informationsverlust für die Analyse terminiert werden kann (vgl. hierzu auch die Ausführungen in Abschnitt 6.2). Es werden also nur endliche Schaltfolgen und somit endliche Erreichbarkeitsmengen berücksichtigt.

Um die Erreichbarkeitsmenge eines markierten Petrinetzes zu ermitteln, werden systematisch alle Schaltmöglichkeiten geprüft und in einer Erreichbarkeitstabelle dargestellt.

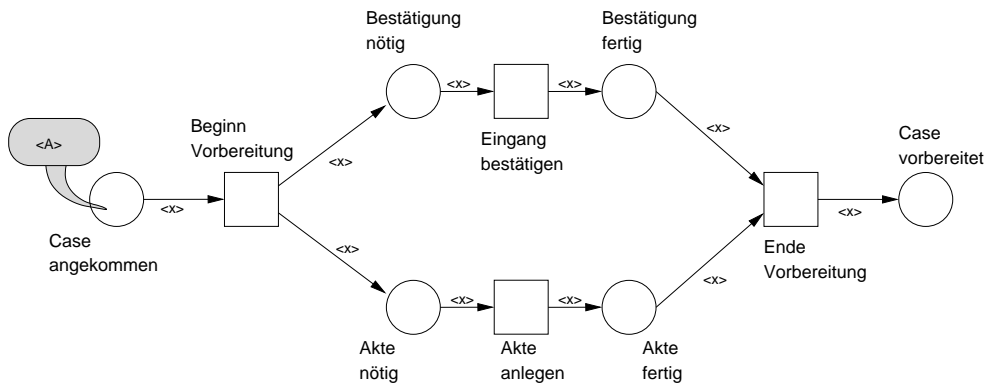


Abbildung 6.2: Geschäftsprozeßmodell aus Abbildung 4.5 mit vereinfachten Marken.

### Beispiel

Abbildung 6.2 zeigt eine Variation des Geschäftsprozeßmodells aus Abbildung 4.5<sup>1</sup>. Mit der abgebildeten Anfangsmarkierung  $m_0$  ergibt sich die folgende Erreichbarkeitstabelle:

$m$	$m(Ca)$	$m(Bn)$	$m(Bf)$	$m(An)$	$m(Af)$	$m(Cv)$	$\tau, p_\tau$	$m'$
$m_0$	{<A>}						BV, $x \leftarrow A$	$m_1$
$m_1$		{<A>}		{<A>}			Eb, $x \leftarrow A$ Aa, $x \leftarrow A$	$m_2$ $m_3$
$m_2$			{<A>}	{<A>}			Aa, $x \leftarrow A$	$m_4$
$m_3$		{<A>}			{<A>}		Eb, $x \leftarrow A$	$m_4$
$m_4$			{<A>}		{<A>}		EV, $x \leftarrow A$	$m_5$
$m_5$						{<A>}		

Tabelle 6.1: Erreichbarkeitstabelle für das Geschäftsprozeßmodell in Abbildung 6.2.

In jeder Zeile der Erreichbarkeitstabelle wird zunächst der Name einer Markierung  $m$  aufgeführt, gefolgt von den Markierungen der einzelnen Stellen. Die vorletzte Spalte gibt an, welche der unter  $m$  aktivierten Transitionen in welchem Modus schalten kann, um die

<sup>1</sup>Um die folgenden Beispiele ein wenig übersichtlicher gestalten zu können, werden die Namen der Stellen und Transitionen des Beispiel-Modells wie folgt abgekürzt:

- *Stellen:*  
Case angekommen ( $Ca$ ), Bestätigung nötig ( $Bn$ ), Bestätigung fertig ( $Bf$ ), Akte nötig ( $An$ ), Akte fertig ( $Af$ ), Case vorbereitet ( $Cv$ ),
- *Transitionen:*  
Beginn Vorbereitung ( $BV$ ), Eingang bestätigen ( $Eb$ ), Akte anlegen ( $Aa$ ), Ende Vorbereitung ( $EV$ ).

in der letzten Spalte angegebene Folgemarkierung zu erreichen. Für jede der von  $m$  aus erreichbaren Folgemarkierungen wird eine neue Zeile in der Erreichbarkeitstabelle angelegt. Ist für eine Markierung  $m$  in der rechten Spalte keine Folgemarkierung eingetragen (im Beispiel ist dies für  $m_5$  der Fall), so ist unter  $m$  keine Transition aktiviert und somit keine weitere Folgemarkierung erreichbar. In diesem Fall wird  $m$  als *tote* Markierung bezeichnet, der Zustand des modellierten Systems als *Deadlock*.

Für das Geschäftsprozeß-Beispiel existieren ausgehend von  $m_0$  fünf erreichbare Markierungen. Die Erreichbarkeitsmenge ist also endlich (vgl. Tabelle 6.2).

Schaltfolge	Transitionen und Schaltmodi			
SF 1	BV, $x \leftarrow A$	Eb, $x \leftarrow A$	Aa, $x \leftarrow A$	EV, $x \leftarrow A$
SF 2	BV, $x \leftarrow A$	Aa, $x \leftarrow A$	Eb, $x \leftarrow A$	EV, $x \leftarrow A$

Tabelle 6.2: Mögliche Schaltfolgen für das Geschäftsprozeßmodell in Abbildung 6.2.

□

Durch Schaltfolgen wird die *sequentielle* Semantik eines Petrinetzes beschrieben (vgl. beispielsweise [DFO97] oder [Des00]). Für die Beschreibung des Verhaltens von Geschäftsprozessen bedeutet dies, daß die jeweilige Menge der Aktivitäten, die zur Bearbeitung eines Case auszuführen sind, als Sequenz von Transitionen dargestellt wird. Ein Vorteil dieses Ansatzes liegt in der vergleichsweise einfachen Definition sequentieller Semantik sowie der intuitiv einsichtigen Interpretation des Systemverhaltens als geordnete Menge der Übergänge zwischen globalen Zuständen (vgl. [DFO97] oder [Fre01]).

Schwierigkeiten bereitet diese Vorgehensweise jedoch bei dem Versuch, die Ausführung eines Geschäftsprozesses zu beschreiben, bei der Aktivitäten unabhängig voneinander (auch: *parallel*, siehe dazu die Ausführungen in Abschnitt 5.2) ausgeführt werden können. Bei der Beschreibung mittels Schaltfolgen müssen auch diese unabhängigen Aktivitäten sequentiell geordnet werden. Dies hat zur Folge, daß das Systemverhalten durch eine Schaltfolge in diesem Fall nicht mehr korrekt dargestellt werden kann.

### **Beispiel**

Für das Geschäftsprozeßmodell in Abbildung 6.2 existieren zwei Schaltfolgen (vgl. Tabelle 6.2), durch die nur auf den ersten Blick unterschiedliche Ausführungen des Geschäftsprozeßmodells beschrieben werden. Tatsächlich wird jedoch durch beide Schaltfolgen dieselbe Bearbeitung des Falls A beschrieben. Die beiden Schaltfolgen sind das Ergebnis des Versuchs, die unabhängig voneinander ausführbaren Aktivitäten **Eingang bestätigen** und **Akte anlegen** in eine sequentielle Ordnung zu bringen.

□

Bei höheren Petrinetzen wird dieser Effekt durch die Unterscheidbarkeit der Marken noch verstärkt, da Transitionen grundsätzlich in unterschiedlichen Modi schalten können. Damit



kann die Transition für unterschiedliche Modi quasi parallel schalten, die Verfügbarkeit eventuell benötigter Ressourcen vorausgesetzt. Diese Art der unabhängigen Ausführung der gleichen Aktivität für unterschiedliche Schaltmodi kann mit Schaltfolgen nicht explizit ausgedrückt werden.

### **Beispiel**

Angenommen, die Anfangsmarkierung des Geschäftsprozeßmodells in Abbildung 6.2 würde die Markierung  $m_0(\text{Case angekommen}) = \{ \langle A \rangle, \langle B \rangle \}$  beinhalten, es sind also statt einem nun zwei Cases, A und B, zu bearbeiten. Da in dem Geschäftsprozeßmodell keine Ressourcen berücksichtigt werden, können die beiden Cases unabhängig voneinander bearbeitet werden. Um die Bearbeitung beider Cases mit Hilfe der sequentiellen Semantik beschreiben zu können, benötigen wir 280 Schaltfolgen. Dies entspricht der Anzahl der Möglichkeiten, die unabhängige Ausführung der Aktivitäten für die beiden Cases in eine sequentielle Ordnung zu bringen<sup>2</sup>. Tatsächlich wird jedoch durch alle Schaltfolgen dasselbe Verhalten beschrieben. □

Für größere Geschäftsprozeßmodelle mit vielen Aktivitäten, die unabhängig voneinander ausgeführt werden können, sind Schaltfolgen nur noch bedingt geeignet, das mögliche Verhalten eines Geschäftsprozeßmodells zu beschreiben. Geübte Benutzer werden zwar in der Lage sein, beispielsweise durch einen Vergleich der Schaltfolgen in Tabelle 6.2 zu folgern, daß die Schaltfolgen unabhängig voneinander ausführbare Aktivitäten enthalten. Für größere Modelle wird dies jedoch nahezu unmöglich. Dazu kommt der Nachteil, daß Schaltfolgen wie Erreichbarkeitstabellen über keine graphische Repräsentation verfügen. Das Verhalten des Geschäftsprozeßmodells wird somit auf sehr abstrakte Weise dargestellt und ist damit für Benutzer nur schwer interpretierbar. Für die Validierung des möglichen Verhaltens von Geschäftsprozeßmodellen zur Überprüfung der Korrektheit des Geschäftsprozeß-Entwurfs im Sinne der Anwendung sind Schaltfolgen daher nicht besonders gut geeignet.

Für Geschäftsprozeßmodelle, bei deren Ausführung Aktivitäten unabhängig voneinander stattfinden können, ist also eine Semantik wünschenswert, durch die bei der Beschreibung des möglichen Verhaltens eigentlich unabhängige Aktivitäten unmittelbar als solche dargestellt werden können. Zudem sollte es möglich sein, das Verhalten der Geschäftsprozeßmodelle in einer Art und Weise darzustellen, die anschaulich und somit auch für ungeübte Benutzer leicht verständlich ist. Im nächsten Abschnitt werden daher halbgeordnete Abläufe zur Beschreibung des Verhaltens von Geschäftsprozeßmodellen eingeführt.

## **6.1.2 Halbgeordnete Abläufe und kausale Semantik**

Im vorherigen Abschnitt wurde festgestellt, daß es in einem Geschäftsprozeßmodell Aktivitäten gibt, die in einer zwingenden Reihenfolge ausgeführt werden müssen, und solche, die unabhängig voneinander ausgeführt werden können. Erstere werden als *kausal*

---

<sup>2</sup>Allgemein gibt es  $\frac{(n+m)!}{n!m!}$  Möglichkeiten, zwei nebenläufige Sequenzen der Längen  $n$  und  $m$  zu einer Schaltfolge zu kombinieren. Daraus ergeben sich für unser Beispiel 70 Schaltfolgen. Da für die beiden Cases aufgrund der Nebenläufigkeit der Aktivitäten *Eingang bestätigen* und *Akte anlegen* je 2 Sequenzen existieren, vergrößert sich die Zahl der Schaltfolgen um den Faktor 4.

geordnet, letztere als *kausal unabhängig* oder *nebenläufig* bezeichnet. Die Menge der Aktivitäten eines Geschäftsprozeßmodells enthält also geordnete und ungeordnete Elemente. Die Reihenfolgebeziehungen der Aktivitäten in Geschäftsprozeßmodellen bilden somit eine *Halbordnung*:

**Definition 32 (Halbordnung)**

Sei  $D$  eine Menge. Eine Relation  $< \subseteq D \times D$  heißt *Halbordnung*  $(D, <)$  genau dann, wenn für alle  $a, b \in D$  gilt:

- $\neg(a < a)$  ( $<$  ist irreflexiv),
- $a < b \wedge b < c \Rightarrow a < c$  ( $<$  ist transitiv).

Sei  $(D, <)$  eine Halbordnung. Mit  $x, y, z \in D$  lassen sich folgende Relationen herleiten:

- $x \leq y: (x < y) \vee (x \text{ id}_D y)$ ;  $\leq$  ist reflexiv, transitiv und antisymmetrisch.
- $x \prec y: (x < y) \wedge (\nexists z : x < z < y)$ ;  $\prec$  ist die unmittelbare Vorgängerrelation.
- $x \bar{k} y: (x \leq y) \vee (y \leq x)$ ;  $x, y$  liegen auf einer Linie.
- $x \text{ co } y: \neg(x < y) \wedge \neg(y < x)$ ;  $x, y$  sind ungeordnet oder identisch (auch: *concurrent* oder *nebenläufig*).

□

Im vorangegangenen Abschnitt wurde bereits auf die Schwierigkeiten eingegangen, die entstehen, wenn das Verhalten von Geschäftsprozessen mit kausal ungeordneten Aktivitäten durch Schaltfolgen und die daraus resultierende sequentielle Semantik beschrieben werden soll. Aus diesem Grund werden in diesem Abschnitt *halbgeordnete Abläufe* zur Beschreibung der kausalen Semantik von Geschäftsprozeßmodellen eingeführt.

Halbgeordnete Abläufe von markierten Petrinetzen können wiederum mit Hilfe einer speziellen Klasse von Petrinetzen, den *Kausalnetzen*, beschrieben und graphisch dargestellt werden. Kausalnetze enthalten *Bedingungen* (graphisch wie Stellen durch Kreise repräsentiert) zur Darstellung von Marken auf Stellen des markierten Petrinetzes, *Ereignisse* (graphisch durch das übliche Transitions-Symbol repräsentiert) zur Darstellung des Schaltens von Transitionen sowie *kausale Abhängigkeiten* (mittels gerichteter Pfeile dargestellt) zwischen Bedingungen und Ereignissen.

**Definition 33 (Kausalnetz)**

Ein Netz  $(B, E, K)$  heißt *Kausalnetz*, wenn es

- *azyklisch* ist ( $K^+$  ist irreflexiv und damit Halbordnung),
- Stellen *unverzweigt* sind ( $|\bullet b|, |b\bullet| \leq 1$  für jede Stelle  $b \in B$ ),
- weder *Ein-* noch *Ausgabeereignisse* existieren ( $|\bullet e|, |e\bullet| \geq 1$  für jede Transition  $e \in E$ ).

□

Mit Hilfe von Kausalnetzen kann jetzt das dynamische Verhalten eines mit Prädikate/Transitionen-Netzen<sup>3</sup> erstellten Geschäftsprozeßmodells durch *halbgeordnete Abläufe* beschrieben werden:

**Definition 34 (Halbgeordneter Ablauf)**

Sei  $N = (S, T, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  ein markiertes Prädikate/Transitionen-Netz. Sei  $(B, E, K)$  ein Kausalnetz sowie  $\alpha : B \rightarrow S$  eine Abbildung der Bedingungen des Kausalnetzes auf Stellen von  $N$  und  $\beta : E \rightarrow T$  eine Abbildung der Ereignisse des Kausalnetzes auf Transitionen von  $N$ .

Sei  $\lambda$  eine Funktion, die Bedingungen und Ereignissen des Kausalnetzes eine *Beschriftung*<sup>4</sup> zuweist. Für eine Bedingung  $b$  ist  $\lambda(b)$  eine Marke  $d \in \mathcal{D}_{\alpha(b)}$  in der zugehörigen Stelle  $\alpha(b)$ , für ein Ereignis  $e$  ist  $\lambda(e)$  ein Schaltmodus  $p_{\beta(e)}$  der korrespondierenden Transition  $\beta(e)$ .

Durch  $(B, E, K, \alpha, \beta, \lambda)$  wird ein *halbgeordneter Ablauf* des markierten Prädikate/Transitionen-Netzes beschrieben, wenn gilt:

- Für jede Marke der Anfangsmarkierung  $m_0$  von  $N$  existiert im Kausalnetz genau eine entsprechend beschriftete Bedingung ohne Vorbereich<sup>5</sup>:

$$\text{Für alle } s \in S : \text{für alle } d \in \mathcal{D}_s : \mu_{0,s}(d) = | \{b \in B \mid \alpha(b) = s, \bullet b = \emptyset, \lambda(b) = d\} | .$$

- Der Vorbereich eines jeden Ereignisses  $e$  im Kausalnetz wird durch  $\alpha$  auf den Vorbereich der jeweils korrespondierenden Transition  $\beta(e)$  des Systemnetzes abgebildet:

$$\text{Für alle } e \in E : \alpha(\bullet e) = \bullet \beta(e).$$

Dieses gilt entsprechend für den Nachbereich von Ereignissen:

$$\text{Für alle } e \in E : \alpha(e \bullet) = \beta(e) \bullet .$$

- Die Bedingungen im Vorbereich eines Ereignisses  $e$  entsprechen in Anzahl und Beschriftung genau denjenigen Marken in  $N$ , welche durch das Schalten der Transition  $\beta(e) = \tau$  im Modus  $\lambda(e) = p_\tau$  gemäß Kantenbeschriftung konsumiert werden<sup>6</sup>:

Für alle  $e \in E$  mit  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

$$\text{für alle } s \in \bullet \tau : \text{für alle } d \in \mathcal{D}_s : [\varphi_{s,\tau}(p_\tau)](d) = | \{b \in B \mid b \in \bullet e, \lambda(b) = d\} | .$$

Entsprechendes gilt für den Nachbereich von Ereignissen und die durch das Schalten von  $\tau$  produzierten Marken:

Für alle  $e \in E$  mit  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

$$\text{für alle } s \in \tau \bullet : \text{für alle } d \in \mathcal{D}_s : [\varphi_{s,\tau}(p_\tau)](d) = | \{b \in B \mid b \in e \bullet, \lambda(b) = d\} | .$$

---

<sup>3</sup>An dieser Stelle benutzen wir Prädikate/Transitionen-Netze ohne die in Kapitel 5 eingeführten *VIP-business*-Erweiterungen. Die Auswirkungen dieser Erweiterungen auf Abläufe werden erst in Abschnitt 6.3 behandelt.

<sup>4</sup>auch: *Label*.

<sup>5</sup>An dieser Stelle sei daran erinnert, daß Stellen durch Markierungen Multimengen zugeordnet werden. Es ist somit  $m(s) = \mu_s$ . Entsprechend sind die Schreibweisen  $m_0(s)$  und  $\mu_{0,s}$  äquivalent.

<sup>6</sup>*Erinnerung*: Aus der Belegung der Variablen in den Kantenbeschriftungen gemäß dem Schaltmodus ergeben sich Multimengen; ihr Definitionsbereich ist gleich der Grundmenge  $\mathcal{D}_s$  der zugehörigen Ein- bzw. Ausgabestelle.

- Falls für  $\tau$  eine Schaltbedingung  $guard(\tau)$  existiert, muß  $guard(\tau)$  für die Belegung der Variablen gemäß  $p_\tau$  wahr sein.

□

Die kausale Semantik eines markierten Petrinetzes ist durch die Menge der halbgeordneten Abläufe gegeben.

Für Entwürfe von Geschäftsprozessen bedeutet dies:

- Das Verhalten des modellierten Geschäftsprozesses wird durch die Menge der halbgeordneten Abläufe des Petrinetzmodells beschrieben.
- In den halbgeordneten Abläufen ist erkennbar, ob Aktivitäten sequentiell oder unabhängig voneinander ausgeführt worden sind.

In dieser Arbeit werden ausschließlich endliche Abläufe von Geschäftsprozeßmodellen betrachtet.

Bei der Erzeugung halbgeordneter Abläufe spielen Auswahlentscheidungen eine entscheidende Rolle. Soll das gesamte mögliche Verhalten eines Systems mittels halbgeordneter Abläufe erfaßt werden, so muß bei jeder Auswahlentscheidung für jede Alternative ein neuer Ablauf generiert werden. Die Anzahl der Abläufe eines Geschäftsprozeßmodells wächst also mit der Zahl der Auswahlentscheidungen.

### **Beispiel (Fortsetzung)**

Abbildung 6.3 zeigt die Bearbeitung zweier Cases A und B als halbgeordneten Ablauf des Geschäftsprozeßmodells aus Abbildung 6.2 (mit  $m_0(\text{Case angekommen}) = \{ \langle A \rangle, \langle B \rangle \}$  als Anfangsmarkierung). Um das Beispiel etwas übersichtlicher gestalten zu können, wurden die Abkürzungen aus Tabelle 6.1.1 übernommen. Die den Bedingungen zugeordneten Marken sowie die den Ereignissen zugeordneten Schaltmodi werden als Ergänzung der Anschrift der entsprechenden graphischen Symbole angegeben. So entspricht  $\text{Ca}(\mathbf{A})$  einer Marke vom Typ A auf der Stelle **Case angekommen**. Entsprechend bezeichnet  $\text{BV}(\mathbf{A})$  das Schalten der Transition im Modus  $x \leftarrow A$ . Da bei der Bearbeitung der Cases keine Auswahlentscheidungen zu treffen sowie keine beschränkten Ressourcen zu berücksichtigen sind, gibt es nur diesen einen halbgeordneten Ablauf<sup>7</sup>. Im vorangegangenen Abschnitt wurde gezeigt, daß für die Beschreibung des gleichen Verhaltens mittels sequentieller Semantik 280 Schaltfolgen nötig sind.

□

Die Beschreibung des Verhaltens von markierten Petrinetzen mittels halbgeordneter Abläufe ist mit einer Reihe von *Vorteilen* (gegenüber Schaltfolgen) verbunden (vgl. [DO95], [DFO97], [Des00], [Fre96], [Fre01]):

- Die Beschreibung des möglichen Verhaltens von Geschäftsprozeßmodellen mittels halbgeordneter Abläufe ist *effizient*: ein nebenläufiger Ablauf entspricht in der Regel einer Menge von Schaltfolgen.

<sup>7</sup>Ein Beispiel mit mehreren Auswahlentscheidungen wird in Abschnitt 7.4.2 eingeführt.

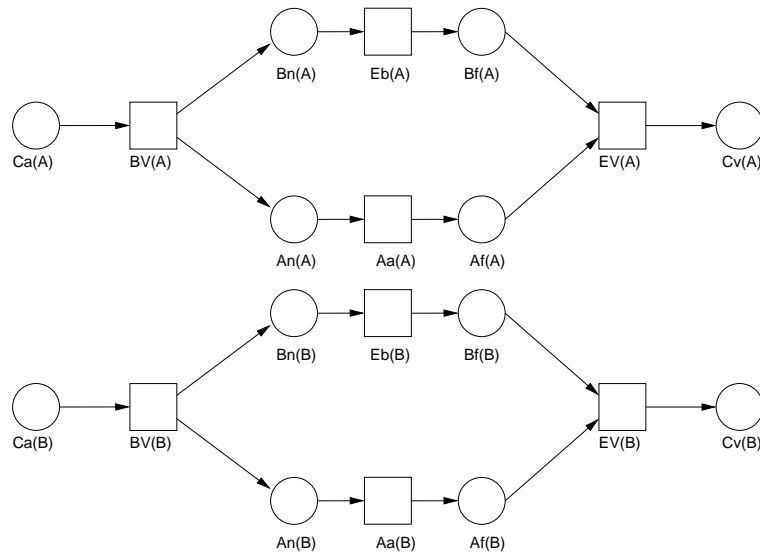


Abbildung 6.3: Ein halbgeordneter Ablauf des Geschäftsprozeßmodells aus Abbildung 6.2.

- Die unabhängige Ausführung von Aktivitäten kann unmittelbar ausgedrückt werden.
- Durch die Verwendung von Kausalnetzen zur graphischen Repräsentation halbgeordneter Abläufe läßt sich das mögliche Verhalten eines Geschäftsprozeßmodells sehr anschaulich darstellen. Halbgeordnete Abläufe eignen sich somit zur Validierung des Verhaltens der entworfenen Geschäftsprozesse.
- Mit Hilfe von Business Rules formulierte Anforderungen an das Verhalten von Geschäftsprozessen können anhand der halbgeordneten Abläufe auf einfache Weise überprüft werden. Halbgeordnete Abläufe eignen sich somit auch zur Verifikation des Verhaltens der entworfenen Geschäftsprozesse.

Im weiteren Verlauf dieser Arbeit werden daher halbgeordnete Abläufe zur Beschreibung des Verhaltens von Geschäftsprozeßmodellen verwendet.

### Beginn und Ende von Abläufen

Für einige der Analyseverfahren in Kapitel 7 werden noch die Begriffe *Beginn* und *Ende* eines Ablaufs benötigt. Beginn und Ende eines Ablaufs sollten einem Anfangs- und einem Endzustand des dazugehörigen Geschäftsprozeßmodells entsprechen. Zur Festlegung von Anfangs- und Endzuständen für Geschäftsprozesse kann grundsätzlich jede Markierung des erstellten Prädikate/Transitionen-Netztes herangezogen werden. Sinnvollerweise wird der Anfangszustand dabei durch die Anfangsmarkierung  $m_0$  modelliert. Im Rahmen dieser Arbeit wird davon ausgegangen, daß der Beginn eines Ablaufs dem Anfangszustand sowie das Ende eines Ablaufs einem Endzustand des dazugehörigen Geschäftsprozesses entspricht. Zur Formalisierung dieses Sachverhalts werden die folgenden Begriffe benötigt:

**Definition 35 (Linie, Schnitt)**

Sei  $(D, <)$  eine Halbordnung.

Eine *Linie* ist eine maximale Teilmenge von  $D$  derart, daß für je zwei Elemente  $x$  und  $y$  dieser Menge gilt:  $x \text{ li } y$ .

Ein *Schnitt* ist eine maximale Teilmenge von  $D$  derart, daß für je zwei Elemente  $x$  und  $y$  dieser Menge gilt:  $x \text{ co } y$ .

□

**Definition 36 (Minimale und maximale Elemente)**

Sei  $(D, <)$  eine Halbordnung.

*Minimale Elemente* sind Elemente, zu denen es keine kleineren gibt.

*Maximale Elemente* sind entsprechend definiert.

Schreibweise:  $Min(D)$ ,  $Max(D)$ .

□

Damit lassen sich jetzt *Beginn* und *Ende* eines endlichen Ablaufs festlegen:

**Definition 37 (Beginn und Ende halbgeordneter Abläufe)**

Sei  $(B, E, K)$  ein endlicher halbgeordneter Ablauf.

Der *Beginn* von  $(B, E, K)$  ist der Schnitt  $C_{\text{Beginn}} = Min(K^+)$ .

Das *Ende* von  $(B, E, K)$  ist der Schnitt  $C_{\text{Ende}} = Max(K^+)$ .

□

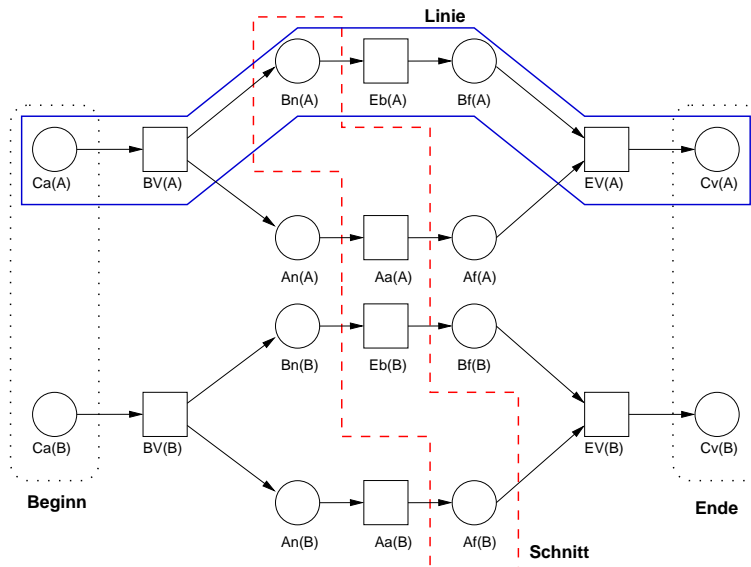


Abbildung 6.4: Begrifflichkeiten für halbgeordnete Abläufe.

### **Beispiel (Fortsetzung)**

In dem Ablauf in Abbildung 6.4 sind  $\text{Ca}(A)$  und  $\text{Ca}(B)$  *minimale*,  $\text{Cv}(A)$  und  $\text{Cv}(B)$  *maximale* Elemente.

Die Menge  $\{\text{Ca}(A), \text{Bv}(A), \text{Bn}(A), \text{Eb}(A), \text{Bf}(A), \text{Ev}(A), \text{Cv}(A)\}$  ist ein Beispiel für eine *Linie*. Die Menge  $\{\text{Bn}(A), \text{Aa}(A), \text{Eb}(B), \text{Af}(B)\}$  ist Beispiel für einen *Schnitt*.

Für den Ablauf ist die Menge  $\{\text{Ca}(A), \text{Ca}(B)\}$  der *Beginn* und  $\{\text{Cv}(A), \text{Cv}(B)\}$  das *Ende*. □

## **6.2 Das klassische VIP-Simulationskonzept**

Die Erfassung des vollständigen Zustandsraums eines realen Systems im Rahmen einer Analyse durch Simulation scheitert oft an der kombinatorischen Explosion der Anzahl möglicher Systemzustände (vgl. beispielsweise [Val96]).

Wie in Abschnitt 6.1.2 bereits geschildert, ist die Darstellung des dynamischen Verhaltens eines Systems mittels halbgeordneter Abläufe effizienter als bei Schaltfolgen. Allerdings können auch bei der Verwendung halbgeordneter Abläufe in der Regel nicht alle möglichen Systemzustände erfaßt werden.

Für die Analyse von Geschäftsprozessen in Kapitel 7 wird daher lediglich eine *Teilmenge* aller möglichen halbgeordneten Abläufe erzeugt. Eine solche Teilmenge wird auch als *Simulation* bezeichnet [DFO97].

Im Rahmen des Forschungsprojekts *VIP* wird untersucht, wie eine möglichst kleine Anzahl möglichst kurzer Abläufe generiert werden kann, ohne daß es zu einem Verlust von Analyse-relevanten Informationen kommt [DFO97]. Insbesondere bei unendlichen Abläufen sind dabei die bei der Generierung verwendeten Abbruchbedingungen von großer Bedeutung. Solche Abbruchbedingungen können *lokal* (bezogen auf einen Ablauf) oder *global* (bezogen auf die Menge aller erzeugten Abläufe) formuliert werden. Als besonders effektiv haben sich hier die sogenannten *cut-off events* erwiesen, die auch auf halbgeordnete Abläufe angewendet werden können. Durch die Identifikation von *cut-off events* wird die Bildung endlicher Präfixe unendlicher Abläufe ohne Informationsverlust ermöglicht (vgl. [Esp94], [McM92]<sup>8</sup>).

Eines der wesentlichen Ergebnisse der Arbeiten im Rahmen des *VIP*-Projekts ist die Anwendung und Weiterentwicklung oben genannter Konzepte für die Realisierung der Simulations-Komponente des *VIPtools*: *VIPsim*<sup>9</sup>. Mit Hilfe von *VIPsim* läßt sich für ein Petrinetzmodell<sup>10</sup> eine Menge halbgeordneter Abläufe erzeugen. Damit eignet sich die *VIPsim*-Komponente hervorragend zur Umsetzung von *Schritt 2* des *VIPbusiness*-Ansatzes (vgl. Abschnitt 3.1.2). Da alle der im letzten Kapitel im Rahmen der *VIPbusi-*

---

<sup>8</sup>Diese Konzepte sind eigentlich für verzweigende Abläufe entwickelt worden, eine Übertragung auf die hier verwendeten Abläufe ist jedoch möglich [Fre01].

<sup>9</sup>Eine ausführliche Beschreibung der Konzepte und Komponenten des *VIPtools* findet sich in Kapitel 8.

<sup>10</sup>Als Netzklasse werden die in Abschnitt 4.3 beschriebenen *VIPmodel*-Netze verwendet.

*ness*-Erweiterungen eingeführten Modellierungs-Bausteine auf die Klasse der *VIPmodel*-Netze zurückgeführt werden können (vgl. dazu auch die Ausführungen in Kapitel 5 sowie in den nachfolgenden Abschnitten), kann die *VIPsim*-Komponente ohne Modifikationen übernommen werden<sup>11</sup>. Zur genauen Darstellung des *VIP*-Simulationskonzepts sei an dieser Stelle auf die entsprechenden Veröffentlichungen zum *VIP*-Projekt (vgl. [DO95], [Fre96], [DFO97], [Fre97a] sowie [Fre97b]) und insbesondere auf [Fre01] verwiesen.

### 6.3 *VIPbusiness*-Erweiterungen

Bevor in Kapitel 7 halbgeordnete Abläufe zur Analyse von Geschäftsprozeßmodellen benutzt werden können, müssen die in Kapitel 5 vorgestellten *VIPbusiness*-Erweiterungen noch auf Abläufe übertragen werden.

Da die *VIPbusiness*-Erweiterungen ausnahmslos auf herkömmliche Petrinetz-Konzepte zurückgeführt werden können (vgl. dazu die Ausführungen in Kapitel 5), bleibt die Erzeugung von Abläufen im Rahmen des *VIPsim*-Konzepts von der Verwendung dieser Erweiterungen unberührt. Die Fragestellungen reduzieren sich daher auf zwei wesentliche Punkte:

- Wie werden die *VIPbusiness*-Erweiterungen in Abläufen dargestellt?

In diesem Zusammenhang muß festgelegt werden, in welcher Weise die für die Erstellung der Geschäftsprozeßmodelle eingeführten Modellierungs-Bausteine bei der Darstellung der Abläufe mit Hilfe von Kausalnetzen Verwendung finden, beispielsweise im Zusammenhang mit Aktivitäts-Transitionen oder Routing-Konstrukten.

- Wie ist die Verwendung von *VIPbusiness*-Erweiterungen in Abläufen zu interpretieren?

Dabei ist beispielsweise zu klären, welche Bedeutung die Beschriftung von Bedingungen mit Zeit- und Kostenwerten haben soll.

Im Einzelnen sind noch die folgenden Punkte zu klären:

- Übertragen des Konzepts der Aktivitäts-Transitionen auf Abläufe,
- Abläufe und Routing-Konstrukte,
- Nutzung von Ressourcen in Abläufen,
- Darstellung und Interpretation der Zeitgrößen,
- Darstellung und Interpretation der Kostengrößen,
- Abläufe von nicht-strukturierten Segmenten,
- Abläufe von hybriden Spezifikationen.

---

<sup>11</sup>Es sei an dieser Stelle daran erinnert, daß für die *VIPbusiness*-Erweiterungen aus den Bereichen *Zeit* und *Kosten* zwar keine Entsprechungen in *VIPmodel*-Netzen existieren, diese Erweiterungen jedoch bei der Simulation nicht berücksichtigt werden (vgl. Abschnitt 3.1.2).



### 6.3.1 Abläufe und Aktivitäts-Transitionen

In Analogie zu den Darstellungsmöglichkeiten für Aktivitäts-Transitionen in markierten Petrinetzen ist es auch für Abläufe möglich, Aktivitäts-Transitionen in vergrößerter Sichtweise oder explizit als Sequenz der Teil-Aktivitäten zu betrachten. Um deutlich zu machen, daß es sich bei einem Ereignis in einem Ablauf um die vergrößerte Darstellung der Ausführung einer Sequenz von Teil-Aktivitäten handelt, werden die Ereignisse in den Abläufen wie Aktivitäts-Transitionen des Geschäftsprozeßmodells mit einem stilisierten  $A$  versehen. Ein entsprechendes Ereignis wird als *Aktivitäts-Ereignis*  $e_A$  bezeichnet. Bezüglich Aktivitäts-Ereignissen können die übrigen Ereignisse und Bedingungen eines Ablaufs als *extern* bzw. *intern* unterschieden werden. Mit  $\beta(e_A) = \tau_A$  können die internen Bedingungen von  $e_A$  wie folgt referenziert werden:

- *Transport*:  $\alpha(b_T(e_A)) = s_T(\tau_A)$ ,
- *Rüsten*:  $\alpha(b_R(e_A)) = s_R(\tau_A)$ ,
- *Bearbeiten*:  $\alpha(b_B(e_A)) = s_B(\tau_A)$ ,
- *fertigungsbedingtes Warten*:  $\alpha(b_{W_f}(e_A)) = s_{W_f}(\tau_A)$ ,
- *Vorbereiten*:  $\alpha(b_V(e_A)) = s_V(\tau_A)$ .

Die Ereignisse innerhalb eines Aktivitäts-Ereignisses können analog referenziert werden, beispielsweise  $B_V(e_A)$  als Beginn der Ausführung der Vorbereitung für eine Aktivitäts-Transition  $\tau_A$ .

Auf die Angabe einer Abbildungsfunktion für die Vergrößerung soll an dieser Stelle verzichtet werden, da die entsprechenden Konzepte in Abschnitt 5.1 bereits ausführlich erläutert worden sind und sich unverändert von markierten Prädikate/Transitionen-Netzen auf halbgeordnete Abläufe übertragen lassen.

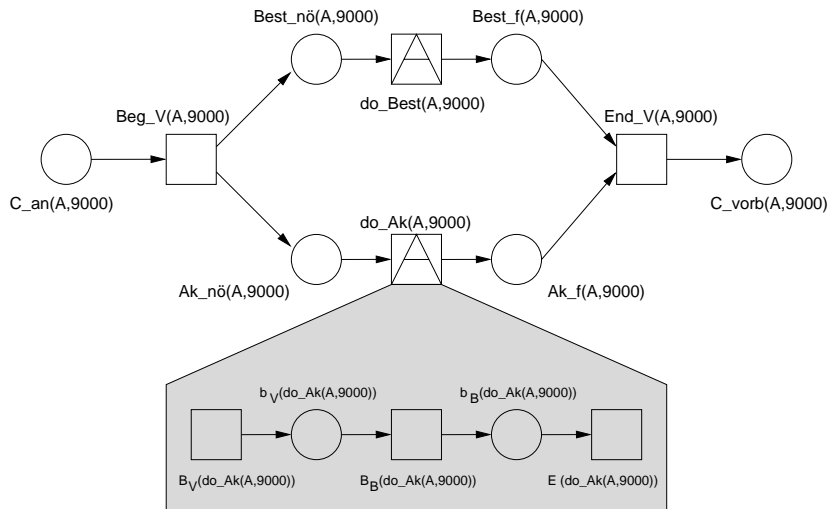


Abbildung 6.5: Darstellung von Aktivitäts-Transitionen in Abläufen.

### **Beispiel**

Abbildung 6.5 zeigt einen Ablauf für das Geschäftsprozeßmodell aus Abbildung 5.3. Um die Zuordnung der Bedingungen und Ereignisse zum Geschäftsprozeßmodell etwas zu erleichtern, wurden als Beschriftung der Bedingungen und Ereignisse etwas ausführlichere Bezeichnungen als in Abschnitt 6.1 gewählt. Das Aktivitäts-Ereignis  $do\_Ak(A)$  ist sowohl in verfeinerter als auch in vergrößerter Form dargestellt.

□

### **6.3.2 Routing-Konstrukte in Abläufen**

Um deutlich machen zu können, welche Ereignisse in einem Ablauf der Ausführung und welche den dabei getroffenen Routing-Entscheidungen entsprechen, werden Routing-Konstrukte auch in Abläufen verwendet. Dies wirkt sich in erster Linie auf die Darstellung der entsprechenden Ereignisse in den zur graphischen Repräsentation von Abläufen verwendeten Kausalnetzen aus.

Ereignisse, die auf AND-Splits oder AND-Joins abgebildet werden, erhalten wie die entsprechenden Transitionen eine Erweiterung ihres graphischen Symbols durch ein stilisiertes logisches UND. Da die Semantik von AND-Splits und AND-Joins der Semantik vorwärts- bzw. rückwärtsverzweigter Transitionen entspricht, ist die Semantik der entsprechenden Ereignisse damit ebenfalls geklärt.

Die Verwendung von impliziten OR-Konstrukten in den Geschäftsprozeßmodellen erfordert keine Erweiterung der in Abschnitt 6.1.2 vorgestellten Konzepte zu halbgeordneten Abläufen.

Gleiches gilt für die Darstellung expliziter OR-Konstrukte in verfeinerter Form, da die Auswahlentscheidung innerhalb dieser Routing-Konstrukte mittels impliziter OR-Konstrukte realisiert wird. Anders als bei Aktivitäts-Transitionen, bei denen die verfeinerte Sicht auch in Abläufen sämtliche Elemente der Sequenz der Teil-Aktivitäten enthält (da zur Ausführung der modellierten Aktivität *alle* Teil-Aktivitäten ausgeführt werden müssen), enthält die verfeinerte Sicht expliziter OR-Konstrukte nur noch Elemente, die der getroffenen Auswahlentscheidung entsprechen. Bei der Darstellung eines expliziten OR-Splits beziehungsweise eines expliziten OR-Joins in vergrößerter Sicht zeigt sich die getroffene Auswahlentscheidung darin, daß die entsprechenden Ereignisse (neben der Ergänzung ihres graphischen Symbols um ein stilisiertes OR) nur noch eine Ausgangs- bzw. nur noch eine Eingangskante haben.

### **Beispiel**

Abbildung 6.6 zeigt einen Ablauf des Geschäftsprozeßmodells aus Abbildung 5.12. Die Ereignisse  $Beg\_V(A, 9000)$  und  $End\_V(A, 9000)$  sind als AND-Split bzw. AND-Join gekennzeichnet worden. Für das der Ausführung eines expliziten OR-Split entsprechende Ereignis  $Ü?(A, 9000)$  ist die getroffene Auswahlentscheidung zusätzlich in verfeinerter Sichtweise angegeben.

□

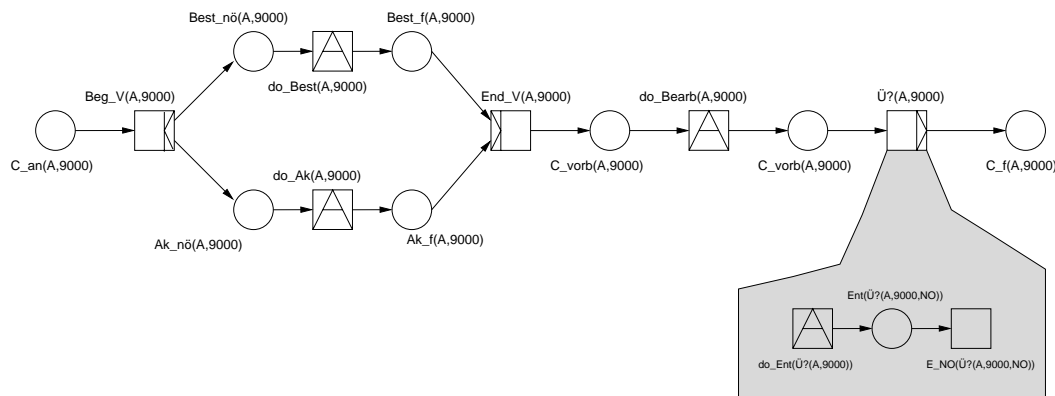


Abbildung 6.6: Darstellung von Routing-Konstrukten in Abläufen.

### 6.3.3 Nutzung von Ressourcen in Abläufen

Die Nutzung von Ressourcen spielt bei der Analyse von Geschäftsprozeßmodellen in Kapitel 7 eine entscheidende Rolle. Kennzahlen wie der *Auslastungsgrad* einer Ressource sind wichtige Anhaltspunkte für die Beurteilung der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs, beschränkte Ressourcen können bei gemeinsamer Nutzung durch unabhängig ausführbare Aktivitäten das Verhalten eines Geschäftsprozeßmodells entscheidend beeinflussen. Daher wird in diesem Abschnitt auf die Nutzung von Ressourcen in Abläufen eingegangen.

Wie in Abschnitt 5.3 erläutert, werden in dieser Arbeit Ressourcen und ihre Nutzung durch Ressourcen-Stellen sowie eine zusätzliche interne Stelle der entsprechenden Aktivitäts-Transition modelliert. Die Anzahl der verfügbaren Ressourcen einer Ressourcen-Klasse entspricht der initialen Markierung der zugehörigen Ressourcen-Stelle. Gemäß der Definition halbgeordneter Abläufe in Abschnitt 6.1.2 wird der Beginn eines Ablaufs um Bedingungen ergänzt, die in Beschriftung und Anzahl zur Anfangsmarkierung der Ressourcen-Stellen korrespondieren. Als wichtige Eigenschaft dieser Bedingungen kann  $\alpha(b) \in R$  festgehalten werden. Die Nutzung einer Ressource durch eine Aktivität spiegelt sich in einer Bedingung  $b_{Res}(e_A)$  wider, die als interne Bedingung eines Aktivitäts-Ereignisses  $e_A$  auf die interne Stelle  $s_{Res}(\tau_A)$  der zugehörigen Aktivitäts-Transition  $\tau_A = \beta(e_A)$  abgebildet wird:  $\alpha(b_{Res}(e_A)) = s_{Res}(\tau_A)$ . Den Kantenbeschriftungen im Geschäftsprozeßmodell entsprechend enthält die Beschriftung dieser Bedingung den Bezeichner der genutzten Ressource. Zusätzlich wird in der Beschriftung der dem Schalten der Eingangs- und Ausgangs-Transitionen der Aktivitäts-Transitionen entsprechenden internen Ereignisse angegeben, welche Ressource beim Schalten ausgewählt wurde.

Durch die Darstellung mittels interner Bedingungen von Aktivitäts-Ereignissen kann die Nutzung von Ressourcen in Abläufen wie in den dazugehörigen Geschäftsprozeßmodellen ein- und ausgeblendet werden. Dies ist vor allem bei der Analyse von *Linearisierungseffekten* in Abschnitt 7.4 von großem Nutzen.

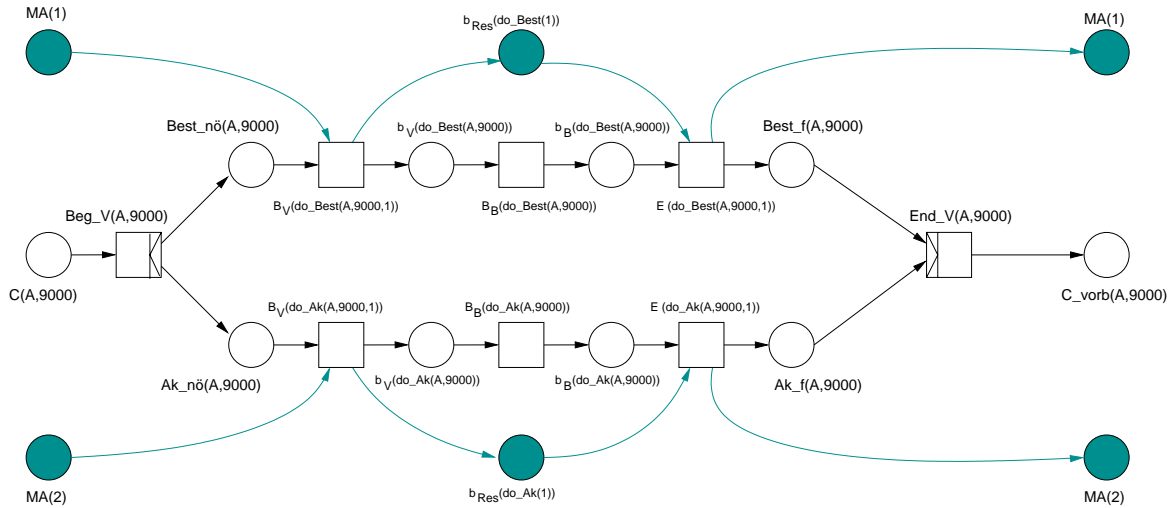


Abbildung 6.7: Nutzung von Ressourcen in Abläufen.

### Beispiel

In dem Ablauf in Abbildung 6.7 ist die Nutzung von Ressourcen für die Aktivitäts-Ereignisse  $do\_Ak(A,9000)$  und  $do\_Best(A,9000)$  in detaillierter Sichtweise dargestellt. In dem dargestellten Ablauf wird das Anlegen der Akte von Mitarbeiter 2 ausgeführt, die Bestätigung des Eingangs erfolgt durch Mitarbeiter 1. Die internen Bedingungen, die der Nutzung dieser Ressourcen entsprechen, haben die Beschriftungen  $b_{Res}(do\_Ak(2))$  bzw.  $b_{Res}(do\_Best(1))$  erhalten. Die Beschriftung der internen Ereignisse, die der Ausführung der Eingangs- und Ausgangs-Transitionen der zugehörigen Aktivitäts-Transition entsprechen, ist um die beim Schalten gewählte Ressource erweitert worden. So hat beispielsweise das Ereignis, das dem Beginn der Vorbereitung der Eingangsbestätigung entspricht, die Beschriftung  $B_V(do\_Best(A,9000,1))$  erhalten.

□

### 6.3.4 Abbildung von Zeitgrößen auf Abläufe

Wie in den Abschnitten 3.1 und 6.2 bereits erläutert, werden Zeit- und Kostenaspekte bei der Simulation eines Geschäftsprozeßmodells nicht berücksichtigt. Mit Hilfe der folgenden Abbildungsfunktion können Zeitbewertungen zu Analysezwecken jedoch problemlos vom Geschäftsprozeßmodell auf die generierten Abläufe übertragen werden:

#### Definition 38 (Zeitbewertung halbgeordneter Abläufe)

Sei  $(B, E, K)$  ein endlicher halbgeordneter Ablauf eines *VIPbusiness*-Netzes  $N$ . Die *Zeitbewertung* von  $(B, E, K)$  ist eine Abbildung  $t : B \rightarrow \mathbb{N}$  mit:

$$t(b) = t(\alpha(b))$$

□

Jede Bedingung  $b$  aus  $B$  erhält also die Zeitbewertung der Stelle des *VIPbusiness*-Netzes, auf die sie abgebildet wird, als zusätzliche Beschriftung. Der Wert  $t(b)$  entspricht dabei der Dauer, die eine Marke  $\lambda(b)$  bei Ausführung des *VIPbusiness*-Netzes auf der Stelle  $\alpha(b)$  verweilt.

Durch die angegebene Abbildung wird echte (variable<sup>12</sup>) Wartezeit Bedingungen zugeordnet, die auf externe Stellen des Geschäftsprozeßmodells abgebildet werden. Die Zeitbewertungen dieser Bedingungen sind also zunächst unbekannte Größen, im Gegensatz zu den Zeitbewertungen für die restlichen (auf interne Stellen des Geschäftsprozeßmodells abgebildeten) Bedingungen, denen die festen Aktivitätszeiten zugeordnet werden. Ein Verfahren zur Ermittlung echter Wartezeit in Abläufen mit Zeitbewertung wird in Abschnitt 7.2.2 vorgestellt.

Da eine Ressource stets für die gesamte Dauer der Ausführung einer Aktivität der entsprechenden Aktivitäts-Transition  $\tau_A$  zugeordnet wird (vgl. Abschnitt 5.4), läßt sich die Dauer der Nutzung einer Ressource durch  $\tau_A$  als Summe der Zeitbewertungen der übrigen internen Bedingungen von  $\tau_A$  ermitteln. Diese Summe wird der Bedingung  $b_{Res}(\tau_A)$  zugeordnet.

Der Vorgehensweise für Aktivitäts-Transitionen in Abschnitt 5.4.3 entsprechend erhalten Aktivitäts-Ereignisse in vergrößerter Darstellung die Summe der Zeitbewertungen der internen Bedingungen als Anschrift.

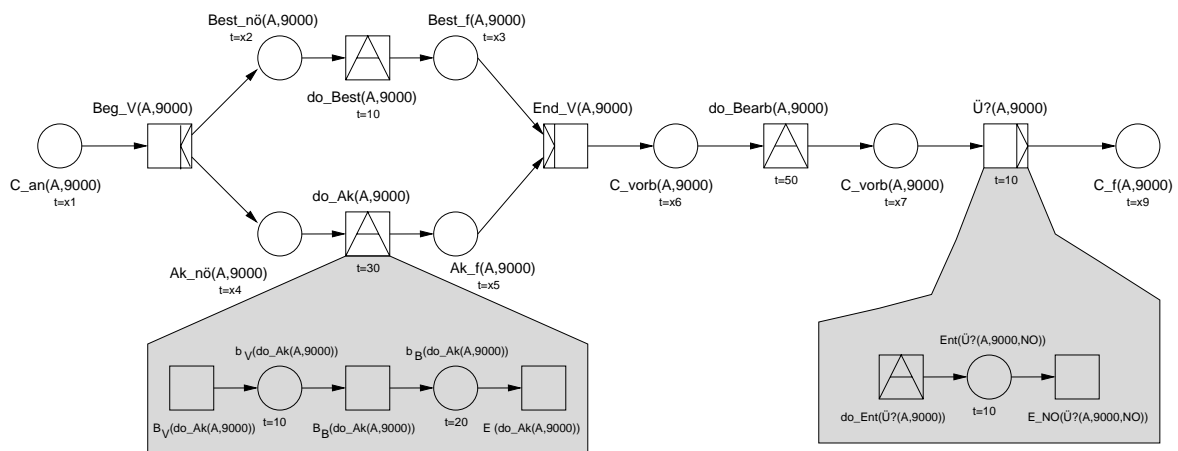


Abbildung 6.8: Ablauf mit Zeitbewertungen.

### Beispiel

Abbildung 6.8 zeigt einen Ablauf mit Zeitbewertungen für das Geschäftsprozeßmodell aus Abbildung 5.21.

□

<sup>12</sup>vgl. Abschnitt 5.4.

### 6.3.5 Abbildung von Kostengrößen auf Abläufe

Kostenbewertungen können analog zu den Zeitbewertungen auf die generierten Abläufe abgebildet werden:

**Definition 39 (Kostenbewertung halbgeordneter Abläufe)**

Sei  $(B, E, K)$  ein halbgeordneter Ablauf eines *VIPbusiness*-Netztes  $N$ . Die *Kostenbewertung* von  $(B, E, K)$  ist eine Abbildung  $k : B \rightarrow \mathbb{N}$  mit:

$$k(b) = k(\alpha(b)) = k(t(\alpha(b))) = k(t(b))$$

□

Jede Bedingung  $b$  aus  $B$  erhält also die Kostenbewertung der Stelle des Geschäftsprozeßmodells, auf die sie abgebildet wird, als zusätzliche Beschriftung. Der Wert  $k(b)$  entspricht den Kosten, die durch das Verweilen einer Marke  $\lambda(b)$  auf der Stelle  $\alpha(b)$  bei Ausführung des *VIPbusiness*-Netztes verursacht werden.

Gemäß den Ausführungen in Abschnitt 5.5 ist die Kostenbewertung einer Bedingung dabei eine Funktion der Zeitbewertung der Bedingung (bzw. der Zeitbewertung der Stelle, auf die die Bedingung abgebildet wird). Aufgrund der Zuordnung der unbekanntem echten Wartezeit zu Bedingungen, die auf externe Stellen des Geschäftsprozeßmodells abgebildet werden, sind die Kostenbewertungen dieser Bedingungen zunächst ebenfalls unbekannte Größen. Die Kostenbewertungen der übrigen (auf interne Stellen des Geschäftsprozeßmodells abgebildeten) Bedingungen lassen sich aus den festen Zeitbewertungen für Aktivitätszeiten ableiten, die Bedingungen erhalten also eine feste Kostenbewertung. Ein Verfahren zur Ermittlung und Optimierung der echten Wartekosten eines Ablaufs wird in Abschnitt 7.3.2 vorgestellt.

Werden bei der Ausführung einer Aktivität Ressourcen in Anspruch genommen, so werden die dabei entstehenden Kosten der internen Bedingung  $b_{Res}(\tau_A)$  zugeordnet. Anders als bei der Zeitbewertung der Bedingungen, bei der die Zeitbewertung von  $b_{Res}(\tau_A)$  der Summe der Zeitbewertungen der übrigen internen Bedingungen von  $\tau_A$  entspricht, ist die Kostenbewertung von  $b_{Res}(\tau_A)$  nicht unbedingt gleich der Summe der Kostenbewertungen der übrigen internen Bedingungen von  $\tau_A$ . Vielmehr gibt sie den durch Einsatz von Ressourcen einer bestimmten Klasse verursachten Anteil der Kosten einer Aktivität  $\tau_A$  an. Für den Fall, daß die Kostenbewertung von  $b_{Res}(\tau_A)$  der Summe der Kostenbewertungen der übrigen internen Bedingungen von  $\tau_A$  entspricht, fallen bei der Ausführung der Aktivität ausschließlich Kosten für Ressourcen an.

Bei der Beschriftung von Aktivitäts-Ereignissen in vergrößerter Darstellung wird analog zur Zeitbewertungsverfahren (vgl. den vorangegangenen Abschnitt zur Zeitbewertung von Abläufen sowie Abschnitt 5.5 zur Beschriftung des Geschäftsprozeßmodells mit Kostenbewertungen).

**Beispiel**

Abbildung 6.9 zeigt einen Ablauf mit Kostenbewertungen für das Geschäftsprozeßmodell aus Abbildung 5.22.

□

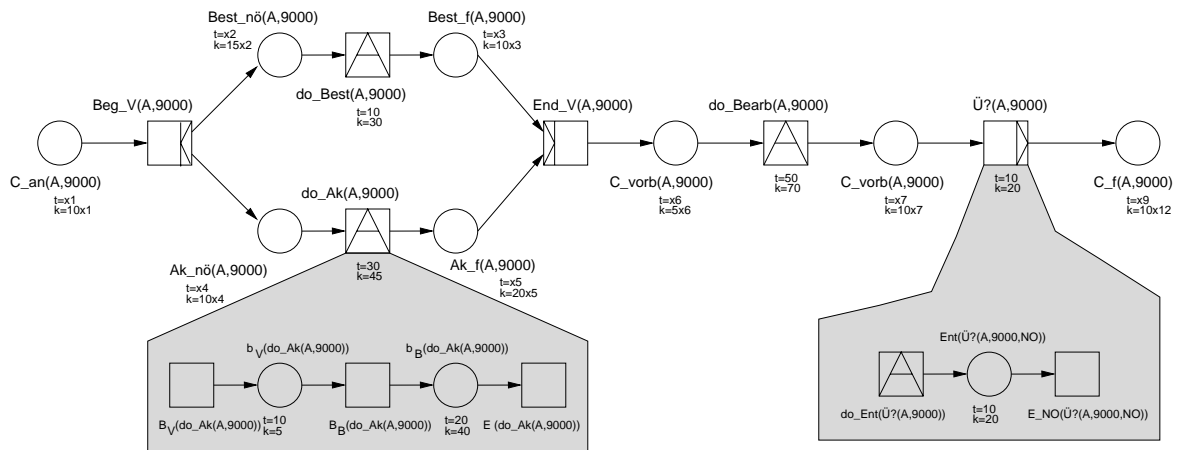


Abbildung 6.9: Ablauf mit Kostenbewertungen.

### 6.3.6 Abläufe von nicht-strukturierten Segmenten

Um die in Abschnitt 4.2 eingeführte Schaltregel für Prädikate/Transitionen-Netze auf hybride Spezifikationen anwenden zu können, müssen in diesem Abschnitt einige semantische Erweiterungen eingeführt werden. Der Grund dafür liegt in der Spezifikation der Aktivitäten der nicht-strukturierten Segmente mittels Netzfragmenten. Bei der Ausführung dieser Aktivitäten stellt sich die Frage, wie die entsprechenden Netzfragmente untereinander beziehungsweise mit dem Start- sowie dem Zielzustand verknüpft werden können.

Dabei läßt sich das Problem, das Verhalten hybrider Spezifikationen mit Hilfe eines Simulators zu generieren (beispielsweise der *VIPsim*-Komponente des *VIPtools*) auf verhältnismäßig einfache Weise lösen. In Abschnitt 5.6.2 wurde bereits darauf hingewiesen, daß jedes nicht-strukturierte Segment durch Erzeugung alternativ ausführbarer Pfade für jede zulässige Kombinationsmöglichkeit der Netzfragmente in eine konstruktive Spezifikation in Form eines Prädikate/Transitionen-Netzes ohne nicht-strukturierte Segmente überführt werden kann. Wird diese Umwandlung für alle nicht-strukturierten Segmente einer hybriden Spezifikation durchgeführt, so ist das Ergebnis ein vollständig strukturiertes Geschäftsprozeßmodell, dessen Simulation mit Hilfe der bekannten *VIPsim*-Konzepte unter Zuhilfenahme der in den vorangegangenen Abschnitten eingeführten *VIPbusiness*-Erweiterungen für Abläufe ohne weiteres möglich ist. Die Umwandlung der nicht-strukturierten Segmente sollte dabei automatisch erfolgen. Die resultierenden Modelle sind in der Regel unübersichtlich und für Anwender nur sehr schwer verständlich. Sie sind damit lediglich als Input für den Simulator zu gebrauchen.

Da die umgewandelten Geschäftsprozeßmodelle für Anwender in der Regel nicht mehr verständlich sind, ist es für eine Interpretation der generierten Abläufe notwendig, die Abläufe mit den nicht-strukturierten Segmenten des ursprünglichen Geschäftsprozeßmodells in Verbindung zu bringen. Aus diesem Grund wird in diesem Abschnitt das Konzept halbgeordneter Abläufe für nicht-strukturierte Segmente erweitert.

In Abschnitt 5.6.2 wurde bereits darauf hingewiesen, daß die Ausführung der Aktivitäten

von nicht-strukturierten Segmenten durch Verbinden der entsprechenden Netzfragmente erfolgt. Dies geschieht durch Verknüpfung von Stellen gleicher Domäne und gleichen Namens. Dabei werden die Stellen des Startzustands oder des Nachbereichs eines Netzfragments mit Stellen im Vorbereich anderer Netzfragmente verknüpft. Zusätzlich können Nachbereichsstellen noch mit Stellen des Zielzustands verknüpft werden. Bevor wir diesen Sachverhalt im weiteren Verlauf dieses Abschnitts formal beschreiben, betrachten wir zunächst ein Beispiel.

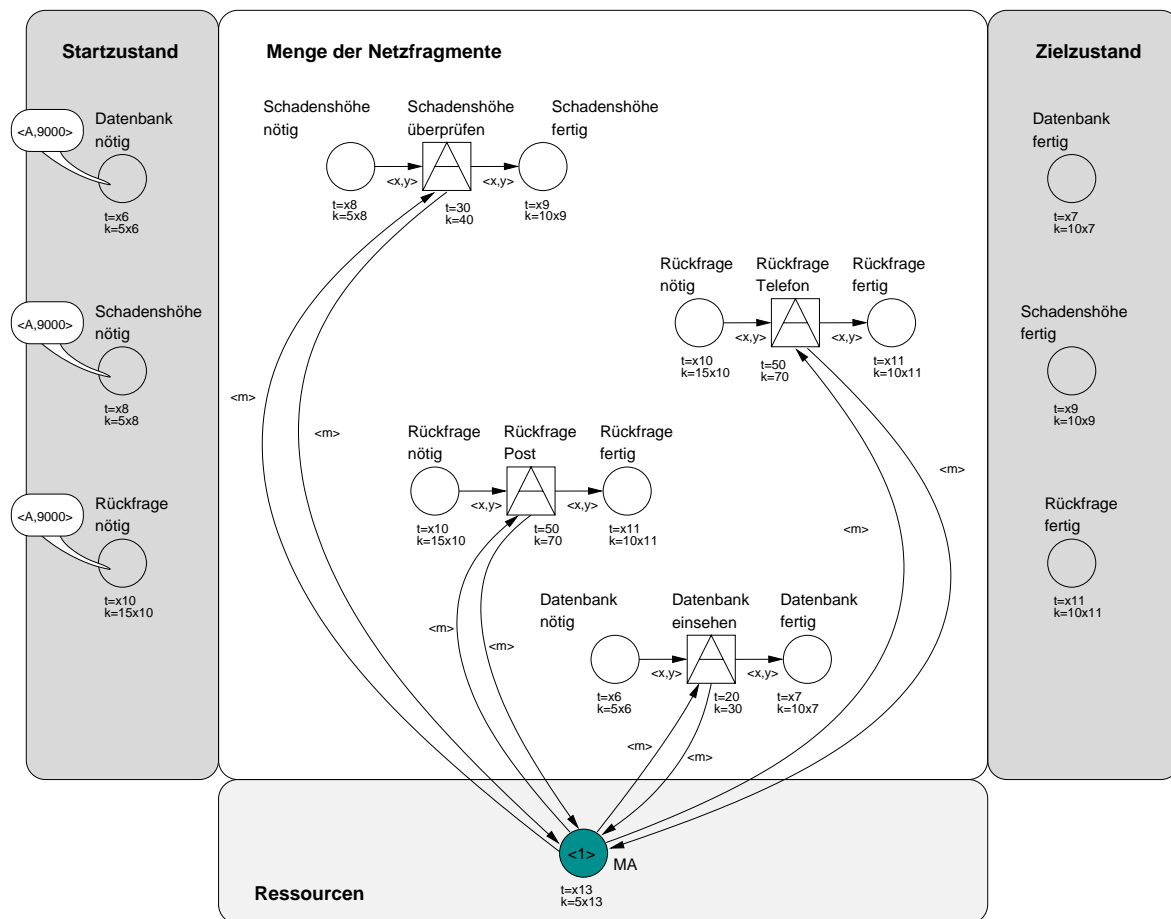


Abbildung 6.10: Nicht-strukturiertes Segment aus Abbildung 5.26 mit markiertem Startzustand.

### Beispiel

Abbildung 6.10 zeigt das nicht-strukturierte Segment der hybriden Spezifikation aus Abbildung 5.26 mit markiertem Startzustand und ohne Business Rules (diese werden später noch betrachtet). Durch das nicht-strukturierte Segment wird die eigentliche Bearbeitung der Schadensfälle durch einen Mitarbeiter der Versicherung modelliert. Dabei sind 3 Ak-



tivitäten auszuführen, möglichst von demselben Mitarbeiter und somit sequentiell<sup>13</sup>. Die Ausführung der Aktivitäten dieses nicht-strukturierten Segments sollen nun als halbgeordneter Ablauf dargestellt werden.

Als Voraussetzung für die Ausführung von Aktivitäten in dem nicht-strukturierten Segment müssen zunächst Bedingungen des Startzustands erfüllt und entsprechend Stellen des Startzustands markiert sein (in Abbildung 6.10 ist diese Voraussetzung erfüllt, da alle Stellen des Startzustands markiert sind). Dabei wird für jede Marke eine (minimale) Bedingung des halbgeordneten Ablaufs erzeugt. Zusätzlich wird eine Bedingung für jede anfangs verfügbare Ressource generiert. Das Ergebnis ist ein Schnitt mit minimalen Elementen, der den Beginn des Ablaufs darstellt (vgl. Abbildung 6.11).

Im nächsten Schritt wird ein Netzfragment ausgewählt, dessen Vorbereichsstellen in der Menge der markierten Stellen des Startzustands enthalten sind. Nun wird das Netzfragment über die Vorbereichsstellen mit dem Startzustand verknüpft. Die Verknüpfung erfolgt dabei durch eine Verschmelzung von Kopien der entsprechenden Stellen. Durch die Verschmelzung von *Kopien* ist es möglich, ein Netzfragment mehrfach zu verwenden. Die modellierte Aktivität kann damit in einem Ablauf mehrfach ausgeführt werden. Die ursprüngliche Markierung geht auf die verschmolzenen Stellen-Kopien über. Damit sollten jetzt alle Vorbereichsstellen des Netzfragments markiert sein, die Aktivitäts-Transition ist damit aktiviert und kann gemäß der bekannten Schaltregel schalten. An dieser Stelle erklärt sich die Forderung nach markierten Stellen im Startzustand, denn es macht nur Sinn, die Vorbereichsstellen des Netzfragments mit markierten Stellen zu verbinden, da sonst die Aktivitäts-Transition nicht schalten kann.

In unserem Beispiel entscheiden wir uns, das Einsehen der Datenbank als erstes vorzunehmen. Die Vorbereichsstelle **Datenbank nötig** ist in der Menge der Stellen des Startzustands enthalten, eine Verknüpfung des Netzfragments mit dem Startzustand also möglich. Zudem ist die Stelle **Datenbank nötig** im Startzustand markiert, das Verschmelzen der Stellen führt also zu einer Aktivierung der Aktivitäts-Transition **Datenbank einsehen**. Durch Schalten dieser Aktivitäts-Transition gelangen wir in unserem Ablauf zu *Schnitt 1* (vgl. Abbildung 6.11). Als Ressource zur Ausführung der Aktivität wurde dabei **Mitarbeiter 1** gewählt. Da die Nachbereichsstelle **Datenbank fertig** zudem in der Stellenmenge des Zielzustands enthalten ist, kann durch Verschmelzen dieser Stellen die Stelle im Zielzustand markiert werden. Die resultierende Bedingung **DB\_f(A,9000)** in Abbildung 6.11 ist maximales Element und damit ein Teil vom Ende des Ablaufs.

Als nächste Aktivität wird die Berechnung der Schadenshöhe vorgenommen. Die Verknüpfung des Netzfragments und die Ausführung der Aktivitäts-Transition erfolgt wie oben beschrieben und führt zu *Schnitt 2* des Ablaufs in Abbildung 6.11.

Mit der Ausführung der telefonischen Rücksprache als letzter Aktivität sind alle nötigen Aktivitäten zur Bearbeitung der Schadensmeldung durchgeführt worden. Damit ist das Ende des Ablaufs in Abbildung 6.11 erreicht.

□

---

<sup>13</sup>Eine detaillierte Beschreibung des modellierten Geschäftsprozesses findet sich in Abschnitt 5.6.2.

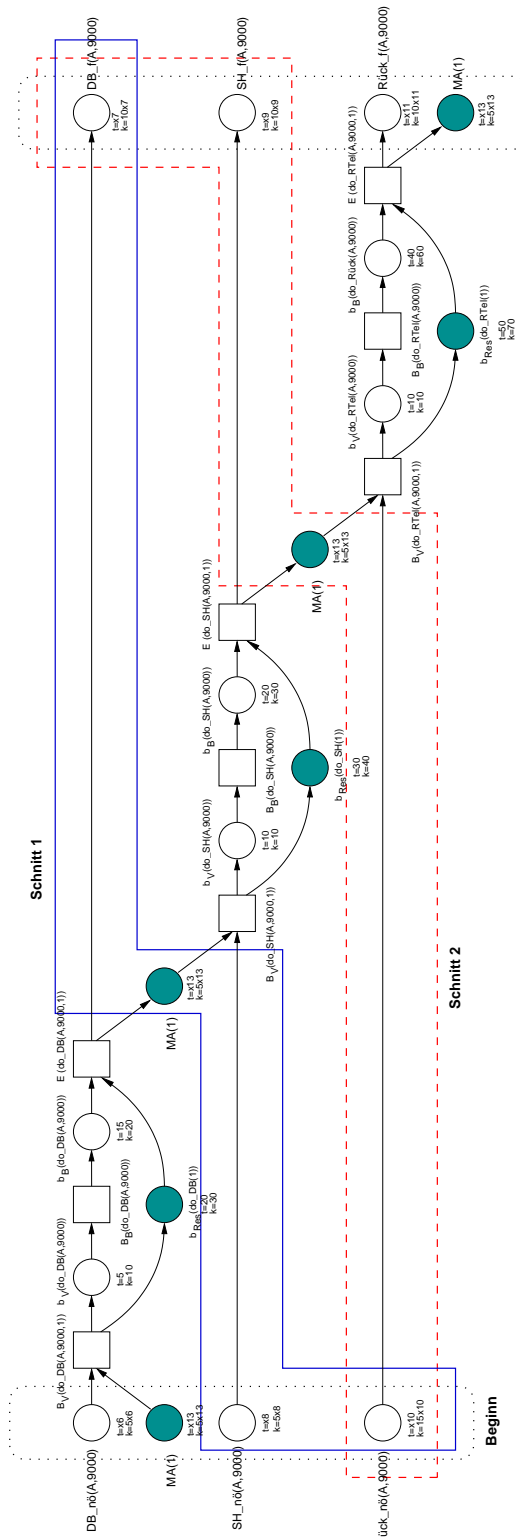


Abbildung 6.11: Ablauf des nicht-strukturierten Segments aus Abbildung 6.10.

Das gerade beschriebene Beispiel hat gezeigt, wie sich durch Zusammensetzen der Netzfragmente Abläufe von nicht-strukturierten Segmenten erzeugen lassen. Für die Analyse dieser Abläufe ist es jedoch von großer Bedeutung, sie mit dem zugehörigen Geschäftsprozeßmodell in Verbindung bringen zu können. In Abschnitt 6.1.2 ist dazu bereits eine Definition halbgeordneter Abläufe eingeführt worden. Auch wenn diese Definition in der bisherigen Form nur für vollständig strukturierte Geschäftsprozeßmodelle anwendbar ist, können wesentliche Teile übernommen werden. So entsprechen die minimalen Elemente des halbgeordneten Ablaufs weiterhin Marken der Anfangsmarkierung, gleiches gilt für die maximalen Elemente und einen vereinbarten Zielzustand. In beiden Fällen entsprechen die Marken, die beim Schalten einer Transition gemäß Schaltregel entfernt bzw. generiert werden, den Vor- und Nachbedingungen des Ereignisses, welches wiederum dem Schalten der Transition im gewählten Schaltmodus entspricht. Der wesentliche Unterschied bei der Ausführung vollständig und nicht vollständig strukturierter Geschäftsprozeßmodelle besteht darin, daß bei der Ausführung der nicht-strukturierten Segmente Netzfragmente durch Verschmelzung von Kopien zusammengefügt werden müssen, bevor Abläufe generiert werden können.

Wie im Beispiel bereits beschrieben, erfolgt das Zusammenfügen der Netzfragmente über eine vorübergehende Verschmelzung von Stellen. Dabei gehen Marken auf die verschmolzenen Stellen über. Durch die vorübergehende Verschmelzung der Stellen wird die Zuordnung der Bedingungen des Ablaufs zu den Marken der Stellen komplizierter: da die Zuordnung der Bedingungen für das nicht-strukturierte Segment im ursprünglichen Zustand (ohne verschmolzene Stellen) erfolgen soll, korrespondiert jede Bedingung des Ablaufs statt zu einer nun zu zwei Stellen<sup>14</sup>. Aus diesem Grund wird die Abbildung  $\alpha$  aus Definition 34 durch zwei Abbildungen  $\alpha_1$  und  $\alpha_2$  ersetzt. Für jede Bedingung müssen beide Abbildungen definiert sein und auf die gleiche Stelle führen. Dabei bildet  $\alpha_1$  die Bedingungen auf Stellen ab, die bei der Verschmelzung der Stellen die Lieferanten der Marken sind. Da Netzfragmente anfänglich keine Marken enthalten, kommen zunächst nur Stellen des Startzustands als Lieferanten von Marken in Frage, nach Schalten ihrer Aktivitäts-Transitions auch die Nachbereichsstellen von Netzfragmenten. Entsprechend sind Empfänger von Marken zunächst die Vorbereichsstellen von Netzfragmenten, schließlich die Stellen des Zielzustands. Auf die Empfänger-Stellen werden Bedingungen mittels  $\alpha_2$  abgebildet.

Damit können jetzt Abläufe von nicht-strukturierten Segmenten definiert werden. Gegenstand dieser Definition ist ausschließlich das zulässige Zusammensetzen der Netzfragmente eines nicht-strukturierten Segments, Business Rules und ihre Einhaltung werden noch nicht berücksichtigt. Aus diesem Grund werden Abläufe, die den in der Definition formulierten Anforderungen genügen, als *konstruktiv korrekte* Abläufe eines nicht-strukturierten Segments bezeichnet:

**Definition 40 (Konstruktiv korrekter Ablauf)**

Sei mit  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, SZ_{ns})$  ein nicht-strukturiertes Segment gegeben. Sei zudem mit  $m_0(S_{0_{ns}})$  eine Anfangsmarkierung der Stellen des Startzustands von  $ns$  gegeben. Sei  $(B, E, K)$  ein Kausalnetz und seien  $\alpha_1, \alpha_2, \beta$  Abbildungen mit:

- $\alpha_1$  bildet die Bedingungen des Kausalnetzes auf Stellen des Startzustands oder Stel-

---

<sup>14</sup>So korrespondiert beispielsweise die Bedingung `DB_nö(A,9000)` zu der Stelle `Datenbank nötig`, die in der Spezifikation des nicht-strukturierten Segments in Abbildung 6.10 sowohl Element des Startzustands als auch Vorbereichsstelle des Netzfragments mit der Aktivitäts-Transition `Datenbank einsehen` ist.

len von Netzfragmenten ab, die einen nichtleeren Vorbereich besitzen:

$$\alpha_1 : B \rightarrow S_{0_{ns}} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid \bullet s \neq \emptyset\}$$

- $\alpha_2$  bildet die Bedingungen des Kausalnetzes auf Stellen des Zielzustands oder Stellen von Netzfragmenten ab, die einen nichtleeren Nachbereich besitzen:

$$\alpha_2 : B \rightarrow S_{Z_{ns}} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid s \bullet \neq \emptyset\}$$

- $\beta$  bildet die Ereignisse des Kausalnetzes auf die Transitionen oder die Aktivitäts-Transitionen der Netzfragmente ab<sup>15</sup>:

$$\beta : E \rightarrow \bigcup_{nf \in NF_{ns}} T_{nf} \cup T_{A_{nf}}$$

Sei  $\lambda$  eine Funktion, die Bedingungen und Ereignissen des Kausalnetzes ein Label zuweist. Für Bedingungen  $b \in B$  ist  $\lambda(b)$  eine Marke auf einer zugehörigen Stelle  $\alpha_1(b)$ <sup>16</sup>. Für Ereignisse  $e \in E$  ist  $\lambda(e)$  ein Schaltmodus der korrespondierenden Transition  $\beta(e)$ .

Durch  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  wird ein *konstruktiv korrekter Ablauf* eines nicht-strukturier-ten Segments beschrieben, wenn gilt:

- Jede Bedingung im Kausalnetz wird durch  $\alpha_1$  und  $\alpha_2$  auf dieselbe Stelle abgebildet:

$$\text{Für alle } b \in B : \alpha_1(b) = \alpha_2(b).$$

- Für jede Marke aus der Anfangsmarkierung  $m_0(S_{0_{ns}})$  der zum Startzustand gehörenden Stellen existiert im Kausalnetz genau eine entsprechend beschriftete Bedingung ohne Vorbereich:

$$\text{Für alle } s \in S_{0_{ns}} : \text{für alle } d \in \mathcal{D}_s : \mu_s(d) = |\{b \in B \mid \alpha_1(b) = s, \bullet b = \emptyset, \lambda(b) = d\}|.$$

- Für jede Marke aus der Markierung  $m(S_{Z_{ns}})$  der Stellen des Zielzustands existiert im Kausalnetz genau eine entsprechend beschriftete Bedingung ohne Nachbereich:

$$\text{Für alle } s \in S_{Z_{ns}} : \text{für alle } d \in \mathcal{D}_s : \mu_s(d) = |\{b \in B \mid \alpha_2(b) = s, b \bullet = \emptyset, \lambda(b) = d\}|.$$

- Der Vorbereich von Ereignissen im Kausalnetz wird durch  $\alpha_2$  auf den Vorbereich der jeweils korrespondierenden Transition eines Netzfragments abgebildet; dieses gilt mit  $\alpha_1$  entsprechend für den Nachbereich von Ereignissen:

$$\text{Für alle } e \in E : \alpha_2(\bullet e) = \bullet \beta(e) \text{ und } \alpha_1(e \bullet) = \beta(e) \bullet.$$

<sup>15</sup>Für den Fall, daß ein Ereignis auf eine Aktivitäts-Transition abgebildet wird, handelt es sich um ein Aktivitäts-Ereignis gemäß der Ausführungen in Abschnitt 6.3.1.

<sup>16</sup>Es sei daran erinnert, daß nur  $\alpha_1$  Bedingungen auf Stellen mit Marken abbildet.

- Die Bedingungen im Vorbereich eines Ereignisses  $e$  entsprechen in Anzahl und Beschriftung genau denjenigen Marken, welche durch Schalten der Transition  $\beta(e) = \tau$  im Modus  $\lambda(e) = p_\tau$  gemäß Kantenbeschriftung konsumiert werden:

Für alle  $e \in E$  mit  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

für alle  $s \in \bullet\tau$  : für alle  $d \in \mathcal{D}_s$  :  $[\varphi_{s,\tau}(p_\tau)](d) = | \{b \in B \mid b \in \bullet e, \lambda(b) = d\} |$  .

Entsprechendes gilt für den Nachbereich von Ereignissen und die durch das Schalten von  $\tau$  produzierten Marken:

Für alle  $e \in E$  :  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

für alle  $s \in \tau\bullet$  : für alle  $d \in \mathcal{D}_s$  :  $[\varphi_{s,\tau}(p_\tau)](d) = | \{b \in B \mid b \in e\bullet, \lambda(b) = d\} |$  .

- Falls für  $\tau$  eine Schaltbedingung  $guard(\tau)$  existiert, muß  $guard(\tau)$  für die Belegung der Variablen gemäß  $p_\tau$  wahr sein.

□

### Beispiel

Abbildung 6.11 zeigt einen konstruktiv korrekten Ablauf des nicht-strukturierten Segments aus Abbildung 6.10.

□

Die Menge der konstruktiv korrekten Abläufe enthält alle Abläufe des nicht-strukturierten Segments, die durch zulässige Kombination der Netzfragmente generiert werden können. In vielen Fällen wird diese Menge durch Business Rules eingeschränkt. Mit Hilfe von Business Rules wurden in Abschnitt 5.6.3 Anforderungen formuliert, die bei der Ausführung der Aktivitäten des nicht-strukturierten Segments stets erfüllt sein müssen. Diese Abläufe werden als *restriktiv korrekte* Abläufe eines nicht-strukturierten Segments bezeichnet:

#### Definition 41 (Restriktiv korrekter Ablauf)

Sei mit  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment gegeben. Durch  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  wird ein *restriktiv korrekter Ablauf* von  $ns$  beschrieben, wenn jede Business Rule  $br \in BR_{ns}$  befolgt wird.

□

Abläufe eines nicht-strukturierten Segments, die sowohl konstruktiv als auch restriktiv korrekt sind, werden als *vollständig korrekte* Abläufe bezeichnet:

#### Definition 42 (Vollständig korrekter Ablauf)

Sei mit  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment gegeben. Durch  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  wird ein *vollständig korrekter Ablauf* von  $ns$  beschrieben, wenn er konstruktiv und restriktiv korrekt ist.

□

Im Hinblick auf die Analyse von Abläufen in Kapitel 7 sind letzten Endes nur vollständig korrekte Abläufe eines nicht-strukturierten Segments von Interesse. Um sicherzustellen, daß die Menge der Abläufe, die zur Analyse des Geschäftsprozeß-Entwurfs benutzt wird, ausschließlich vollständig korrekte Abläufe enthält, bieten sich grundsätzlich zwei Vorgehensweisen an:

- Die Business Rules werden bei der Generierung der Abläufe durch den Simulator berücksichtigt. Sobald eine der Regeln verletzt ist, wird die Generierung eines Ablaufs abgebrochen. Die so erzeugte Menge der Abläufe enthält ausschließlich vollständig korrekte Abläufe.
- Es werden zunächst alle konstruktiv korrekten Abläufe generiert. Anschließend wird untersucht, welche dieser Abläufe den mit Hilfe der Business Rules formulierten Anforderungen nicht genügen. Diese Abläufe werden aus der Menge der generierten Abläufe entfernt.

Die erste Methode hat den Vorteil, daß durch den sofortigen Abbruch der Generierung von nicht vollständig korrekten Abläufen gegebenenfalls der Simulations-Aufwand reduziert werden kann. Jedoch muß im Gegenzug für jeden Simulations-Schritt überprüft werden, ob eine Verletzung von Business Rules stattgefunden hat. Diese Überprüfung ist mit erheblichem Anpassungs-Aufwand für die Simulations-Komponente *VIPsim* des *VIPtools* verbunden. Zudem können bei diesem Vorgehen nur Regeln berücksichtigt werden, die zum Zeitpunkt der Simulation bereits formuliert worden sind. Da es dem Anwender jedoch möglich sein soll, im Rahmen der Durchführung von *Schritt 3* des *VIPbusiness*-Ansatzes auch mit den Auswirkungen verschiedener Business Rules zu experimentieren, wählen wir im Rahmen dieser Arbeit die zweite der oben beschriebenen Methoden zur Generierung von vollständig korrekten Abläufen.

Die Überprüfung der restriktiven Korrektheit von Abläufen wird daher in die verschiedenen Möglichkeiten zur Analyse von Abläufen in Kapitel 7 eingeordnet. Entsprechend wird in Abschnitt 7.1.2 erläutert, wie die restriktive Korrektheit eines Ablaufs bezüglich der 5 in Abschnitt 5.6.3 eingeführten Arten von Business Rules überprüft werden kann.

### 6.3.7 Abläufe von hybriden Spezifikationen

Die bisherigen Ausführungen zu Abläufen von nicht-strukturierten Segmenten beziehen sich ausschließlich auf einzelne nicht-strukturierte Segmente. Diese Segmente sind jedoch in der Regel Teile hybrider Spezifikationen von Geschäftsprozessen. Eine hybride Spezifikation setzt sich aus einem Prädikate/Transitionen-Netz zur Modellierung der strukturierten Teile des Geschäftsprozesses sowie einer Menge von nicht-strukturierten Segmenten zusammen (vgl. Abschnitt 5.6.4). Entsprechend können im Ablauf einer hybriden Spezifikation eine Menge von Teilnetzen identifiziert werden, die der Ausführung der Aktivitäten der nicht-strukturierten Segmente entsprechen. Die Teile des Ablaufs, die in keinem dieser Teilnetze enthalten sind (im folgenden als *Restnetz* bezeichnet), entsprechen der Ausführung der Aktivitäten des strukturierten Teils der hybriden Spezifikation. Um den Zusammenhang zwischen Kausalnetz zur Darstellung eines Ablaufs und der zugehörigen hybriden Spezifikation beschreiben zu können, müssen die bereits aus den Definitionen 34 und 40 bekannten Abbildungsfunktionen für Bedingungen und Ereignisse des Kausalnetzes auf

die Elemente des Geschäftsprozeßmodells an die Zerlegung des Kausalnetzes in Teilnetze und Restnetz angepaßt werden. Diese Überlegungen bilden den Ausgangspunkt für die nachfolgende Definition der Abläufe von hybriden Spezifikationen:

**Definition 43 (Ablauf einer hybrid strukturierten Spezifikation)**

Sei eine hybride Spezifikation eines Geschäftsprozesses mittels eines Prädikate/Transitionen-Netzes mit Aktivitäts-Transitionen  $N_{TA} = (S, T, T_A, F, \mathcal{D}, m_0, \mathcal{P}, \Phi)$  sowie einer Menge  $NS$  von nicht-strukturierten Segmenten  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, SZ_{ns})$  gegeben.

Seien  $(B, E, K)$  ein Kausalnetz und  $TN$  eine Menge stellenberandeter Teilnetze  $tn$  der Form  $(B_{tn}, E_{tn}, K_{tn})$  von  $(B, E, K)$ . Dabei besitzen die Teilnetze in  $TN$  gemeinsame Elemente allenfalls derart, daß maximale Bedingungen des einen Teilnetzes minimale Bedingungen eines anderen Teilnetzes sind:

Für je zwei  $tn_1, tn_2 \in TN$  mit  $tn_1 \neq tn_2$  gilt:

$$(B_{tn_1} \cup E_{tn_1}) \cap (B_{tn_2} \cup E_{tn_2}) \subseteq \{b \in B_{tn_1} \mid b\bullet = \emptyset\} \cap \{b \in B_{tn_2} \mid \bullet b = \emptyset\} \text{ oder} \\ (B_{tn_1} \cup E_{tn_1}) \cap (B_{tn_2} \cup E_{tn_2}) \subseteq \{b \in B_{tn_1} \mid \bullet b = \emptyset\} \cap \{b \in B_{tn_2} \mid b\bullet = \emptyset\}.$$

Die Vor- bzw. Nachbereiche der Bedingungen sind dabei bezogen auf das jeweilige Teilnetz zu verstehen.

Bezeichne  $rn = (B_{rn}, E_{rn}, K_{rn})$  das Restnetz, das sich ergibt, wenn aus  $(B, E, K)$  alle in  $TN$  enthaltenen Teilnetze mit Ausnahme ihrer Ränder  $R_{tn}$  entfernt werden:

$$B_{rn} = B \setminus \bigcup_{tn \in TN} (B_{tn} \setminus R_{tn}), \\ E_{rn} = E \setminus \bigcup_{tn \in TN} E_{tn} \text{ und} \\ K_{rn} = K \cap [(B_{rn} \times E_{rn}) \cup (E_{rn} \times B_{rn})].$$

Seien  $A_1, A_2$  Mengen von Abbildungen mit:

- Die Abbildungen  $\alpha_{1_{tn}}$  aus  $A_1$  bilden jeweils die Bedingungen eines Teilnetzes  $tn$  aus  $TN$  auf Stellen des Startzustands eines nicht-strukturierten Segments oder auf Stellen der Netzfragmente ab, die einen nichtleeren Vorbereich besitzen:

$$\alpha_{1_{tn}} : B_{tn} \rightarrow \bigcup_{ns \in NS} \left( S_{0_{ns}} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid \bullet s \neq \emptyset\} \right).$$

- Die Abbildungen  $\alpha_{2_{tn}}$  aus  $A_2$  bilden jeweils die Bedingungen eines Teilnetzes  $tn$  aus  $TN$  auf Stellen des Zielzustands eines nicht-strukturierten Segments oder auf Stellen der Netzfragmenten ab, die einen nichtleeren Nachbereich besitzen:

$$\alpha_{2_{tn}} : B_{tn} \rightarrow \bigcup_{ns \in NS} \left( SZ_{ns} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid s\bullet \neq \emptyset\} \right).$$

Sei  $\alpha$  eine Abbildung der Bedingungen des Restnetzes auf Stellen von  $N_{TA}$ :

$$\alpha : B_{rn} \rightarrow S.$$

Sei  $\beta$  ein Abbildung der Ereignisse von  $(B, E, K)$  auf Transitionen von  $N_{TA}$  oder Transitionen der Netzfragmente der nicht-strukturierten Segmente:

$$\beta : E \rightarrow T \cup \bigcup_{ns \in NS} \left( \bigcup_{nf \in NF_{ns}} T_{nf} \right).$$

Sei  $\lambda$  eine Funktion, die Bedingungen und Ereignissen des kausalen Gesamtnetzes ein Label zuweist:

- für Bedingungen  $b \in B$  ist  $\lambda(b)$  eine Marke auf einer zugehörigen Stelle  $\alpha_{1tn}(b)$  bzw.  $\alpha(b)$ .
- für Ereignisse  $e \in E$  ist  $\lambda(e)$  ein Schaltmodus der zugehörigen Transition  $\beta(e)$ .

Durch  $(B, E, K, A_1, A_2, \alpha, \beta, \lambda)$  wird ein *Ablauf einer hybriden Spezifikation* beschrieben, wenn gilt:

- Für jede Marke aus der Anfangsmarkierung des Pr/T-Netzes existiert in  $(B, E, K)$  genau eine entsprechend beschriftete Bedingung ohne Vorbereich:

$$\text{Für alle } s \in S : \text{für alle } d \in \mathcal{D}_s : \mu_s(d) = | \{ b \in B \mid \alpha(b) = s, \bullet b = \emptyset, \lambda(b) = d \} | .$$

- Für jedes nicht-strukturierte Segment  $ns$  in  $NS$  gibt es genau ein Teilnetz  $tn \in TN$  sowie je eine Abbildung  $\alpha_{1tn} \in A_1$  und  $\alpha_{2tn} \in A_2$  derart, daß  $(B_{tn}, E_{tn}, K_{tn}, \alpha_{1tn}, \alpha_{2tn}, \beta, \lambda)$  einen vollständig korrekten Ablauf von  $ns$  beschreibt. Dabei sind für paarweise verschiedene nicht-strukturierte Segmente in  $NS$  auch die korrespondierenden Teilnetze in  $TN$  sowie die zugehörigen Abbildungen in  $A_1$  und  $A_2$  verschieden.
- Für jedes Teilnetz  $tn \in TN$  gibt es genau ein nicht-strukturiertes Segment  $ns \in NS$  sowie je eine Abbildung  $\alpha_{1tn} \in A_1$  und  $\alpha_{2tn} \in A_2$  derart, daß  $(B_{tn}, E_{tn}, K_{tn}, \alpha_{1tn}, \alpha_{2tn}, \beta, \lambda)$  einen vollständig korrekten Ablauf von  $ns$  beschreibt.
- Der Vorbereich von Ereignissen im Kausalnetz wird durch  $\alpha$  auf den Vorbereich der jeweils korrespondierenden Transition des Systemnetzes abgebildet:

$$\text{Für alle } e \in E_{rn} : \alpha(\bullet e) = \bullet \beta(e).$$

Dieses gilt entsprechend für den Nachbereich von Ereignissen:

$$\text{Für alle } e \in E_{rn} : \alpha(e\bullet) = \beta(e)\bullet .$$

- Die Bedingungen im Vorbereich eines Ereignisses  $e$  des kausalen Restnetzes entsprechen in Anzahl und Beschriftung genau denjenigen Marken in  $N_{TA}$ , welche durch das Schalten der Transition  $\beta(e) = \tau$  im Modus  $\lambda(e) = p_\tau$  gemäß Kantenbeschriftung konsumiert werden:

Für alle  $e \in E_{rn}$  mit  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

$$\text{für alle } s \in \bullet \tau : \text{für alle } d \in \mathcal{D}_s : [\varphi_{s,\tau}(p_\tau)](d) = | \{ b \in B \mid b \in \bullet e, \lambda(b) = d \} | .$$



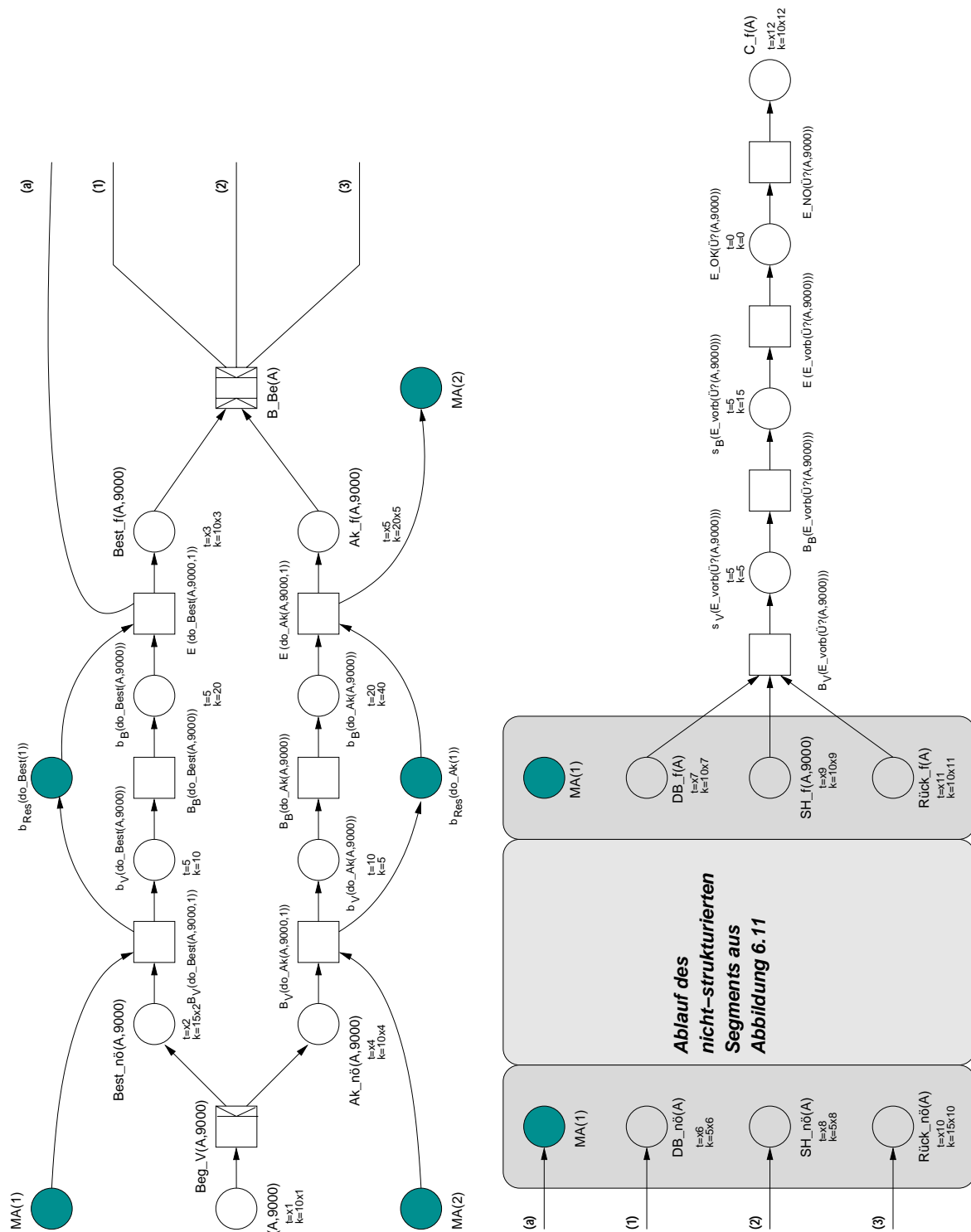


Abbildung 6.12: Ablauf des hybrid strukturierten *VIPbusiness*-Netzes aus Abbildung 5.26.

Entsprechendes gilt für den Nachbereich von Ereignissen und die durch das Schalten von  $\tau$  produzierten Marken:

Für alle  $e \in E_{\tau n}$  mit  $\beta(e) = \tau$  : es existiert  $p_\tau \in \mathcal{P}_\tau$  mit :

für alle  $s \in \tau \bullet$  : für alle  $d \in \mathcal{D}_s$  :  $[\varphi_{s,\tau}(p_\tau)](d) = | \{b \in B \mid b \in e \bullet, \lambda(b) = d\} |$  .

Falls für  $\tau$  eine Schaltbedingung  $guard(\tau)$  existiert, muß  $guard(\tau)$  für die Belegung der Variablen gemäß  $p_\tau$  wahr sein.

□

### **Beispiel**

Abbildung 6.12 zeigt einen Ablauf des hybrid strukturierten *VIPbusiness*-Netzes aus Abbildung 5.26.

□

In diesem Kapitel sind alle Konzepte eingeführt worden, die zur Simulation von Geschäftsprozeßmodellen und damit zur Durchführung von *Schritt 2* des *VIPbusiness*-Ansatzes benötigt werden. Die im Rahmen der Simulation erzeugte Menge halbgeordneter Abläufe bildet die Grundlage für die Ausführungen zur Analyse von Geschäftsprozeßmodellen im nachfolgenden Kapitel.

# Kapitel 7

## Analyse und Optimierung

Nachdem in den vorangegangenen Kapiteln mit der Erstellung von Geschäftsprozeß-Modellen und deren Simulation die ersten beiden Schritte des *VIPbusiness*-Ansatzes beschrieben worden sind, wenden wir uns in diesem Kapitel *Schritt 3* des Ansatzes - der Analyse und der Optimierung der entworfenen Geschäftsprozesse - zu.

Dazu werden in Abschnitt 7.1 zunächst Methoden zur qualitativen<sup>1</sup> Analyse der erstellten Geschäftsprozeß-Modelle vorgestellt. Im Blickpunkt stehen dabei vor allem Verfahren aus dem Bereich *Konsistenz/Verifikation* eines Geschäftsprozeßmodells beziehungsweise des zugehörigen *VIPbusiness*-Netzes.

Der Begriff der *restriktiven Korrektheit* von Abläufen nicht-strukturierter Segmente wurde bereits in Abschnitt 6.3.6 eingeführt, die zugehörigen *Business Rules* zur Formulierung von Anforderungen an das Verhalten von Geschäftsprozeßmodellen wurden in Abschnitt 5.6.3 vorgestellt. In Abschnitt 7.1.2 wird erläutert, wie Abläufe hinsichtlich dieser formulierten Anforderungen überprüft werden können.

Mit den Verfahren zur *Zeitanalyse* in Abschnitt 7.2 beginnt die Vorstellung einer Reihe von Methoden zur *quantitativen* Analyse der Geschäftsprozeß-Modelle. Diese Verfahren setzen auf der Menge der generierten Abläufe und damit auf dem Ergebnis von Schritt 2 des *VIPbusiness*-Ansatzes auf. Ausgehend von den Verfahren zur Zeitanalyse werden in Abschnitt 7.3 Verfahren zur *Kostenanalyse* eingeführt. In Abschnitt 7.4 wird auf Aspekte des Einsatzes beschränkter Ressourcen eingegangen. Dabei handelt es sich um die Untersuchung sogenannter *Linearisierungs-Effekte* sowie um Möglichkeiten der optimalen Nutzung von beschränkten Ressourcen durch Konfiguration spezieller Kontrollparameter.

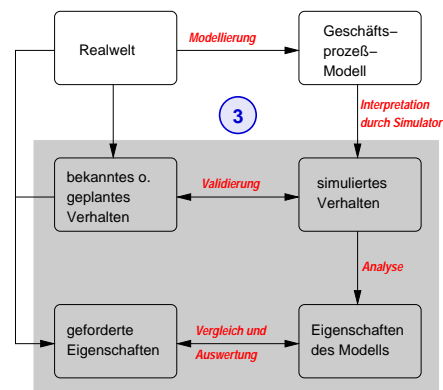


Abbildung 7.1: Schritt 3: Analyse und Optimierung.

<sup>1</sup>Der Begriff *qualitativ* wird in diesem Zusammenhang als Abgrenzung zur Untersuchung *quantitativer* Qualitätsmerkmale in den Abschnitten 7.2 bis 7.5 verwendet.

In Abschnitt 7.5 wird schließlich gezeigt, wie sich mit Hilfe von *Konfigurationen* und *Szenarien* eine systematische Analyse von Geschäftsprozeß-Entwürfen bezüglich wichtiger Kenngrößen durchführen läßt.

## 7.1 Methoden zur qualitativen Analyse

In diesem Abschnitt werden Methoden zur qualitativen Analyse von Geschäftsprozeß-Entwürfen vorgestellt. Die dabei verwendeten Verfahren und untersuchten Qualitätsmerkmale sind den Bereichen *Korrektheit/Validierung* und *Konsistenz/Verifikation* zuzuordnen (vgl. Abschnitt 2.2.1 und 2.2.2), wobei der Schwerpunkt auf den Bereich *Verifikation* gelegt wird. Auf die Möglichkeiten zur Validierung von Geschäftsprozeß-Entwürfen durch den *VIPbusiness*-Ansatz, beispielsweise durch Visualisierung des Verhaltens in Form einer graphischen Darstellung der Abläufe, wurde bereits in Abschnitt 3.1.3 eingegangen. Ein Beispiel für die Anwendung des *VIPbusiness*-Ansatzes und des *VIPtools* zur Validierung eines Geschäftsprozeß-Entwurfs findet sich Abschnitt 8.2.

Einer der Vorteile der Verwendung von Petrinetzen ist die Vielzahl der verfügbaren Konzepte und Werkzeuge zur Analyse, insbesondere im Bereich der *Verifikation* [Aal98a]. Grundsätzliches Ziel der Verifikation ist die Überprüfung oder der Nachweis von Eigenschaften, die in jedem Fall von dem erstellten Petrinetz-Modell zu erfüllen sind. In Abschnitt 7.1.1 wird mit dem bekannten Konzept der *Soundness* eines Geschäftsprozeß-Modells (vgl. beispielsweise [Aal98d] oder [VBA01]) ein Beispiel zur Nutzung bekannter Verfahren zur Analyse von *VIPbusiness*-Netzen vorgestellt.

Mit dem Begriff der *restriktiven Korrektheit* von Abläufen durch Erfüllung der zugehörigen *Business Rules* ist in den Abschnitten 6.3.6 und 5.6.3 ein weiteres Konzept zur Überprüfung der Konsistenz von Geschäftsprozeß-Entwürfen vorgestellt worden. Wie Abläufe auf die Erfüllung von Business Rules hin überprüft werden können, wird in Abschnitt 7.1.2 erläutert.

### 7.1.1 Verifikation mit Hilfe „klassischer“ Eigenschaften von Petrinetzen

Bei der *Verifikation* geht es um den formalen Nachweis von Eigenschaften, die in jedem Fall von dem Geschäftsprozeß-Entwurf zu erfüllen sind, beispielsweise die Vermeidung von *Verklemmungen*<sup>2</sup>. Für Petrinetze existieren grundsätzlich zwei verschiedene Möglichkeiten, solche Eigenschaften zu überprüfen:

- Strukturanalyse

Die *strukturellen Eigenschaften* eines (markierten) Petrinetzes zeichnen sich dadurch aus, daß sie ohne explizite Betrachtung des möglichen Verhaltens analysiert werden können. Es muß lediglich die Struktur des Petrinetzes sowie gegebenenfalls eine Anfangsmarkierung berücksichtigt werden. Der große Vorteil struktureller Eigenschaften liegt darin, daß sie auf vergleichsweise effiziente Weise überprüft werden können.

---

<sup>2</sup>vgl. beispielsweise [Rei86]. Bei einer Verklemmung kommt es zu einem Stillstand bei der Bearbeitung eines Case vor Erreichen des festgelegten Endes der Bearbeitung.

Bedauerlicherweise folgen aus ihrem Nachweis meist nur relevante dynamische Eigenschaften für spezielle Klassen von Petrinetzen [VBA01]. Da zur Analyse struktureller Eigenschaften eines Geschäftsprozesses keine Betrachtung des im Rahmen von Simulation generierten Verhaltens des Geschäftsprozeß-Modells nötig ist, sollte man die entsprechenden Analysen in der Praxis *vor* der Simulation durchführen, um den Aufwand für die Simulation fehlerhafter Modelle zu vermeiden. Die entsprechenden Verfahren kommen im Kontext des *VIPbusiness*-Ansatzes also *zwischen* Schritt 1 und Schritt 2 zum Einsatz (vgl. Abschnitt 3.1.4).

- Verhaltensanalyse

Bei der Überprüfung von *Eigenschaften des Verhaltens* von markierten Petrinetzen werden ausgehend von der Anfangsmarkierung die Menge der Schaltfolgen, der Markierungsgraph (oder ein relevanter Ausschnitt des Markierungsgraphen) oder die Menge der halbgeordneten Abläufe betrachtet. Verglichen mit strukturellen Eigenschaften ist der Rechenaufwand hier um einiges höher [EN94].

Für beide Arten der Analyse existieren bekannte Eigenschaften oder Konzepte, die beim Nachweis der Konsistenz eines Geschäftsprozeß-Entwurfs wertvolle Hilfe leisten können, beispielsweise *Stelleninvarianten* und *Transitionsinvarianten* (vgl. beispielsweise [DR98, Rei86]). Dies gilt insbesondere für strukturell eingeschränkte Netzklassen wie *Free-Choice-Netze* [Des92, DE95]. Für die Überprüfung dieser Eigenschaften ist eine Vielzahl an Verfahren und zugehörigen Werkzeugen entwickelt worden, beispielsweise *linear-algebraische* Verfahren (vgl. zum Beispiel [Des96, Des98]).

Im Kontext der Analyse relevanter Eigenschaften von Geschäftsprozeß-Modellen existiert mit dem Konzept der *Soundness* eines Geschäftsprozeß-Modells ein Ansatz, der die Analyse von Geschäftsprozeß-Modellen auf die Analyse „klassischer“ Eigenschaften von Petrinetzen zurückführt. Ausgangspunkt ist dabei die Postulierung der folgenden Anforderungen als *Minimal-Anforderungen*, die von jedem Geschäftsprozeß-Modell erfüllt werden sollten [Aal98d]:

- Die Bearbeitung eines jeden Case sollte stets beendet werden können<sup>3</sup>.
- Mit der Beendigung der Bearbeitung sind keine weiteren Aktivitäten mehr durchzuführen<sup>4</sup>.
- Für jede Aktivität sollte eine Bearbeitung eines Case existieren, bei der diese Aktivität auch wirklich ausgeführt wird.

Ob ein Geschäftsprozeß-Modell diesen Anforderungen gerecht wird, läßt sich durch Überprüfung „klassischer“ Eigenschaften von Petrinetzen<sup>5</sup> ermitteln. Damit ist Soundness ein etabliertes Beispiel dafür, wie Anforderungen an die Konsistenz eines Geschäftsprozeß-Entwurfs mit Hilfe der Vielzahl an Verfahren, die für Petrinetze bereits entwickelt worden sind, untersucht werden können.

---

<sup>3</sup>In [VBA01] wird dies als *option to complete* bezeichnet. Es ist natürlich wünschenswert, daß die Bearbeitung eines jeden Case beendet wird. Diese Eigenschaft wird von Modellen jedoch zumeist nicht erfüllt, solange keine *Fairness* bei der Auswahl von Alternativen angenommen wird [AKD98].

<sup>4</sup>auch: *proper completion* [VBA01].

<sup>5</sup>Dabei handelt es sich um *Lebendigkeit* und *Beschränktheit*, vgl. beispielsweise [DR98].

Da sich die bei der Modellierung von Geschäftsprozessen mit *VIPbusiness* benutzten Modellierungs-Bausteine auf „klassische“ Petrinetz-Konzepte zurückführen lassen, können Anforderungen an die Konsistenz für *VIPbusiness*-Netze problemlos mit Hilfe bekannter Verfahren und Werkzeuge überprüft werden. Im Rahmen der Werkzeugunterstützung des *VIPbusiness*-Ansatzes ist dazu eine Schnittstelle zwischen dem *VIPtool* und *Woflan*, einem Werkzeug der TU Eindhoven zur Verifikation von Geschäftsprozeß-Modellen mit *Workflow-Netzen*, entwickelt worden (vgl. Kapitel 8).

### 7.1.2 Restriktive Korrektheit: Überprüfung von Business Rules

In den Abschnitten 2.2.2 und 5.6.3 wurden Business Rules als Mittel zur Formulierung wünschenswerter Eigenschaften eines Geschäftsprozeß-Entwurfs eingeführt. Mit Hilfe spezieller Transitionen wurden fünf Arten von Business Rules entwickelt, um insbesondere das Verhalten der nicht-strukturierten Teile (Segmente) von Geschäftsprozessen einschränken zu können. In Abschnitt 6.3.7 wurde bereits der Begriff des *restriktiv korrekten* Ablaufs für nicht-strukturierte Segmente eingeführt: ein restriktiv korrekter Ablauf liegt vor, wenn jede spezifizierte Business Rule befolgt wird.

Business Rules können auf zwei verschiedene Arten genutzt werden:

- Steuerung der Simulation

Dabei werden Business Rules während der Simulation berücksichtigt, d.h. der Simulator generiert ausschließlich restriktiv korrekte Abläufe. Diese Art der Nutzung von Business Rules ist darüber hinaus für die Kontrolle entsprechender Informationssysteme zur Laufzeit von großem Interesse, beispielsweise beim Einsatz von Workflow-Komponenten auf Basis der entworfenen Geschäftsprozesse [Obe96]. Auf diese Weise kann die konsistente Bearbeitung von Cases sichergestellt werden.

- Bewertung der generierten Abläufe

Es werden zunächst alle möglichen Abläufe erzeugt. Diese werden anschließend auf Verletzungen von Business Rules überprüft. Durch Analyse der Umstände dieser Verletzungen lassen sich wertvolle Hinweise auf eventuelle Fehler im Geschäftsprozeß-Entwurf gewinnen.

Ausgangspunkt der Untersuchung von Abläufen auf restriktive Korrektheit sind jeweils zwei Mengen von Bedingungen  $B_1$  und  $B_2$  (mit Ausnahme von Fakt-Transitionen, dort wird lediglich  $B_1$  benötigt). Bei  $B_1$  handelt es sich um eine Menge von Bedingungen, die Marken zur Aktivierung der Transition der Business Rule entsprechen. Die Bedingungen in  $B_2$  entsprechen Marken, die durch (fiktives) Schalten der Transition der Business Rule produziert werden. Um die nachfolgenden Ausführungen zur Erfüllung der einzelnen Arten von Business Rules übersichtlicher gestalten zu können, werden  $B_1$  und  $B_2$  an dieser Stelle wie folgt definiert:

**Definition 44** ( $B_1, B_2$ )

Sei  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  eine Business Rule mit  $T_{br} = \{\tau_{br}\}$ . Ferner sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, SZ_{ns})$  ein nicht-strukturiertes Segment sowie  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  ein zugehöriger konstruktiv korrekter Ablauf.

$B_1 \subseteq B$  ist eine Menge von Bedingungen, für die gilt:

- Die Bedingungen in  $B_1$  werden durch  $\alpha_1$  auf Stellen des Startzustands oder Stellen der Netzfragmente mit nichtleerem Vorbereich abgebildet, die gleich den in der Business Rule enthaltenen Stellen sind:

$$\alpha_1 : B_1 \rightarrow \left( S_{0_{ns}} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid \bullet s \neq \emptyset\} \right) \cap S_{br}$$

wobei  $B_1$  als minimal angenommen wird.

- Für jede Stelle  $s \in S_{br}$  existiert eine solche Startstelle beziehungsweise Stelle eines Netzfragments mit Urbild in  $B_1$ .
- Darüber hinaus entsprechen die Bedingungen in  $B_1$  in Anzahl und Beschriftung einer Markierung von Stellen aus  $S_{br}$ , unter der die Transition  $\tau_{br}$  für einen Schaltmodus  $p_{\tau_{br}}$  aktiviert ist:

Es existiert  $p_{\tau_{br}} \in \mathcal{P}_{\tau_{br}}$  mit :

für alle  $s \in \bullet\tau_{br}$  und für alle  $d \in \mathcal{D}_s : [\varphi_{s, \tau_{br}}(p_{\tau_{br}})](d) = |\{b \in B_1 \mid \lambda(b) = d\}|$  .

Falls für  $\tau_{br}$  eine Schaltbedingung  $guard(\tau_{br})$  existiert, so muß  $guard(\tau_{br})$  für die Belegung der Variablen gemäß  $p_{\tau_{br}}$  wahr sein.

- Die Bedingungen in  $B_1$  sind paarweise ungeordnet und somit in mindestens einem Schnitt gemeinsam enthalten.

$B_2 \subseteq B$  ist eine Menge von Bedingungen, für die gilt:

- Die Bedingungen in  $B_2$  werden durch  $\alpha_2$  auf Stellen des Zielzustands oder Stellen der Netzfragmente mit nichtleerem Nachbereich abgebildet, die gleich den in der Business Rule enthaltenen Stellen sind:

$$\alpha_2 : B_2 \rightarrow \left( SZ_{ns} \cup \bigcup_{nf \in NF_{ns}} \{s \in S_{nf} \mid s \neq \emptyset\bullet\} \right) \cap S_{br}$$

wobei  $B_2$  als minimal angenommen wird.

- Für jede Stelle  $s \in S_{br}$  existiert eine solche Startstelle beziehungsweise Stelle eines Netzfragments mit Urbild in  $B_2$ .
- Darüber hinaus entsprechen die Bedingungen in  $B_2$  in Anzahl und Beschriftung solchen Marken auf Stellen aus  $S_{br}$ , die durch (fiktives) Schalten von  $\tau_{br}$  im Modus  $p_{\tau_{br}}$  produziert werden:

Es existiert  $p_{\tau_{br}} \in \mathcal{P}_{\tau_{br}}$  mit :

für alle  $s \in \tau_{br}\bullet$  und für alle  $d \in \mathcal{D}_s : [\varphi_{s, \tau_{br}}(p_{\tau_{br}})](d) = |\{b \in B_2 \mid \lambda(b) = d\}|$  .

- Die Bedingungen in  $B_2$  sind paarweise ungeordnet und somit in mindestens einem Schnitt gemeinsam enthalten.

□

### Fakt-Transitionen

Wie in Abschnitt 5.6.3 bereits erläutert, werden Fakt-Transitionen benutzt, um Eigenschaften eines Geschäftsprozesses zu spezifizieren, die während der gesamten Bearbeitung eines Case erfüllt sein sollen. Es darf also bei der Bearbeitung eines Case für keinen Schaltmodus der Fakt-Transition eine geeignete Menge von Marken auf den Vorbereichsstellen vorhanden sein. Ein Ablauf zur Beschreibung der Bearbeitung eines Case erfüllt die mittels einer Fakt-Transition spezifizierten Anforderungen, falls es keinen Schnitt mit Bedingungen gibt, die einer solche Menge von Marken entsprechen.

#### Definition 45 (Erfüllung von Business Rules mit Fakt-Transitionen)

Sei  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  eine Business Rule mit  $T_{br} = \{\tau_f\}$ , wobei  $\tau_f$  eine Fakt-Transition mit einer Menge von Schaltmodi  $\mathcal{P}_{\tau_f}$  und  $F_{br} = \{(s, \tau_f) \mid s \in S_{br}\}$  ist. Ferner sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment.

Ein Ablauf  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  erfüllt die Business Rule  $N_{br}$  genau dann, wenn keine Menge  $B_1 \subseteq B$  existiert.

□

### Ziel-Transitionen

Mit Hilfe von Zielen läßt sich fordern, daß ausgehend von einem spezifizierten Zustand 1 in jedem Fall irgendwann ein spezifizierter Zustand 2 erreicht wird (vgl. Abschnitt 5.6.3). Im Rahmen der Analyse soll herausgefunden werden, ob bei der Bearbeitung eines Case eine Verfehlung von Zielen möglich ist. Um einen Ablauf diesbezüglich zu untersuchen, wird dieser auf die Existenz zweier Schnitte untersucht. Falls ein Schnitt mit Bedingungen existiert, die auf eine Menge von Marken abgebildet werden, die zur Aktivierung der Ziel-Transition geeignet sind, so muß ein zweiter Schnitt existieren, dessen Bedingungen den Marken entsprechen, die durch ein Schalten der Ziel-Transition generiert würden. Für den zweiten Schnitt muß zusätzlich gelten, daß er nur Bedingungen enthält, zu denen im ersten Schnitt Bedingungen existieren, die bezüglich der Halbordnung kleiner oder gleich sind.

#### Definition 46 (Erfüllung von Business Rules mit Ziel-Transitionen)

Sei  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  eine Business Rule mit  $T_{br} = \{\tau_z\}$ , wobei  $\tau_z$  eine Ziel-Transition mit einer Menge von Schaltmodi  $\mathcal{P}_{\tau_z}$  und  $F_{br} = \{(s, \tau_z) \cup (\tau_z, s) \mid s \in S_{br}\}$  ist. Ferner sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment.

Ein Ablauf  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  erfüllt die Business Rule  $N_{br}$  genau dann, wenn gilt:

Falls eine Menge  $B_1 \subseteq B$  existiert, so muß eine Menge  $B_2 \subseteq B$  existieren mit: Für jede Bedingung  $b_2 \in B_2$  muß ein  $b_1 \in B_1$  mit  $b_1 \leq b_2$  existieren.

□



## Kausalketten-Transitionen

Mit Hilfe von *Kausalketten-Transitionen* lassen sich unmittelbare kausale Abhängigkeiten zwischen Aktivitäten untersuchen beziehungsweise spezifizieren [Des00, Fre01]. Für zwei Aktivitäten  $a_1$  und  $a_2$  kann auf diese Weise angegeben werden, welche Sequenz von Aktivitäten nach  $a_1$  ausgeführt werden muß, bevor  $a_2$  ausgeführt wird. Um einen Ablauf auf die Existenz einer solchen Sequenz hin zu untersuchen, werden wie bei Ziel-Transitionen zwei Schnitte  $B_1$  und  $B_2$  gesucht, deren Bedingungen einer Menge von Marken zur Aktivierung der Kausalketten-Transition beziehungsweise einer durch (fiktives) Schalten der Kausalketten-Transition generierten Menge von Marken entsprechen. Zusätzlich muß für jede Bedingung aus  $B_1$  ein Pfad zu einer Bedingung aus  $B_2$  existieren, bei dem die Beschriftungen der Ereignisse in Anzahl und Reihenfolge Schaltmodi der für die Kausalketten-Transition angegebenen Sequenz von Aktivitäts-Transitionen entsprechen.

### Definition 47 (Erfüllung von Business Rules mit Kausalketten-Transitionen)

Sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment. Sei eine Business Rule  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  mit  $T_{br} = \{\tau_k\}$  gegeben, wobei  $\tau_k$  eine Kausalketten-Transition mit einer Menge von Schaltmodi  $\mathcal{P}_{\tau_k}$  und  $F_{br} = \{(s, \tau_k) \cup (\tau_k, s) \mid s \in S_{br}\}$  ist. Sei  $\tau_{k_1}, \dots, \tau_{k_n}$  die Sequenz der Aktivitäts-Transitionen, die durch  $\tau_k$  spezifiziert wird.

Ein Ablauf  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  erfüllt die Business Rule  $N_{br}$  genau dann, wenn gilt:

Falls eine Menge  $B_1 \subseteq B$  existiert, so muß eine Menge  $B_2 \subseteq B$  existieren, wobei für je zwei Bedingungen  $b_1 \in B_1$  und  $b_2 \in B_2$  gelten muß:

- $b_1 < b_2$ .
- Es existiert ein Pfad von  $b_1$  nach  $b_2$  mit: Für die Sequenz  $e_1, \dots, e_n$  der Ereignisse auf diesem Pfad gilt:  $\lambda(e_1) = p_{\tau_{k_1}}, \dots, \lambda(e_n) = p_{\tau_{k_n}}$ .

□

## Nina-Transitionen

Mit Hilfe von Nina-Transitionen kann die Anforderung formuliert werden, daß bei der Bearbeitung eines Case bestimmte Aktivitäten *nicht nach* anderen Aktivitäten ausgeführt werden dürfen. Dabei beziehen sich die mit Hilfe von Nina-Transitionen formulierten Bedingungen ausschließlich auf die Bearbeitung einzelner Cases. Im Gegensatz zu den vorherigen Anfrage-Transitionen werden bei Nina-Transitionen nicht die Schnitte, sondern die Linien eines Ablaufs untersucht. Auf diese Weise lassen sich bestimmte Abhängigkeitsbeziehungen von Zuständen sowie Lebensläufe von Marken untersuchen beziehungsweise verbieten [Sch99]. Für die Abläufe ergibt sich daraus die folgende Anforderung: Es dürfen nicht alle Bedingungen, die einer Markierung von Stellen nach (fiktivem) Schalten der Nina-Transition entsprechen, Nachfolger aller Bedingungen sein, die einer Markierung entsprechen, unter der die Nina-Transition für den betreffenden Schaltmodus aktiviert ist. Anders formuliert, es muß mindestens eine Bedingung existieren, die auf eine Stelle im Nachbereich der Nina-Transition abgebildet wird und nicht nach einer auf eine Stelle im Vorbereich der Nina-Transition abgebildeten Bedingung auftritt.

**Definition 48 (Erfüllung von Business Rules mit Nina-Transitionen)**

Sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment. Sei eine Business Rule  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  mit  $T_{br} = \{\tau_n\}$  gegeben, wobei  $\tau_n$  eine Nina-Transition mit einer Menge von Schaltmodi  $\mathcal{P}_{\tau_n}$  und  $F_{br} = \{(s, \tau_n) \cup (\tau_n, s) \mid s \in S_{br}\}$  ist.

Ein Ablauf  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  erfüllt die Business Rule  $N_{br}$  genau dann, wenn gilt:

Es existieren keine zwei Mengen  $B_1, B_2 \subseteq B$ , für die gilt: Für alle  $b_2 \in B_2$  existiert ein  $b_1 \in B_1$  mit:  $b_1 \leq b_2$ .

□

**Business Rules auf Basis von ausgeschlossenen Transitionen**

Mit Hilfe von *ausgeschlossenen Transitionen* [Obe90, Vos87] lassen sich verbotene Zustandsübergänge und damit ebenfalls verbotene Reihenfolgen von Aktivitäten formulieren. Im Gegensatz zu Regeln auf Basis von Nina-Transitionen stellen ausgeschlossene Transitionen keine Anforderungen hinsichtlich des Weges, auf dem die kritischen Folgen von Zuständen vorkommen (vgl. Abschnitt 5.6.3). Ausgehend von einer Markierung, unter der die ausgeschlossene Transition für einen Schaltmodus aktiviert ist, darf keine Markierung existieren, wie sie durch (fiktives) Schalten der ausgeschlossenen Transition zustande käme. Dieses Verbot betrifft nicht nur unmittelbare, sondern sämtliche Folgemarkierungen. Für Abläufe ist damit zu untersuchen, ob ausgehend von einem Schnitt, dessen Bedingungen einer Markierung entsprechen, unter der die ausgeschlossene Transition aktiviert ist, ein zweiter Schnitt existiert, der einer Folgemarkierung der ersten Markierung entspricht, die durch ein (fiktives) Schalten der ausgeschlossenen Transition entstünde.

**Definition 49 (Erfüllung von Business Rules mit ausgeschlossenen Transitionen)**

Sei  $ns = (S_{0_{ns}}, NF_{ns}, R_{ns}, BR_{ns}, S_{Z_{ns}})$  ein nicht-strukturiertes Segment. Sei eine Business Rule  $N_{br} = (S_{br}, T_{br}, F_{br}, \mathcal{D}_{br}, \mathcal{P}_{br}, \Phi_{br})$  mit  $T_{br} = \{\tau_a\}$ <sup>6</sup> gegeben, wobei  $\tau_a$  eine ausgeschlossene Transition mit einer Menge von Schaltmodi  $\mathcal{P}_{\tau_a}$  und  $F_{br} = \{(s, \tau_a) \cup (\tau_a, s) \mid s \in S_{br}\}$  ist.

Ein Ablauf  $(B, E, K, \alpha_1, \alpha_2, \beta, \lambda)$  erfüllt die Business Rule  $N_{br}$  genau dann, wenn gilt:

Es existieren keine zwei Mengen  $B_1, B_2 \subseteq B$ , für die gilt:

- Für alle  $b_1 \in B_1$  und alle  $b_2 \in B_2 : b_1 \leq b_2$ .
- Es existiert mindestens ein  $b_2 \in B_2$  mit :  $b_1 < b_2$ .

□

---

<sup>6</sup>Nicht zu verwechseln mit Aktivitäts-Transitionen  $\tau_A$ .

## 7.2 Methoden zur Zeitanalyse

In diesem Abschnitt werden eine Reihe Verfahren vorgestellt, die es ermöglichen, für einzelne Abläufe eines Geschäftsprozeß-Modells zeitbezogene Kenngrößen zu ermitteln. Die wichtigste Kenngröße stellt dabei die *Durchlaufzeit* eines Ablaufs dar (vgl. Abschnitt 2.2.3). Da sich die Durchlaufzeit eines Ablaufs aus Aktivitätszeit (beziehungsweise den verschiedenen Bestandteilen der Aktivitätszeit) sowie echter Wartezeit ergibt, werden in Abschnitt 7.2.1 und Abschnitt 7.2.2 zunächst Verfahren zur Ermittlung dieser beiden Kenngrößen eingeführt. In Abschnitt 7.2.3 kann dann die Durchlaufzeit eines Ablaufs berechnet werden. In Abschnitt 7.2.4 werden mit dem *Auslastungsgrad* sowie der *Nutzungsdauer* wichtige auf Ressourcen bezogene Zeitgrößen ermittelt, bevor in Abschnitt 7.2.5 einige abgeleitete Zeitgrößen und Möglichkeiten zu ihrer Interpretation eingeführt werden.

### 7.2.1 Ermittlung der Aktivitätszeit

Wie in Abschnitt 2.2.3 bereits erläutert, setzt sich die Aktivitätszeit aus den Bestandteilen

- Transportzeit,
- Rüstzeit,
- Bearbeitungszeit sowie
- fertigungsbedingte Wartezeit

zusammen. Für Dienstleistungs-Aktivitäten, bei denen es nicht um die Erstellung physischer Produkte geht, kann es sinnvoll sein, Transport- und Rüstzeit zu *Vorbereitungszeit* zusammenzufassen und auf fertigungsbedingte Wartezeit zu verzichten. In diesem Fall werden damit die folgenden Bestandteile berücksichtigt:

- Vorbereitungszeit,
- Bearbeitungszeit.

#### Transportzeit

Die zur Ermittlung der Transportzeit eines Ablaufs relevanten Zeitgrößen sind die Zeitbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_T(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netztes abgebildet werden (vgl. 6.3.1). Die Transportzeit  $t_T(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht der Summe der Zeitbewertungen aller entsprechenden Bedingungen  $b$ :

$$t_T(A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_T(\tau_A)$$

#### Rüstzeit

Die zur Ermittlung der Rüstzeit eines Ablaufs relevanten Zeitgrößen sind die Zeitbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_R(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netztes abgebildet werden (vgl. 6.3.1). Die Rüstzeit

$t_R(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht der Summe der Zeitbewertungen aller entsprechenden Bedingungen  $b$ :

$$t_R(A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_R(\tau_A)$$

### Vorbereitungszeit

Die zur Ermittlung der Vorbereitungszeit eines Ablaufs relevanten Zeitgrößen sind die Zeitbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_V(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. 6.3.1). Die Vorbereitungszeit  $t_V(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht der Summe der Zeitbewertungen aller entsprechenden Bedingungen  $b$ :

$$t_V(A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_V(\tau_A)$$

### Beispiel

Für den Ablauf  $A$  in Abbildung 7.2 ergibt sich als Vorbereitungszeit:

$$\begin{aligned} t_V(A) &= t(s_V(do\_Best(A, 9000))) + t(s_V(do\_Ak(A, 9000))) + t(s_V(do\_DB(A, 9000))) \\ &\quad + t(s_V(do\_SH(A, 9000))) + t(s_V(do\_Rück(A, 9000))) \\ &\quad + t(s_V(E\_vorb(\ddot{U}?(A, 9000))) \\ &= 5 + 10 + 5 + 10 + 10 + 5 \\ &= 45 \end{aligned}$$

□

### Bearbeitungszeit

Die zur Ermittlung der Bearbeitungszeit eines Ablaufs relevanten Zeitgrößen sind die Zeitbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_B(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. 6.3.1). Die Bearbeitungszeit  $t_B(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht der Summe der Zeitbewertungen aller entsprechenden Bedingungen  $b$ :

$$t_B(A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_B(\tau_A)$$

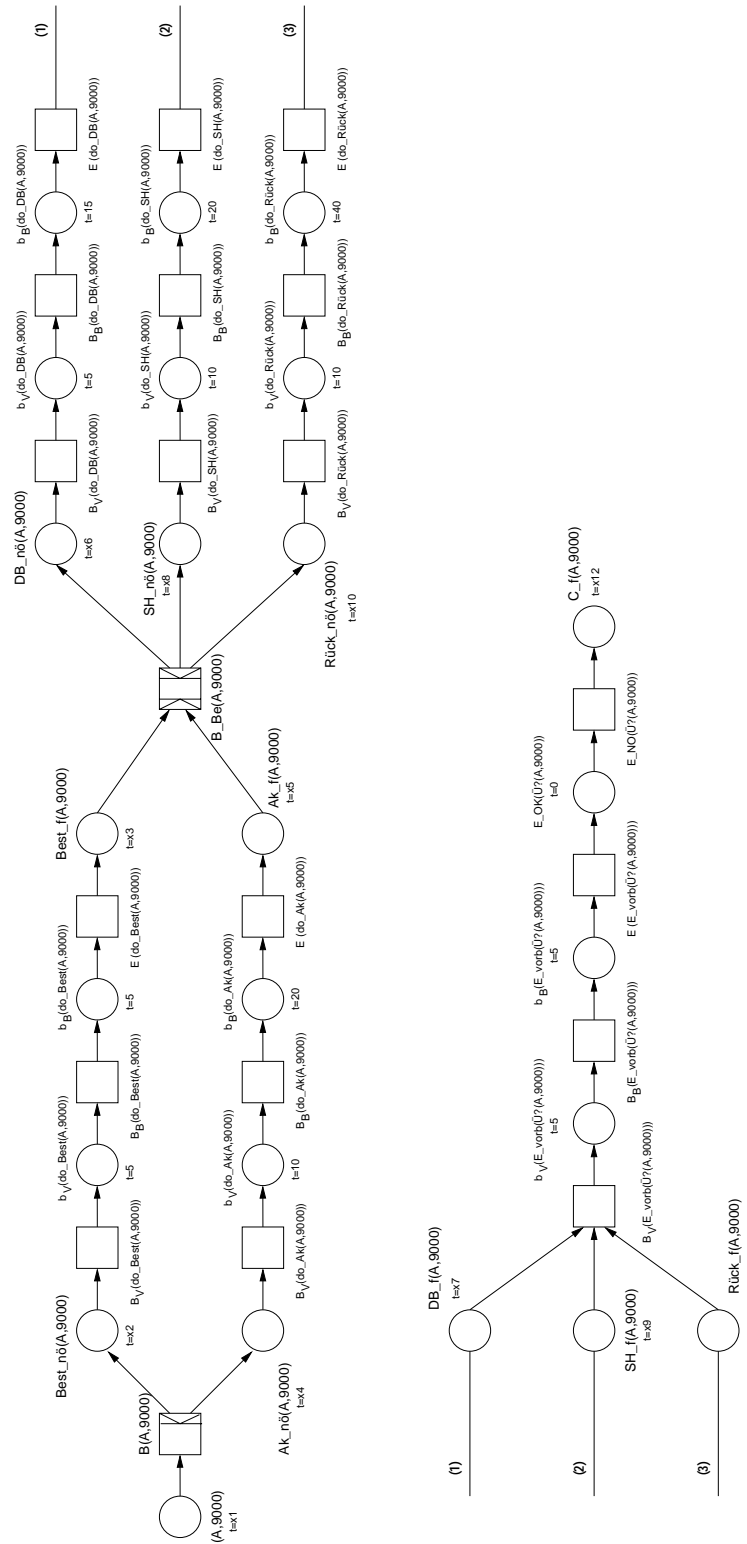


Abbildung 7.2: Ablauf des Geschäftsprozessmodells aus Abbildung 5.21 mit Zeitbewertungen.

### Beispiel

Für den Ablauf  $A$  in Abbildung 7.2 ergibt sich als Bearbeitungszeit:

$$\begin{aligned}t_B(A) &= t(s_B(do\_Best(A, 9000))) + t(s_B(do\_Ak(A, 9000))) + t(s_B(do\_DB(A, 9000))) \\ &\quad + t(s_B(do\_SH(A, 9000))) + t(s_B(do\_Rück(A, 9000))) \\ &\quad + t(s_B(E\_vorb(\ddot{U}?(A, 9000)))) \\ &= 5 + 20 + 15 + 20 + 40 + 5 \\ &= 105\end{aligned}$$

□

### Fertigungsbedingte Wartezeit

Die zur Ermittlung der fertigungsbedingten Wartezeit eines Ablaufs relevanten Zeitgrößen sind die Zeitbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_{W_f}(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. 6.3.1). Die fertigungsbedingte Wartezeit  $t_{W_f}(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht der Summe der Zeitbewertungen aller entsprechenden Bedingungen  $b$ :

$$t_{W_f}(A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_{w_f}(\tau_A)$$

### Aktivitätszeit

Die Aktivitätszeit  $t_A(\tau_A)$  einer Aktivitäts-Transition  $\tau_A$  ergibt sich als Summe der Zeitbewertungen der internen Stellen  $s_T(\tau_A), s_R(\tau_A), s_B(\tau_A), S_{W_f}(\tau_A)$  beziehungsweise  $s_V(\tau_A)$  und  $s_B(\tau_A)$ . Entsprechend ergibt sich für die Zeitbewertung  $t(e_A)$  eines Ereignisses  $e_A$ , das auf eine Aktivitäts-Transition  $\tau_A$  in vergrößernder Sichtweise abgebildet wird (vgl. 6.3.1):

$$\begin{aligned}t_A(e_A) &= t(b_1) + t(b_2) + t(b_3) + t(b_4) \\ \text{mit } \alpha(b_1) &= s_T(\tau_A), \alpha(b_2) = s_R(\tau_A), \alpha(b_3) = s_B(\tau_A), \alpha(b_4) = s_{w_f}(\tau_A)\end{aligned}$$

beziehungsweise

$$\begin{aligned}t_A(e_A) &= t(b_1) + t(b_2) \\ \text{mit } \alpha(b_1) &= s_V(\tau_A), \alpha(b_2) = s_B(\tau_A).\end{aligned}$$

Die Aktivitätszeit  $t_A(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  kann jetzt als Summe der Zeitbewertungen  $t(e_A)$  aller Ereignisse  $e_A$ , die auf eine Aktivitäts-Transition in vergrößernder Sichtweise abgebildet werden, berechnet werden:

$$t_A(A) = \sum_{e_A \in E} t(e_A) \text{ mit } \beta(e_A) \text{ ist eine Aktivitäts-Transition.}$$

Alternativ läßt sich die Aktivitätszeit  $t_A(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  auch als Summe der bereits bekannten Summen der Bestandteile der Aktivitätszeit ermitteln:

$$t_A(A) = t_T(A) + t_R(A) + t_B(A) + t_{W_f}(A)$$

beziehungsweise

$$t_A(A) = t_V(A) + t_B(A).$$

### **Beispiel**

Für den Ablauf  $A$  in Abbildung 7.2 ergibt sich als Aktivitätszeit

$$t_A(A) = t_V(A) + t_B(A) = 45 + 105 = 145$$

□

### **7.2.2 Ermittlung der echten Wartezeit**

Den Annahmen in Abschnitt 5.4.3 entsprechend ist echte Wartezeit in Form von variablen Zeitattributen externer Stellen modelliert worden, da sie von Ablauf zu Ablauf variieren kann. Für einzelne Abläufe kann die anfallende echte Wartezeit jetzt ermittelt werden.

Ausgangspunkt des hier vorgestellten Algorithmus zur Ermittlung echter Wartezeit ist die Feststellung, daß echte Wartezeit nur im Vorbereich einer rückwärtsverzweigten Transition beziehungsweise eines rückwärtsverzweigten Ereignisses (also bei der Synchronisation nebenläufig ausführbarer Aktivitäten) sowie für den Fall, daß das Ende des Ablaufs mehr als eine Bedingung enthält, auftreten kann.

Da es in diesem Abschnitt lediglich um die *Ermittlung* echter Wartezeit und nicht um deren exakte *Zuordnung* geht, wird auftretende echte Wartezeit in Abläufen den jeweiligen Bedingungen im unmittelbaren Vorbereich  $\bullet e$  eines rückwärtsverzweigten Ereignisses  $e$  zugeordnet. In Abschnitt 7.3.2 wird unter Berücksichtigung von Kostenaspekten gezeigt, daß diese Art der Verteilung nicht immer die sinnvollste Lösung darstellt. Entsprechend wird dort ein Verfahren zur kostenoptimalen Verteilung von echter Wartezeit entwickelt.

Die Bedingungen des zu untersuchenden Ablaufs erhalten ein zusätzliches Zeitattribut  $t_{Go}$ , durch welches der *frühestmögliche Schaltzeitpunkt* des nachfolgenden Ereignisses bestimmt wird.

Ausgehend von dem minimalen Schnitt wird der Ablauf von „links nach rechts“ durchlaufen. Dabei wird die Menge der jeweils aktuell zu betrachtenden Bedingungen  $B_{Fl}$  mit dem minimalen Schnitt des Ablaufs initialisiert. Ist für ein Ereignis im Nachbereich einer Bedingung der frühestmögliche Schaltzeitpunkt ermittelt, wird diese Bedingung in  $B_{Fl}$  gegen die Bedingung(en) aus dem Nachbereich dieses Ereignisses ausgetauscht. Bei  $B_{Fl}$  handelt es sich also um einen maximalen Schnitt, der wie ein „Kamm“ von links nach rechts über den Ablauf gezogen wird.

Für rückwärtsverzweigte Ereignisse wird untersucht, ob bei der Synchronisation echte Wartezeit anfällt. Dies ist der Fall, wenn sich  $t_{Go}$ -Werte der Bedingungen im Vorbereich des rückwärtsverzweigten Ereignisses unterscheiden. Der frühestmögliche Schaltzeitpunkt eines rückwärtsverzweigten Ereignisses wird durch die Bedingung  $b$  mit dem höchsten Wert für  $t_{Go}(b)$  im Vorbereich des Ereignisses bestimmt. Als Zeitbewertung wird den übrigen Vorbereichs-Bedingungen die Differenz zwischen jeweiligem  $t_{Go}$ -Wert der Bedingung und dem Maximum der  $t_{Go}$ -Werte des Vorbereichs als echte Wartezeit zugeordnet.

Da echte Wartezeit in der Regel mit Kosten verbunden ist, wird für den Ablauf eine minimale Durchlaufzeit angestrebt. Daher wird für alle Bedingungen, die auf externe Stellen des markierten Petrinetzes abgebildet werden, zunächst eine Zeitbewertung  $t(b) = 0$  vorgenommen. Stellt sich beim Durchlaufen des Ablaufs heraus, daß es im Vorbereich eines rückwärtsverzweigten Ereignisses zu echter Wartezeit kommt, wird die Zeitbewertung der Bedingungen im Vorbereich dieses Ereignisses durch die oben beschriebene Berechnung des Differenzwerts angepaßt.

Zusätzlich zu den rückwärtsverzweigten Ereignissen muß das Ende eines Ablaufs untersucht werden. Da der Ablauf erst zu Ende ist, wenn alle Bedingungen des maximalen Schnitts erreicht worden sind, kann es auch hier zu echter Wartezeit kommen. Diese Wartezeit wird in Analogie zu dem oben beschriebenen Vorgehen über den frühestmöglichen Schaltzeitpunkt eines (imaginären) Ereignisses ermittelt, in dessen Vorbereich alle Bedingungen des maximalen Schnitts liegen. Dieser Schaltzeitpunkt entspricht dem Maximum der  $t_{Go}$ -Werte dieser Bedingungen. Enthält der maximale Schnitt des Ablaufs nur eine Bedingung, entfällt dieser Schritt.

An dieser Stelle sei noch einmal daran erinnert, daß Zeit formal den Bedingungen (und nicht den Aktivitäts-Ereignissen) eines Ablaufs zugeordnet ist. Zur Verdeutlichung dieses Sachverhalts ist der Ablauf aus Abbildung 6.8 in Abbildung 7.2 ausschließlich in detaillierter Sicht ohne Aktivitäts-Ereignisse dargestellt.

Die Grundzüge des Algorithmus zur Ermittlung echter Wartezeit sollen zunächst anhand des Beispiels in Abbildung 7.2 skizziert werden:

- Ausgangspunkt ist der minimale Schnitt des Ablaufs, der in diesem Fall nur die Bedingung  $C(A,9000)$ <sup>7</sup> enthält. Damit ist  $C(A,9000)$  einziges Element der Menge der aktuell zu untersuchenden Bedingungen  $B_{Fl}$ <sup>8</sup>. Das Attribut  $t_{Go}(C(A,9000))$  für den frühestmöglichen Schaltzeitpunkt für Ereignisse im Nachbereich von  $C(A,9000)$  erhält den Wert 0, da alle Elemente des minimalen Schnitts auf externe Stellen im Systemnetz abgebildet werden, deren Zeitbewertungen echte Wartezeit darstellen (und zunächst mit dem Wert  $t(b) = 0$  angesetzt werden, s.o.). Einen Ablauf von vornherein mit echter Wartezeit beginnen zu lassen, widerspricht der Forderung nach Minimierung der Durchlaufzeit.
- Da  $C(A,9000)$  einzige Bedingung im Vorbereich des Ereignisses  $B(A,9000)$  ist, kann

<sup>7</sup>Es sei darauf hingewiesen, daß  $A$  an dieser Stelle nicht für den Ablauf  $A$ , sondern für den Identifikator des zu bearbeitenden Case steht.

<sup>8</sup>Enthält der minimale Schnitt mehr als ein Element, kann mit einer beliebigen Bedingung aus dieser Menge begonnen werden.



$B(A, 9000)$  sofort stattfinden. Die Bedingungen  $Best\_nö(A, 9000)$  und  $Ak\_nö(A, 9000)$  im Nachbereich von  $B(A, 9000)$  erhalten ebenfalls den  $t_{Go}$ -Wert 0, da das Schalten von Ereignissen nicht mit Zeitverbrauch verbunden ist.  $C(A, 9000)$  wird aus  $B_{Fl}$  entfernt, die Bedingungen  $Best\_nö(A, 9000)$  und  $Ak\_nö(A, 9000)$  werden  $B_{Fl}$  zugefügt. Welches der beiden Elemente in  $B_{Fl}$  als nächstes betrachtet wird, spielt keine Rolle. In diesem Beispiel wird  $Best\_nö(A, 9000)$  gewählt.

- Das Ereignis  $B_V(do\_Best(A, 9000))$  im Nachbereich von  $Best\_nö(A, 9000)$  kann ebenfalls sofort stattfinden. Da  $t(s_V(do\_Best(A, 9000))) = 10$ , erhält das zugehörige Attribut  $t_{Go}(s_V(do\_Best(A, 9000)))$  den Wert 10. Damit kann das nachfolgende Ereignis  $B_B(do\_Best(A, 9000))$  im Nachbereich von  $s_V(do\_Best(A, 9000))$  frühestens 10 Zeiteinheiten (ZE) nach Beginn des Ablaufs stattfinden.
- Die übrigen Zeitbewertungen  $t_{Go}(b)$  für die Bedingungen der beiden nebenläufigen Pfade für die Bestätigung des Eingangs sowie für das Anlegen der Akte lassen sich analog ermitteln. Auf diese Weise erhält man schließlich die Zeitbewertungen  $t_{Go}(Best\_f(A, 9000)) = 20$  und  $t_{Go}(Ak\_f(A, 9000)) = 30$ .
- Diese beiden Bedingungen liegen im Vorbereich des rückwärtsverzweigten Ereignisses  $B_{Be}(A, 9000)$ . Durch dieses Ereignis findet also eine Synchronisation nebenläufiger Aktivitäten statt. Der frühestmögliche Schaltzeitpunkt für  $B_{Be}(A, 9000)$  ergibt sich aus dem Maximum der  $t_{Go}$ -Werte der Bedingungen im Vorbereich. In diesem Fall bestimmt also das Anlegen der Akte den frühestmöglichen Beginn der eigentlichen Bearbeitung der Schadensmeldung. Die entsprechende echte Wartezeit, die bei der Bestätigung des Eingangs anfällt, entspricht der Differenz der  $t_{Go}$ -Werte der beiden Vorbereichs-Bedingungen, und wird (in einem ersten Ansatz) der entsprechenden Bedingung im direkten Vorbereich des rückwärtsverzweigten Ereignisses als Zeitbewertung  $t(b)$  zugeordnet. In diesem Fall ergibt sich damit  $t(Be\_f(A, 9000)) = 20$ . Um das Maximum der  $t_{Go}$ -Werte im Vorbereich eines rückwärtsverzweigten Ereignisses bestimmen zu können, muß für alle Vorbereichs-Bedingungen ein solcher Wert ermittelt worden sein. Wird beim Durchlaufen des Ablaufs festgestellt, daß für das Ereignis im Nachbereich der aktuell betrachteten Bedingung noch nicht alle  $t_{Go}$ -Werte bekannt sind, wird die aktuell betrachtete Bedingung aus  $B_{Fl}$  entfernt, und eine der verbleibenden Bedingungen in  $B_{Fl}$  als aktuell betrachtete Bedingung ausgewählt.
- Die weitere Bearbeitung der Schadensmeldung verläuft zunächst nebenläufig, bis durch das Ereignis  $B_V(E\_vorb(\ddot{U}?(A, 9000)))$  wieder synchronisiert wird. Für die Bedingungen in dessen Vorbereich ergeben sich die Werte  $t_{Go}(DB\_f(A, 9000)) = 50$ ,  $t_{Go}(SH\_f(A, 9000)) = 60$  sowie  $t_{Go}(Rück\_f(A, 9000)) = 80$ . Als der frühestmögliche Zeitpunkt des Beginns der Entscheidung über eine Überprüfung wird das Maximum dieser Werte,  $t_{Go}(Rück\_f(A, 9000)) = 80$ , gewählt, entsprechend ergeben sich die folgenden echten Wartezeiten:  $t(DB\_f(A, 9000)) = 30$  und  $t(SH\_f(A, 9000)) = 20$ .
- Da in dem Ablauf keine weiteren rückwärtsverzweigten Ereignisse existieren, werden die  $t_{Go}$ -Werte der restlichen Bedingungen sukzessive durch Addition der Zeitbewertungen der Bedingungen erhöht. Da darüber hinaus der maximale Schnitt des Ablaufs nur das Element  $C\_f(A, 9000)$  enthält, müssen die  $t_{Go}$ -Werte der Bedingungen im maximalen Schnitt nicht mehr auf Differenzen und dadurch eventuell anfallende echte Wartezeit hin untersucht werden.

Damit sind die Zeitbewertungen aller Bedingungen des Ablaufs bekannt (vgl. Abbildung 7.3).

Die echte Wartezeit, die bei der durch den Ablauf dargestellten Ausführung des Geschäftsprozesses anfällt, läßt sich als Summe der Zeitbewertungen der Bedingungen ermitteln, die auf externe Stellen abgebildet werden. Für die Summe der echten Wartezeit des Ablaufs  $A$  in Abbildung 7.2 ergibt sich damit:

$$\begin{aligned} t_{W_e}(A, 9000) &= t(\text{Best}_f(A, 9000)) + t(\text{DB}_f(A, 9000)) + t(\text{SH}_f(A, 9000)) \\ &= 20 + 30 + 20 = 70 \end{aligned}$$

Ausgehend von dem skizzierte Vorgehen bei der Ermittlung echter Wartezeit wird der folgende *Algorithmus zur Ermittlung echter Wartezeit* für ein zeitbewerteten halbgeordneten Ablauf formuliert:

### **Input**

Zeitbewerteter halbgeordneter Ablauf  $A = (B, E, K, \alpha, \beta, \lambda)$  mit unbekanntenen Werten für echte Wartezeit

### **Output**

Zeitbewertungen für alle Bedingungen

### **Initialisierung**

#Initialisierung der Menge der zu betrachtenden Bedingungen

**Bilde  $B_{Fl}$  =minimaler Schnitt von  $A$ .**

#Initialisierung des frühestmöglichen Schaltzeitpunkts für Ereignisse im Nachbereich des minimalen Schnitts

**Für alle  $b$  in  $B_{Fl}$  setze  $t_{Go}(b) := 0$ .**

#Initialisierung der echten Wartezeit als Zeitbewertung der Bedingungen, die auf externe Stellen abgebildet werden

**Für alle  $b$  in  $B$  mit:  $\alpha(b)$  ist externe Stelle setze  $t(b) := 0$ .**

## Ablauf

Solange  $B_{Fl}$  nicht leer:

Wähle eine Bedingung  $b'$  aus  $B_{Fl}$ .

Falls  $b'$  kein Element des maximalen Schnitts  $c_{max}$  von  $A$  ist:

Betrachte  $e'$  aus  $b' \bullet$ . # das Ereignis  $e'$  im Nachbereich von  $b'$

Falls  $|\bullet e'| = 1$ : #  $b'$  ist einziges Element im Vorbereich von  $e'$

# Berechnung der  $t_{Go}$ -Werte für den Nachbereich von  $e'$

Für alle  $b$  in  $e' \bullet$ :

setze  $t_{Go}(b) := t_{Go}(b') + t(b')$ .

# Austausch der Elemente von  $B_{Fl}$

Entferne  $b'$  aus  $B_{Fl}$ .

Füge alle  $b$  aus  $e' \bullet$  der Menge  $B_{Fl}$  hinzu.

Falls  $|\bullet e'| > 1$ : # mehr als ein Element im Vorbereich von  $e'$

Falls in  $\bullet e'$  ein  $b$  ohne  $t_{Go}$ -Wert existiert :

Entferne  $b'$  aus  $B_{Fl}$ .

Sonst: # für alle Bedingungen in  $\bullet e'$  ist ein  $t_{Go}$ -Wert ermittelt

# Berechnung eventuell entstehender echter Wartezeit als

# Zeitbewertung der Bedingungen im Vorbereich von  $e'$

Für alle  $b$  in  $\bullet e'$ :

setze  $t(b) := \max_{b \in \bullet e'}(t_{Go}(b)) - t_{Go}(b)$ .

# Berechnung der  $t_{Go}$ -Werte für den Nachbereich von  $e'$

Für alle  $b$  in  $e' \bullet$ :

setze  $t_{Go}(b) := \max_{b \in e' \bullet}(t_{Go}(b))$ .

# Austausch der Elemente von  $B_{Fl}$

Entferne  $b'$  aus  $B_{Fl}$ .

Füge alle  $b$  aus  $e' \bullet$  der Menge  $B_{Fl}$  hinzu.

Sonst: #  $b'$  ist im maximalen Schnitt von  $B, E, K$  enthalten

Falls  $|c_{max}| = 1$

Setze  $t(b') := 0$  und entferne  $b'$  aus  $B_{Fl}$ .

Falls  $|c_{max}| > 1$

Falls in  $c_{max}$  ein  $b$  ohne  $t_{Go}$ -Wert existiert:

Entferne  $b'$  aus  $B_{Fl}$ .

Sonst: # für alle Bedingungen in  $c_{max}$  ist ein  $t_{Go}$ -Wert ermittelt

Für alle  $b$  in  $c_{max}$ :

setze  $t(b) := \max_{b \in c_{max}}(t_{Go}(b)) - t_{Go}(b)$ .

Entferne  $b'$  aus  $B_{Fl}$ .

Wie eingangs bereits erwähnt, wird die echte Wartezeit bei dem hier vorgestellten Verfahren den Bedingungen zugeordnet, die im unmittelbaren Vorbereich des entsprechenden rückwärtsverzweigten Ereignisses liegen. Falls keine Wartekosten berücksichtigt werden, spielt die genaue Zuordnung der echten Wartezeit auch keine Rolle. Anders sieht es dagegen für den Fall aus, daß Wartezeit sehr wohl (für verschiedene Bedingungen unterschiedliche) Wartekosten verursacht. In Abschnitt 7.3.2 wird daher im Rahmen der Berechnung der echten Wartekosten eines Ablaufs ein Verfahren zur kostenoptimalen Verteilung echter Wartezeit entwickelt.

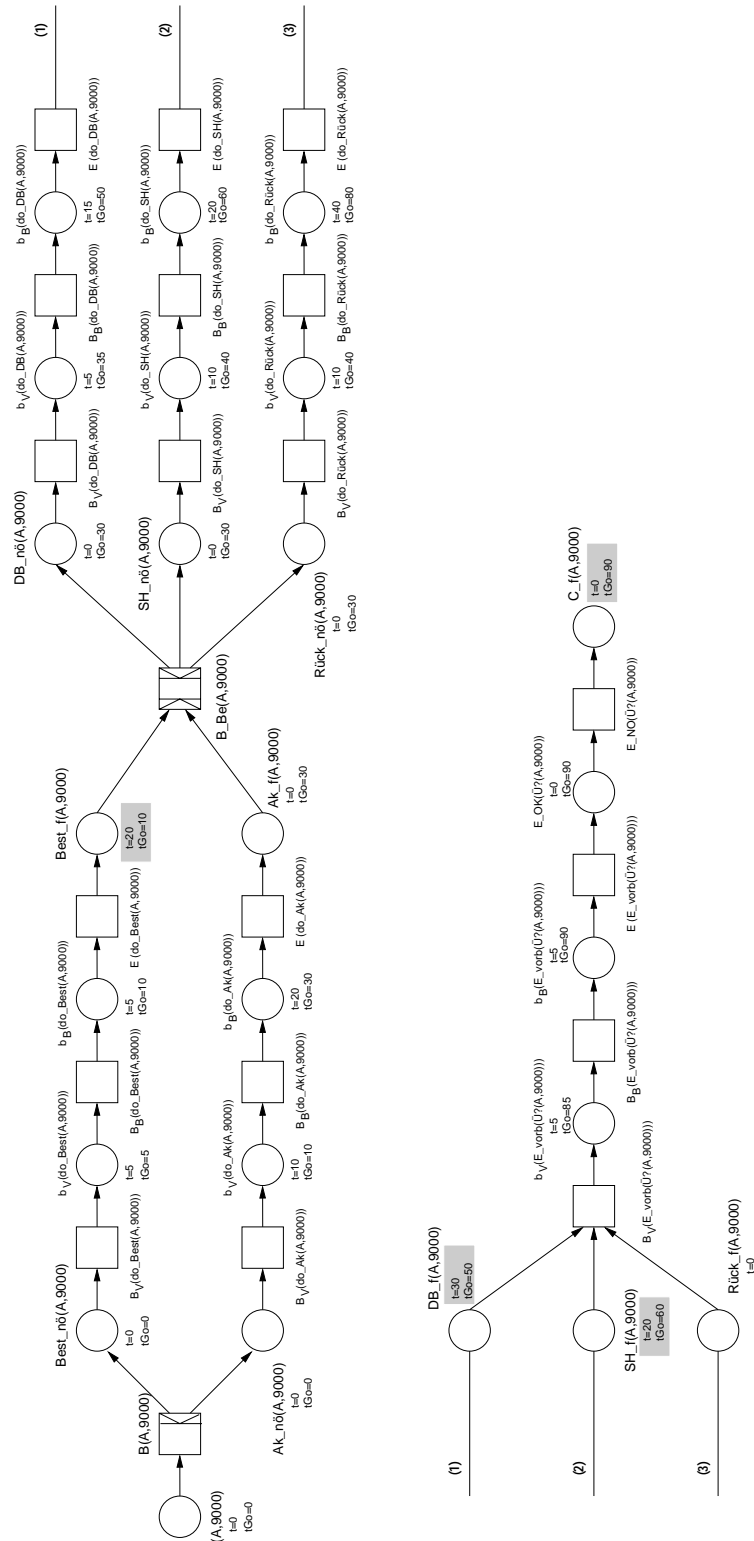


Abbildung 7.3: Ablauf aus Abbildung 7.2 mit  $t_{G0}$ -Werten und Wartezeiten.

### 7.2.3 Ermittlung der Durchlaufzeit

Mit der Ermittlung der echten Wartezeit sind alle Zeitgrößen für einen Ablauf bekannt. Damit kann nun auch die *minimale Durchlaufzeit* ermittelt werden, die zur Ausführung des Ablaufs benötigt wird. Die minimale Durchlaufzeit  $DLZt_{D_{min}}$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entspricht dem Maximum der  $t_{Go}$ -Werte im maximalen Schnitt  $c_{max}$  des Ablaufs  $A$ :

$$t_{D_{min}}(A) = \max_{b \in c_{max}} (t_{Go}(b))$$

#### **Beispiel**

Für den Beispiel-Ablauf in Abbildung 7.2 ergibt sich als minimale Durchlaufzeit:

$$t_{D_{min}}(A) = t_{Go}(C_{-}f(A, 9000)) = 90$$

□

### 7.2.4 Ressourcenbezogene Zeitgrößen

Auch wenn auf die mit der Berücksichtigung von Ressourcen verbundenen Besonderheiten wie *Linearisierungen* erst in Abschnitt 7.4 eingegangen wird, sollen in diesem Abschnitt bereits die Verfahren zur Ermittlung der beiden wichtigsten zeitbezogenen Kenngrößen in Bezug auf Ressourcen vorgestellt werden. Bei diesen Kenngrößen handelt es sich um den *Auslastungsgrad* sowie die *Nutzungsdauer* von Ressourcen. Durch die explizite Modellierung einzelner Ressourcen mittels unterscheidbarer Marken können diese Kenngrößen nicht nur für Ressourcen-Klassen, sondern auch für einzelne Ressourcen berechnet werden. Diese Möglichkeit ist einer der entscheidenden Vorteile des Ressourcen-Konzepts des *VIPbusiness*-Ansatzes gegenüber Techniken aus dem Bereich der Warteschlangen-Theorie (vgl. Abschnitt 3.1.3).

Zunächst wird der Ablauf aus Abbildung 7.2 um Ressourcen erweitert.

#### **Beispiel**

In Abbildung 7.4 ist basierend auf dem Geschäftsprozeßmodell in Abbildung 5.15 die Nutzung von Ressourcen der Klasse *Mitarbeiter* (MA) berücksichtigt worden. Dabei wurde die Bestätigung des Eingangs sowie die Einsicht der Datenbank von Mitarbeiter 1, das Anlegen der Akte und die Rücksprache mit dem Versicherungsnehmer von Mitarbeiter 2 durchgeführt.

□

#### **Ermittlung der Nutzungsdauer**

Die Möglichkeiten zur Ermittlung der Nutzungsdauer von Ressourcen in einem Ablauf hängen von der Art der Modellierung der Ressourcennutzung ab (vgl. Abschnitt 5.3). Bei der Modellierung der Nutzung von Ressourcen mittels interner Stellen läßt sich die Dauer der Nutzung über die Zeitbewertungen der Bedingungen ermitteln, die auf diese internen Stellen abgebildet werden. Für den Fall, daß Ressourcen in einem Ablauf mehrfach genutzt



werden, ergibt sich die Nutzungsdauer als Summe der Zeitbewertungen aller Bedingungen, die auf die entsprechenden internen Stellen abgebildet werden.

Sei mit  $A$  ein halbgeordneter Ablauf eines *VIPbusiness*-Netzes  $N$  gegeben. Sei  $T_{A_r}$  die Menge der Aktivitäts-Transitionen, die eine Ressource aus einer Ressourcen-Klasse  $r$  nutzen. Damit ergibt sich die *Nutzungsdauer*  $t_N(r)$  einer Ressourcen-Klasse  $r$  als:

$$t_N(r) = \sum_{\tau_A \in T_{A_r}} \left( \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \right)$$

### **Beispiel**

Für den Ablauf in Abbildung 7.4 ergibt sich als Nutzungsdauer der Ressourcen-Klasse **MA**:

$$\begin{aligned} t_N(MA) &= t(b_{Res}(do\_Best(1))) + t(b_{Res}(do\_DB(1))) \\ &\quad + t(b_{Res}(do\_Ak(2))) + t(b_{Res}(do\_Rück(2))) \\ &= 10 + 20 + 30 + 50 \\ &= 110 \end{aligned}$$

□

Um überprüfen zu können, ob sich die Nutzung einer Ressourcen-Klasse gleichmäßig auf alle Ressourcen der Klasse verteilt, ist neben der Nutzungsdauer der Ressourcen-Klasse auch die Nutzungsdauer für einzelne Ressourcen von Interesse.

Sei  $d_r \in D_r$  eine Ressource beziehungsweise ein *Ressourcen-Typ* (vgl. Abschnitt 5.3) der Ressourcen-Klasse  $r$ . Dann ergibt sich die *Nutzungsdauer*  $t_N(d_r)$  einer Ressource  $d_r$  als:

$$t_N(d_r) = \sum_{\tau_A \in T_{A_r}} \left( \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \text{ und } \lambda(b) = d_r \right)$$

### **Beispiel**

Für den Ablauf in Abbildung 7.4 ergibt sich als Nutzungsdauer der Ressource **1**:

$$\begin{aligned} t_N(1) &= t(b_{Res}(do\_Best(1))) + t(b_{Res}(do\_DB(1))) \\ &= 10 + 20 \\ &= 30 \end{aligned}$$

Für die Ressource **2** ergibt sich:

$$\begin{aligned} t_N(2) &= t(b_{Res}(do\_Ak(2))) + t(b_{Res}(do\_Rück(2))) \\ &= 30 + 50 \\ &= 80 \end{aligned}$$

□

Abschließend wird mit der Nutzungsdauer von Ressourcen *pro Aktivität* eine wichtige Kenngröße im Hinblick auf die Identifizierung kostentreibender Aktivitäten bei der Analyse von Kostengrößen in Abschnitt 7.3.3 eingeführt. Die Nutzungsdauer kann dabei nach Ressourcen-Klassen oder einzelnen Ressourcen(-Typen) aufgegliedert werden. Letzteres ist nur sinnvoll, falls mit dem Einsatz der einzelnen Ressourcen unterschiedliche Kosten verbunden sind.

Mit den bisher eingeführten Notationen ergibt sich als Nutzungsdauer einer Ressourcen-Klasse  $r$  durch eine Aktivität  $\tau_A$ :

$$t_N(r, \tau_A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_{Res}(\tau_A)$$

Entsprechend ergibt sich als Nutzungsdauer einer Ressource  $d_r$  durch eine Aktivität  $\tau_A$ :

$$t_N(d_r, \tau_A) = \sum_{b \in B} t(b) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \text{ und } \lambda(b) = d_r$$

### **Beispiel**

Exemplarisch sei die Aktivität **do\_Best** für den Ablauf in Abbildung 7.4 und das zugehörige Geschäftsprozeßmodell aus Abbildung 5.15 betrachtet. Als Nutzungsdauer der Ressourcen-Klasse MA durch **do\_Best** ergibt sich:

$$\begin{aligned} t_N(MA, do\_Best) &= t(b_{Res}(do\_Best(1))) \\ &= 10 \end{aligned}$$

Für die Nutzungsdauer der Mitarbeiter vom Typ 1 durch **do\_Best** ergibt sich:

$$\begin{aligned} t_N(1, do\_Best) &= t(b_{Res}(do\_Best(1))) \\ &= 10 \end{aligned}$$

Die Angabe der Nutzungsdauer für andere Ressourcen erübrigt sich, da durch Ausführung der Aktivität für den gegebenen Ablauf ausschließlich Mitarbeiter vom Typ 1 genutzt wurden. □

An dieser Stelle sei darauf hingewiesen, daß durch die Zeitbewertungen der Bedingungen, die dem Verbleib der die Ressource modellierenden Marke in der die Ressourcen-Klasse modellierenden Stelle entsprechen (und damit nicht-genutzte Ressourcen im Ablauf darstellen), im Prinzip echte variable Wartezeit modelliert wird. Da jedoch anders als bei den übrigen auf externe Stellen abgebildeten Bedingungen keine Variationsmöglichkeiten bei der Zuordnung dieser Wartezeit bestehen, spielen diese Zeitbewertungen für Verfahren wie beispielsweise die kostenoptimale Verteilung von echter Wartezeit in Abschnitt 7.3.2 keine Rolle.



## Ermittlung des Auslastungsgrads

Eine weitere wichtige Kenngröße bei der Leistungsbewertung eines Geschäftsprozeß-Entwurfs ist der Auslastungsgrad der Ressourcen. Dieser ergibt sich als Verhältnis der Nutzungsdauer der Ressourcen zur (minimalen) Durchlaufzeit der zur Bearbeitung von Cases gehörenden Abläufe. Der „optimale“ Auslastungsgrad variiert mit der Art der Ressource. Während eine Maschine durchaus zu 100% ausgelastet werden kann, liegt bei Mitarbeitern diese Größe je nach Art der auszuführenden Tätigkeit deutlich niedriger.

Wie bei der Nutzungsdauer kann auch der Auslastungsgrad für Ressourcen-Klassen oder einzelne Ressourcen ermittelt werden. Bei dem Auslastungsgrad einer Ressourcen-Klasse handelt es sich um den durchschnittlichen Auslastungsgrads einer Ressource dieser Klasse.

Sei mit  $A$  ein halbgeordneter Ablauf eines *VIPbusiness*-Netzes gegeben. Sei mit  $t_{D_{min}}(A)$  die (minimale) Durchlaufzeit des Ablaufs und mit  $t_N(r)$  die Nutzungsdauer einer Ressourcen-Klasse  $r$  gegeben. Mit  $\sum_{d_r \in D_r} \mu_{0,r}(d_r)$  als Anzahl der anfangs verfügbaren Ressourcen ergibt sich als *durchschnittlicher Auslastungsgrad*  $l_r$  der Ressourcen-Klasse  $r$  für den Ablauf  $A$ :

$$l_r(r) = \frac{1}{\sum_{d_r \in D_r} \mu_{0,r}(d_r)} \frac{t_N(r)}{t_{D_{min}}(A)}$$

### Beispiel

Für den Ablauf in Abbildung 7.4 ergibt sich als durchschnittlicher Auslastungsgrad einer Ressource der Ressourcen-Klasse **MA**:

$$\begin{aligned} l_r(MA) &= \frac{1}{2} \frac{110}{90} \\ &= 0.61 \end{aligned}$$

□

Sei  $d_r \in D_r$  eine Ressource beziehungsweise ein Ressourcen-Typ der Ressourcen-Klasse  $r$ . Sei  $\mu_{0,r}(d_r)$  die Anzahl der anfangs verfügbaren Ressourcen. Dann ergibt sich der *Auslastungsgrad*  $l_r(d_r)$  einer Ressource  $d_r$  als:

$$l_r(d_r) = \frac{1}{\mu_{0,r}(d_r)} \frac{t_N(r)}{t_{D_{min}}(A)}$$

### Beispiel

Für den Ablauf in Abbildung 7.4 ergibt sich als Auslastungsgrad der Ressource 1:

$$\begin{aligned} l_r(1) &= \frac{1}{1} \frac{30}{90} \\ &= 0.33 \end{aligned}$$

Für die Ressource 2 ergibt sich:

$$\begin{aligned}l_r(1) &= \frac{1\ 80}{1\ 90} \\ &= 0.88\end{aligned}$$

Es zeigt sich also, daß die Arbeit ungleich auf die Mitarbeiter vom Typ 1 und 2 verteilt ist. Die Mitarbeiter vom Typ 2 stellen dabei einen potentiellen Engpaß dar. Für den Fall, daß eine große Zahl von Cases zu bearbeiten ist, kann es damit zu einer Verlängerung der minimalen Durchlaufzeit aufgrund nicht verfügbarer Ressourcen kommen. Bei einer bloßen Betrachtung der Ressourcen-Klasse ohne explizite Unterscheidung einzelner Ressourcen wäre dieses Problem aufgrund des gemäßigt erscheinenden Auslastungsgrads von 61 % nicht erkannt worden.

□

### 7.2.5 Abgeleitete Zeitgrößen

Die in den vorangegangenen Abschnitten ermittelten Kenngrößen können als Ausgangspunkt zur Ermittlung weiterer Kenngrößen genutzt werden. Wie in Kapitel 2 bereits angedeutet wurde, existieren prinzipiell beliebig viele Kenngrößen, die untersucht werden können. In diesem Abschnitt werden zunächst mit einer Reihe von *Verhältnisgrößen*<sup>9</sup>, beispielsweise das Verhältnis von Aktivitätszeit und Durchlaufzeit eines Ablaufs, aggregierte Kenngrößen vorgestellt, die einen wichtigen Indikator für die Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs darstellen. Für den Fall, daß durch diese Indikatoren ein Problem angedeutet wird, ist eine detaillierte Analyse einzelner Zeitgrößen des Ablaufs notwendig. Dazu wird im zweiten Teil dieses Abschnitts eine Tabelle eingeführt, die eine systematische Aufbereitung detaillierter Zeitgrößen für einzelne Aktivitäten ermöglicht.

#### Verhältnisgrößen

Verhältnisgrößen stellen aggregierte Kenngrößen dar, deren Interpretation auf kompakte Weise eine Einschätzung der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs ermöglicht. Zwei bekannte Verhältnisgrößen sind das Verhältnis der Durchlaufzeit eines Ablaufs zur Aktivitätszeit sowie zur echten Wartezeit:

- (Echte) Wartezeit vs. Durchlaufzeit:  $\frac{t_{W_e}(A)}{t_{D_{min}}(A)}$

Das Verhältnis der echten Wartezeit eines Ablaufs zu seiner Durchlaufzeit ist eine wichtige Größe zur Beurteilung der Effizienz bei der Bearbeitung von Cases. Da das Auftreten echter Wartezeit nach Möglichkeit zu vermeiden ist, sollte  $\frac{t_{W_e}(A)}{t_{D_{min}}(A)}$  im Idealfall gegen 0 gehen. In der Realität kann es dagegen durchaus vorkommen, daß dieser Wert gegen 1 oder sogar darüber hinaus geht<sup>10</sup>. Ein Beispiel ist die Bearbeitung von Steuererklärungen durch das Finanzamt, dort steht Durchlaufzeit von zwei bis drei Monaten eine Aktivitätszeit von schätzungsweise 30 Minuten gegenüber, der Rest ist echte Wartezeit. Aus Sicht des Kunden stellt ein solcher Geschäftsprozeß-Entwurf sicherlich keine befriedigende Lösung dar.

---

<sup>9</sup>auch: *Ratios* [DE01].

<sup>10</sup>Dieser Fall kann vorkommen, wenn nebenläufig gewartet wird.

- Aktivitätszeit vs. Durchlaufzeit:  $\frac{t_A(A)}{t_{D_{min}}(A)}$

Das Verhältnis der Aktivitätszeit eines Ablaufs zu seiner Durchlaufzeit verbindet die Effekte nebenläufiger Ausführung von Aktivitäten mit dem Auftreten von Wartezeit. Eine Ablauf, dessen Aktivitätszeit deutlich größer als die Durchlaufzeit ist, zeichnet sich durch einen hohen Anteil nebenläufig ausführbarer Aktivitäten aus. Ein solcher Ablauf kann gegebenenfalls als „Vorbild“ genutzt werden: der Geschäftsprozeß wird so gestaltet, daß möglichst viele Cases gemäß Ausführung der Aktivitäten für diesen Ablauf bearbeitet werden (vgl. Abschnitt 7.5).

Falls  $\frac{t_A(A)}{t_{D_{min}}(A)}$  für einen Ablauf kleiner 1 ist, tritt echte Wartezeit auf. In diesem Fall ist es sinnvoll, die einzelnen Werte für echte Wartezeit und ihr Auftreten zu überprüfen, insbesondere wenn wie oben beschrieben das Verhältnis echter Wartezeit zur Durchlaufzeit vergleichsweise groß ausfällt.

### **Beispiel**

Für den Ablauf in Abbildung 7.4 ergibt sich als Verhältnis von echter Wartezeit zur Durchlaufzeit:

$$\frac{t_{W_e}(A)}{t_{D_{min}}(A)} = \frac{70}{90} = 0.77$$

Ein Blick auf den Ablauf zeigt, daß echte Wartezeit zum einen durch die Synchronisation der Ausführung nach Vorbereitung der Bearbeitung der Schadensmeldung entsteht. Der Großteil der Wartezeit resultiert jedoch aus der Synchronisation der drei nebenläufigen ausgeführten Aktivitäten zur eigentlichen Bearbeitung des Case.

Für das Verhältnis der Aktivitätszeit zur Durchlaufzeit ergibt sich:

$$\frac{t_A(A)}{t_{D_{min}}(A)} = \frac{150}{90} = 1.67$$

Hier zeigen sich die Vorteile der nebenläufigen Ausführung von Aktivitäten. Bei sequentieller Ausführung aller notwendigen Aktivitäten wäre die minimale Durchlaufzeit des Ablaufs um den Faktor 1.67 höher.

□

### **Detaillierte Analyse einzelner Zeitwerte**

Die bisher ermittelten Zeitgrößen eignen sich zu einer ersten Beurteilung der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs anhand einzelner Abläufe. Ebenso wichtig wie die Ermittlung dieser Zeitgrößen ist die Ermittlung der Ursachen ihres Zustandekommens. Dazu müssen die einzelnen Aktivitäten des Geschäftsprozesse betrachtet werden. Dies geschieht mit Hilfe einer für jeden Ablauf  $A$  aufgestellten Tabelle, in der für jede Aktivität  $\tau_A$  die folgenden Zeitgrößen eingetragen werden:

- Transportzeit  $t_T(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Rüstzeit  $t_R(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- (*alternativ*) Vorbereitungszeit  $t_V(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Bearbeitungszeit  $t_B(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- (*optional*) fertigungsbedingte Wartezeit  $t_{W_f}(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Aktivitätszeit  $t_A(\tau_A)$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Anteil der  $\tau_A$  zurechenbaren Aktivitätszeit an der Aktivitätszeit  $t_A(A)$  des Ablaufs,
- Anteil der  $\tau_A$  zurechenbaren Aktivitätszeit an der Durchlaufzeit  $t_{D_{min}}(A)$  des Ablaufs.

Anhand dieser Tabelle kann beispielsweise ermittelt werden, welche Aktivitäten große Anteile an der Durchlaufzeit eines Ablaufs haben. Diese Information kann in einem zweiten Schritt genutzt werden, um die Reihenfolge der Aktivitäten im Sinne einer erhöhten Nebenläufigkeit umzugestalten. Darüber hinaus sollten die bei der Modellierung verwendeten Schätz- oder Sollwerte für diese Aktivitäten besonders kritisch überprüft werden.

### Beispiel

In Tabelle 7.1 sind für alle Aktivitäten des zum Ablauf in Abbildung 7.4 gehörenden Geschäftsprozeßmodells die einzelnen Zeitgrößen eingetragen worden. Es zeigt sich, daß die Rückfrage beim Kunden die meiste Zeit in Anspruch nimmt. Für den Fall, daß die Leistungsbewertung des Geschäftsprozeß-Entwurfs nicht den Erwartungen entspricht, bietet diese Aktivität das größte Potential bezüglich einer Reduzierung von Aktivitätszeit. Ein Vergleich der Aktivitätszeiten der unterschiedlichen Aktivitäten zeigt, daß die Aktivitätszeit sowie ihre Bestandteile ansonsten einigermaßen gleichmäßig auf alle Aktivitäten verteilt sind. Für diesen Ablauf existiert also keine einzelne „wichtigste“ Aktivität.

	Eingang bestätigen	Akte anlegen	Datenbank einsehen	Schadenshöhe ermitteln	Rückfrage	Überprüfung?
$t_V(\tau_A)$	5	10	5	10	10	5
$t_B(\tau_A)$	5	20	15	20	40	5
$t_A(\tau_A)$	10	30	20	30	50	10
$\%t_A(A)$	6.7	20.0	13.3	20.0	33.3	6.7
$\%t_D(A)$	11.1	33.3	22.2	33.3	55.5	11.1

Tabelle 7.1: Detaillierte Analyse der Zeitgrößen für Aktivitäten.

□

Das Auftreten echter Wartezeit ist einer der wichtigsten Anhaltspunkte für die mögliche Reduzierung der Durchlaufzeit eines Ablaufs. Daher wird für externe Stellen eine Tabelle

zur detaillierten Analyse der Verteilung echter Wartezeit aufgestellt. In diese werden für jede externe Stelle  $s$  die folgenden Zeitgrößen eingetragen:

- die Summe der echten Wartezeit, die auf einer externen Stelle angefallen ist, als Summe der echten Wartezeit  $t_{W_e}(b)$  für alle Bedingungen mit  $\alpha(b) = s$ ,
- Anteil der  $s$  zurechenbaren Wartezeit an der echten Wartezeit  $t_{W_e}(A)$  des Ablaufs,
- Anteil der  $\tau_A$  zurechenbaren Wartezeit an der Durchlaufzeit  $t_{D_{min}}(A)$  des Ablaufs.

Auch hier gibt die Analyse der einzelnen, insbesondere der minimalen und maximalen, Werte aufschlußreiche Hinweise für die Gestaltung des zugrundeliegenden Geschäftsprozeß-Entwurfs. Da eine sinnvolle Zuordnung der echten Wartezeit erst mit Einführung einer entsprechenden Kostenfunktion möglich ist (vgl. Abschnitt 7.2.2), findet sich das zugehörige Beispiel erst in Abschnitt 7.3.5.

Für den Fall, daß es sich bei den externen Stellen um Ressourcen-Stellen handelt, bietet die Tabelle zusätzlich Aufschluß über die Nutzung beziehungsweise die Nicht-Nutzung von Ressourcen<sup>11</sup>.

Die in diesem Abschnitt ermittelten abgeleiteten Zeitgrößen können darüber hinaus als Ausgangspunkt für die Anwendung weiterer, spezieller Analyseverfahren genutzt werden, beispielsweise zur Identifikation und optimalen Anordnung sogenannter *Knock-Out-Aktivitäten*<sup>12</sup> in [Aal98c].

### 7.3 Methoden zur Kostenanalyse

In 2.2.3 wurden Kosten als zeitabhängige Größe eingeführt. Analog zum Zeitkonzept mit seiner Unterscheidung zwischen Aktivitätszeit und echter Wartezeit ergibt sich damit eine Aufspaltung in Aktivitätskosten und echte Wartekosten. Da es sich bei den Werten für Aktivitätszeit um bekannte oder geschätzte, in jedem Fall aber feste Größen handelt, stellen Aktivitätskosten für Abläufe ebenfalls feste Größen dar. Die Verfahren zur Ermittlung der entsprechenden Kostengrößen werden in Abschnitt 7.3.1 vorgestellt. Der Variabilität der echten Wartezeit entsprechend handelt es sich bei echten Wartekosten auch um variable Größen, deren Werte für einzelne Abläufe ermittelt werden können. Da die durch echte Wartezeit verursachten Kosten von der Verteilung dieser Wartezeit abhängen, wird in Abschnitt 7.3.2 ein Verfahren zur kostenoptimalen Verteilung von Wartezeit eingeführt. Die Kosten für Ressourcen sind in den Kostenfunktionen der entsprechenden (Teil-)Aktivitäten bereits enthalten. Um Aufschluß über den Anteil der Ressourcenkosten an den Aktivitätskosten zu erhalten, werden die Ressourcenkosten mittels der Bedingungen, die auf entsprechende, Ressourcenklassen repräsentierende Stellen abgebildet werden, in Abschnitt 7.3.3 noch gesondert betrachtet. In 7.3.4 werden die Gesamtkosten eines Ablaufs berechnet, bevor in Abschnitt 7.3.5 die Ermittlung und Interpretation einiger abgeleiteter Kostengrößen behandelt wird.

---

<sup>11</sup>Es sei an dieser Stelle daran erinnert, daß die auf Ressourcen-Stellen anfallende echte Wartezeit bei der Ermittlung der echten Wartezeit eines Ablaufs *nicht* berücksichtigt wird (vgl. Abschnitt 7.2.2).

<sup>12</sup>Bei *Knock-Out-Aktivitäten* handelt es sich um Aktivitäten, die über die weitere Bearbeitung eines Case entscheiden, beispielsweise die Entscheidung über eine detaillierte Prüfung der Bearbeitung der Schadensmeldung in unserem Beispiel.

### 7.3.1 Ermittlung der Aktivitätskosten

Analog zu der Zusammensetzung von Aktivitätszeit ergeben sich für die Kosten von Aktivitäten die folgenden Bestandteile (vgl. Abschnitt 2.2.3 und 2.2.3):

- Transportkosten,
- Rüstkosten,
- Bearbeitungskosten sowie
- fertigungsbedingte Wartekosten.

Für Dienstleistungs-Aktivitäten, bei denen es nicht um die Erstellung physischer Produkte geht, kann es sinnvoll sein, Transport- und Rüstkosten zu *Vorbereitungskosten* zusammenzufassen und auf fertigungsbedingte Wartekosten zu verzichten. In diesem Fall werden damit die folgenden Bestandteile berücksichtigt:

- Vorbereitungskosten,
- Bearbeitungskosten.

#### Transportkosten

Die zur Ermittlung der Transportkosten eines Ablaufs relevanten Kostengrößen sind die Kostenbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_T(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. dazu auch Abschnitt 6.3.1). Die Transportkosten  $k_T$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entsprechen der Summe der Kostenbewertungen aller entsprechenden Bedingungen  $b$ :

$$k_T(A) = \sum_{b \in B} k(b) \text{ mit } \alpha(b) = s_T(\tau_A)$$

#### Rüstkosten

Die zur Ermittlung der Rüstkosten eines Ablaufs relevanten Kostengrößen sind die Kostenbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_R(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. Abschnitt 6.3.1). Die Rüstkosten  $k_R$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entsprechen der Summe der Kostenbewertungen aller entsprechenden Bedingungen  $b$ :

$$k_R(A) = \sum_{b \in B} k(b) \text{ mit } \alpha(b) = s_R(\tau_A)$$

#### Vorbereitungskosten

Die zur Ermittlung der Vorbereitungskosten eines Ablaufs relevanten Kostengrößen sind die Kostenbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_V(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. Abschnitt 6.3.1). Die Vorbereitungskosten  $k_V$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entsprechen der Summe der Kostenbewertungen aller entsprechenden Bedingungen  $b$ :

$$k_V(A) = \sum_{b \in B} k(b) \text{ mit } \alpha(b) = s_V(\tau_A)$$

### **Beispiel**

Für den Ablauf  $A$  in Abbildung 7.2 ergeben sich als Vorbereitungskosten

$$\begin{aligned} k_V(A) &= k(s_V(do\_Best(A, 9000))) + k(s_V(do\_Ak(A, 9000))) + k(s_V(do\_DB(A, 9000))) \\ &\quad + k(s_V(do\_SH(A, 9000))) + k(s_V(do\_Rück(A, 9000))) \\ &\quad + k(s_V(E\_vorb(\ddot{U}?(A, 9000)))) \\ &= 10 + 5 + 10 + 10 + 10 + 5 = 50 \end{aligned}$$

□

### **Bearbeitungskosten**

Die zur Ermittlung der Bearbeitungskosten eines Ablaufs relevanten Kostengrößen sind die Kostenbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_B(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. Abschnitt 6.3.1). Die Bearbeitungskosten  $k_B$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entsprechen der Summe der Kostenbewertungen aller entsprechenden Bedingungen  $b$ :

$$k_B(A) = \sum_{b \in B} k(b) \text{ mit } \alpha(b) = s_B(\tau_A)$$

### **Beispiel**

Für den Ablauf  $A$  in Abbildung 7.2 ergeben sich als Bearbeitungskosten

$$\begin{aligned} k_B(A) &= k(s_B(do\_Best(A, 9000))) + k(s_B(do\_Ak(A, 9000))) + k(s_B(do\_DB(A, 9000))) \\ &\quad + k(s_B(do\_SH(A, 9000))) + k(s_B(do\_Rück(A, 9000))) \\ &\quad + k(s_B(E\_vorb(\ddot{U}?(A, 9000)))) \\ &= 20 + 40 + 20 + 30 + 60 + 15 = 185 \end{aligned}$$

□

### **Fertigungsbedingte Wartekosten**

Die zur Ermittlung der fertigungsbedingten Wartekosten eines Ablaufs relevanten Kostengrößen sind die Kostenbewertungen der Bedingungen  $b$ , die auf die entsprechenden internen Stellen  $s_{W_f}(\tau_A)$  einer Aktivitätstransition  $\tau_A$  eines *VIPbusiness*-Netzes abgebildet werden (vgl. Abschnitt 6.3.1). Die fertigungsbedingten Wartekosten  $k_{W_f}$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  entsprechen der Summe der Kostenbewertungen aller entsprechenden Bedingungen  $b$ :

$$k_{W_f}(A) = \sum_{b \in B} k(b) \text{ mit } \alpha(b) = s_{w_f}(\tau_A)$$





## Aktivitätskosten

Die Aktivitätskosten  $k_A(\tau_A)$  einer Aktivitäts-Transition  $\tau_A$  ergeben sich als Summe der Kostenbewertungen der internen Stellen  $s_T(\tau_A), s_R(\tau_A), s_B(\tau_A), s_{W_f}(\tau_A)$  beziehungsweise  $s_V(\tau_A), s_B(\tau_A)$ . Entsprechend ergibt sich für die Kostenbewertung  $k(e_A)$  eines Ereignisses  $e_A$ , das auf eine Aktivitäts-Transition  $\tau_A$  abgebildet wird (vgl. Abschnitt 6.3.1):

$$k_A(e_A) = k(b_1) + k(b_2) + k(b_3) + k(b_4) \\ \text{mit } \alpha(b_1) = s_T(\tau_A), \alpha(b_2) = s_R(\tau_A), \alpha(b_3) = s_B(\tau_A), \alpha(b_4) = s_{w_f}(\tau_A)$$

beziehungsweise

$$k_A(e_A) = k(b_1) + k(b_2) \\ \text{mit } \alpha(b_1) = s_V(\tau_A), \alpha(b_2) = s_B(\tau_A)$$

Die Aktivitätskosten  $k_A(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  können jetzt als Summe der Kostenbewertungen  $k(e_A)$  aller Ereignisse  $e_A$ , die auf eine Aktivitäts-Transition in vergrößernder Sichtweise abgebildet werden, berechnet werden:

$$k_A(A) = \sum_{e_A \in E} k(e_A) \text{ mit } \beta(e_A) \text{ ist eine Aktivitäts-Transition.}$$

Alternativ lassen sich die Aktivitätskosten  $k_A(A)$  eines Ablaufs  $A = (B, E, K, \alpha, \beta, \lambda)$  auch als Summe der bereits bekannten Summen der Bestandteile der Aktivitätskosten ermitteln:

$$k_A(A) = k_T(A) + k_R(A) + k_B(A) + k_{W_f}(A)$$

beziehungsweise

$$k_A(A) = k_V(A) + k_B(A)$$

### Beispiel

Für den Ablauf  $A$  in Abbildung 7.2 ergeben sich als Aktivitätskosten

$$k_A(A) = k_V(A) + k_B(A) = 50 + 185 = 235$$

□

### 7.3.2 Ermittlung der echten Wartekosten

Im Mittelpunkt der Kostenanalyse von Geschäftsprozessen im Rahmen dieser Arbeit steht die Ermittlung der Kosten, die durch echte Wartezeit entstehen. Wieviel echte Wartezeit bei der Ausführung eines Geschäftsprozesses entsteht, wurde in Abschnitt 7.2.2 anhand des der Ausführung entsprechenden nebenläufigen Ablaufs ermittelt. Die anfallende echte Wartezeit wurde dabei willkürlich auf Bedingungen im direkten Vorbereich rückwärtsverzweigter Ereignisse verteilt. Die auf diese Weise vorgenommene Verteilung der Kosten, die durch echte Wartezeit entstehen, ist in der Regel nicht optimal. Daher wird in Abschnitt 7.3.2 ein Verfahren zur kostenoptimalen Verteilung echter Wartezeit entwickelt. Die Ermittlung echter Wartekosten als einziger variabler Kostengröße ist eine wichtige Voraussetzung zur Berechnung der Gesamtkosten für die Ausführung eines Geschäftsprozesses in Abschnitt 7.3.4.

Echte Wartezeit ist bisher den Bedingungen zugeordnet worden, die im direkten Vorbereich des rückwärtsverzweigten Ereignisses liegen (vgl. Abschnitt 7.2.2). Da Kostenaspekte bei der Ermittlung der minimalen Durchlaufzeit keine Rolle spielen, verursachte diese (willkürliche) Zuordnung keine Probleme. Anders sieht es für den Fall aus, daß die Kosten, die durch echte Wartezeit in realen Geschäftsprozessen in der Regel entstehen (z.B. Lagerkosten), bei der Verteilung dieser Wartezeit berücksichtigt werden sollen. Je nachdem, welche Kosten durch Wartezeit an den verschiedenen Stellen eines Systems entstehen, kann es günstiger sein, Aktivitäten schnellstmöglich auszuführen (sobald die entsprechenden Vorbedingungen für die Ausführung erfüllt sind), oder die Ausführung der Aktivität zu verzögern.

#### *Beispiel*

Dieser Sachverhalt kann anhand des Beispiels des kostenbewerteten Ablaufs in Abbildung 6.9 erläutert werden. Wie bei der Berechnung der echten Wartezeit in Abschnitt 7.2.2 gesehen, kommt es im Vorbereich des Beginns der eigentlichen Bearbeitung sowie im Vorbereich der Entscheidung über eine detaillierte Prüfung der Bearbeitung zu echter Wartezeit in Höhe von 20 beziehungsweise  $30 + 20 = 50$ .

Bisher wurde diese Wartezeit den entsprechenden Bedingungen im unmittelbaren Vorbereich des jeweiligen Ereignisses zugeordnet ( $Best\_f(A, 9000)$  beziehungsweise  $DB\_f(a, 9000)$  und  $SH\_f(A, 9000)$ ). Für die Bearbeitung der Schadensmeldung würde dies bedeuten, daß die Bestätigung des Eingangs sowie die Einsicht der Datenbank und die Ermittlung der Schadenshöhe unmittelbar nach Aktivierung der entsprechenden Aktivitäts-Transitionen ausgeführt wird, um anschließend auf die Beendigung des Anlegens der Akte beziehungsweise der Rückfrage beim Kunden warten zu müssen.

Durch die Einführung einer zeitabhängigen Kostenfunktion für alle externen Stellen (vgl. Abbildung 7.5) läßt sich nun die Frage nach einer kostenoptimalen Verteilung der anfallenden Wartezeit stellen. Durch Vergleich der relativen Wartekosten der externen Stellen **Datenbank nötig** und **Datenbank fertig** ( $5x_6$  vs.  $10x_7$ ) stellt sich beispielsweise heraus, daß es kostengünstiger ist, mit dem Einsehen der Datenbank solange zu warten, daß nach der Einsicht keine Wartezeit anfällt.

□

Die kostenoptimale Verteilung der echten Wartezeit läßt sich als *Optimierungsproblem* formulieren. Dieses Optimierungsproblem kann mit Hilfe der entsprechenden Verfahren aus dem Gebiet des *Operations Research* gelöst werden (vgl. beispielsweise [Gal87], [Mül73] oder [NM93]). Dazu werden eine *Zielfunktion* sowie eine Reihe von *Nebenbedingungen* benötigt. Ausgangspunkt für die Aufstellung des Optimierungsproblems ist ein zeit- und kostenbewerteter halbgeordneter Ablauf  $A$  eines *VIPbusiness*-Netzes  $N$ .

### Zielfunktion

Ziel der Optimierung ist die *Minimierung* der Kosten, die durch echte Wartezeit entstehen. Diese Kosten ergeben sich aus den Kostenbewertungen für Bedingungen, die auf externe Stellen des dazugehörigen Systemnetzes abgebildet werden.

Sei  $B_{ext}$  die Menge der Bedingungen aus  $B$ , die auf externe Stellen in  $S$  abgebildet werden. Damit ergibt sich als **Zielfunktion**:

$$k_{W_e}(A) = \sum_{b \in B_{ext}} k(t(b))$$

Diese Zielfunktion gilt es zu minimieren.

### Beispiel

Für den Ablauf  $A$  in Abbildung 7.5 ergibt sich als Zielfunktion:

$$\begin{aligned} k_{W_e}(A) = & 10x_1 + 15x_2 + 10x_3 + 10x_4 + 20x_5 + 5x_6 + 10x_7 + 5x_8 \\ & + 10x_9 + 15x_{10} + 10x_{11} + 10x_{12} \rightarrow \min \end{aligned}$$

### Nebenbedingung 1

Wartezeit entsteht im Vorbereich rückwärtsverzweigter Ereignisse. Für jedes rückwärtsverzweigte Ereignis muß sichergestellt werden, daß die Summe der Zeitbewertungen der Pfade, die zu diesem Ereignis führen, gleich ist.

Sei  $E_R$  die Menge aller rückwärtsverzweigter Ereignisse des Ablaufs  $A$ . Für jedes Ereignis  $e_R$  aus  $E_R$  sei  $B_{e_R}$  die Menge aller Bedingungen im Vorbereich von  $e_R$ . Für jedes  $b$  aus  $B_{e_R}$  sei  $p$  ein gerichteter Pfad vom minimalen Schnitt von  $A$  nach  $b$ . Sei  $L(b)$  die Menge der Elemente  $x$  aus  $B \cup E$ , für die gilt:  $x$  liegt auf  $p$ . Sei  $t(L(b))$  die Summe der Zeitbewertungen der Bedingungen in  $L(b)$ . Für je zwei Bedingungen  $b, b'$  aus  $B_{e_R}$  muß gelten:  $t(L(b)) = t(L(b'))$ .

Damit können folgende Nebenbedingungen aufgestellt werden:

**Für alle  $e_R$  in  $E_R$ :**

Für alle  $(b, b')$  aus  $(B_{e_R} \times B_{e_R})$  mit  $b \neq b'$ :  
Setze  $t(L(b))$  und  $t(L(b'))$  gleich.

### Beispiel

Der Ablauf in Abbildung 7.5 enthält zwei rückwärtsverzweigte Ereignisse:  $B\_Be(A, 9000)$  und  $B\_V(E\_vorb(\ddot{U}?(A, 9000)))$ . Daraus ergeben sich die Nebenbedingungen:

(NB 1)

für  $B\_Be(A, 9000)$

$$x_1 + x_2 + 5 + 5 + x_3 = x_1 + x_4 + 10 + 20 + x_5$$

für  $B\_V(E\_vorb(\ddot{U}?(A, 9000)))$

$$x_1 + x_2 + 5 + 5 + x_3 + x_6 + 5 + 15 + x_7 = x_1 + x_4 + 10 + 20 + x_5 + x_8 + 10 + 20 + x_9$$

$$x_1 + x_2 + 5 + 5 + x_3 + x_6 + 5 + 15 + x_7 = x_1 + x_4 + 10 + 20 + x_5 + x_{10} + 10 + 40 + x_{11}$$

$$x_1 + x_4 + 10 + 20 + x_5 + x_8 + 10 + 20 + x_9 = x_1 + x_4 + 10 + 20 + x_5 + x_{10} + 10 + 40 + x_{11}$$

□

### Nebenbedingung 2

Falls der maximale Schnitt (also das Ende) des Ablaufs mehr als eine Bedingung enthält, muß sichergestellt werden, daß die Summe der Zeitbewertungen bis zum Erreichen dieser Bedingungen für alle Bedingungen gleich ist. Kann eine der Bedingungen schneller erreicht werden, muß entsprechend echte Wartezeit verteilt werden, da der Ablauf erst beendet ist, wenn alle Bedingungen des Endes erreicht sind.

Sei  $c_{Ende}$  der maximale Schnitt des Ablaufs. Sei  $B_c$  die Menge aller Bedingungen in  $c_{Ende}$ . Für jedes  $b$  aus  $B_c$  sei  $p$  ein gerichteter Pfad vom minimalen Schnitt von  $A$  nach  $b$ . Sei  $L(b)$  die Menge der Elemente  $x$  aus  $B \cup E$ , für die gilt:  $x$  liegt auf  $p$ . Sei  $t(L(b))$  die Summe der Zeitbewertungen der Bedingungen in  $L(b)$ . Für je zwei Bedingungen  $b, b'$  aus  $B_c$  muß gelten:  $t(L(b)) = t(L(b'))$ .

Damit lassen sich folgende Nebenbedingungen aufstellen:

**Für alle  $(b, b')$  aus  $(B_c \times B_c)$  mit  $b \neq b'$ :**

Setze  $\tau(L(b))$  und  $\tau(L(b'))$  gleich.

### Beispiel

Der maximale Schnitt des Ablaufs in Abbildung 7.5 enthält nur ein Element. Damit entfällt diese Nebenbedingung.

□

### Nebenbedingung 3

Die minimale Durchlaufzeit des Ablaufs darf nicht überschritten werden.

Zur Formulierung dieser Nebenbedingung muß zunächst der Begriff des *kritischen Pfades*<sup>13</sup> eingeführt werden.

<sup>13</sup>In [NM93] findet sich auch der Begriff *kritischer Weg*.

**Definition 50 (Kritischer Pfad)**

Einen längsten<sup>14</sup> Weg in einem bewerteten Graphen bezeichnet man als *kritischen Pfad*. Die Vorgänge entlang dieses Weges werden *kritische Vorgänge* genannt.

□

Die Elemente des kritischen Pfades sind also die Aktivitäten, deren (durch die kausale Struktur des Graphen vorgegebene) Hintereinanderausführung am meisten Zeit in Anspruch nimmt. Daraus ergibt sich die wichtige Feststellung, daß für externe Stellen des Systemnetzes (und die entsprechenden Bedingungen im Ablauf), die auf dem kritischen Pfad liegen, keine echte Wartezeit anfallen kann, da sich dadurch die Durchlaufzeit des gesamten Ablaufs automatisch verlängern würde.

Auf die ausführliche Darstellung von Verfahren zur Ermittlung von kritischen Pfaden wird im Rahmen dieser Arbeit verzichtet. Es sei statt dessen auf die entsprechende Literatur aus dem Gebiet des *Operations Research* verwiesen (beispielsweise [NM93] oder [WW67]).

**Beispiel**

Der kritische Pfad des Beispiel-Ablaufs ist in Abbildung 7.6 eingezeichnet.

□

Sei  $B_{krit}$  die Menge der Bedingungen aus  $B$ , die auf externe Stellen aus  $S$  abgebildet werden und auf dem kritischen Pfad des Ablaufs liegen. Für alle  $b$  aus  $B_{krit}$  gilt, daß die Zeitbewertung echter Wartezeit entspricht. Da auf dem kritischen Pfad keine echte Wartezeit anfallen darf (s.o.), sind die Zeitbewertungen für alle  $b$  aus  $B_{krit}$  mit dem Wert 0 anzusetzen. Die Summe der Zeitbewertungen der übrigen Bedingungen auf dem kritischen Pfad entspricht der minimalen Durchlaufzeit des Ablaufs. Durch die Forderung der Nebenbedingung 2, daß die Summe der Zeitbewertungen der Pfade vom minimalen bis zum maximalen Schnitt gleich sein muß, ist sichergestellt, daß im Ablauf kein Pfad existiert, für den die Summe der Zeitbewertungen der Bedingungen, die auf diesem Pfad liegen, größer als die minimale Durchlaufzeit ist.

Die Forderung der Nebenbedingung 3, daß die kostenoptimale Verteilung der echten Wartezeit keine Verlängerung der minimalen Durchlaufzeit verursachen darf, läßt sich damit folgendermaßen formulieren:

**Für alle  $b$  in  $B_{krit}$ :**

Setze  $t(b) := 0$ .

---

<sup>14</sup>Der Begriff *längster* bezieht sich in diesem Zusammenhang auf die Summe der Zeitbewertungen entlang eines Pfades.

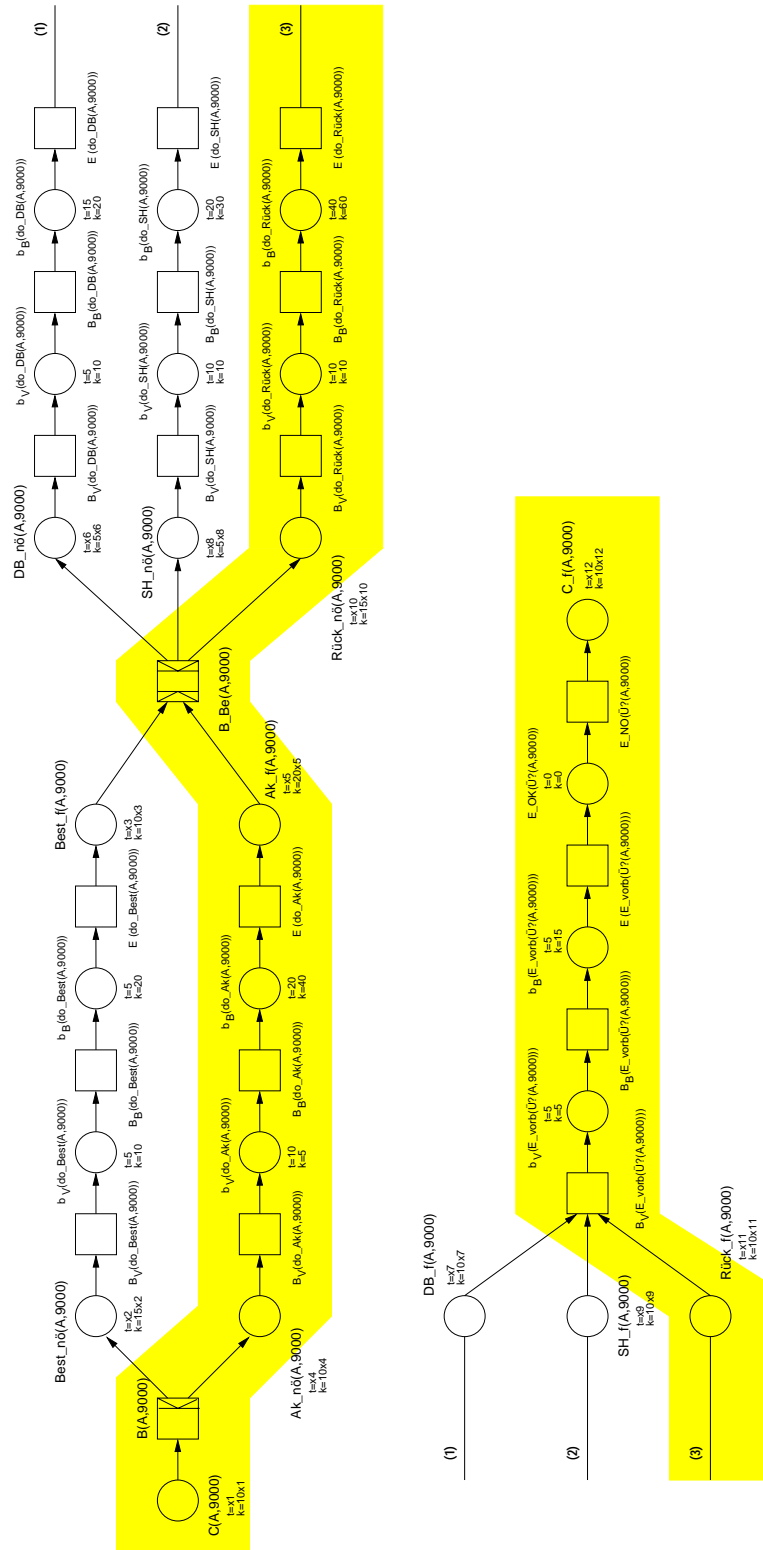


Abbildung 7.6: Kritischer Pfad.

### **Beispiel**

Für den Ablauf in Abbildung 7.5 ergibt sich:

(NB 3)

$$\begin{aligned}x_1 = 0 & \quad x_{10} = 0 \\x_4 = 0 & \quad x_{11} = 0 \\x_5 = 0 & \quad x_{12} = 0\end{aligned}$$

□

### **Nebenbedingung 4**

Negative Werte für die Zeitbewertungen von Bedingungen sind nicht zulässig:

**Für alle  $b$  in  $B$  muß gelten:**

$$t(b) \geq 0.$$

### **Beispiel**

Für den Ablauf in Abbildung 7.5 ergibt sich:

(NB 4)

$$x_i \geq 0 \text{ für alle } x_i \text{ mit } i = \{1, 2, \dots, 6\}$$

□

Zusammengefaßt läßt sich für den Ablauf aus Abbildung 7.5 folgendes Optimierungsproblem zur kostenoptimalen Verteilung von Wartezeit aufstellen:

Minimiere

$$\begin{aligned}k_{W_e}(A) = & 10x_1 + 15x_2 + 10x_3 + 10x_4 + 20x_5 + 5x_6 + 10x_7 + 5x_8 \\ & + 10x_9 + 15x_{10} + 10x_{11} + 10x_{12}\end{aligned}$$

unter den Nebenbedingungen:

(NB 1)

$$\begin{aligned}x_1 + x_2 + 5 + 5 + x_3 &= x_1 + x_4 + 10 + 20 + x_5 \\x_1 + x_2 + 5 + 5 + x_3 + x_6 + 5 + 15 + x_7 &= x_1 + x_4 + 10 + 20 + x_5 + x_8 + 10 + 20 + x_9 \\x_1 + x_2 + 5 + 5 + x_3 + x_6 + 5 + 15 + x_7 &= x_1 + x_4 + 10 + 20 + x_5 + x_{10} + 10 + 40 + x_{11} \\x_1 + x_4 + 10 + 20 + x_5 + x_8 + 10 + 20 + x_9 &= x_1 + x_4 + 10 + 20 + x_5 + x_{10} + 10 + 40 + x_{11}\end{aligned}$$

(NB 2)

entfällt

(NB 3)

$$x_1 = 0 \quad x_{10} = 0$$

$$x_4 = 0 \quad x_{11} = 0$$

$$x_5 = 0 \quad x_{12} = 0$$

(NB 4)

$$x_i \geq 0 \text{ für alle } x_i \text{ mit } i = \{1, 2, \dots, 6\}$$

Als Lösung ergibt sich:

$$k_{W_e}(A) = 450$$

$$\text{mit } x_1 = 0, x_2 = 0, x_3 = 20, x_4 = 0, x_5 = 0, x_6 = 30, x_7 = 0$$

$$x_8 = 20, x_9 = 0, x_{10} = 0, x_{11} = 0, x_{12} = 0$$

In Abbildung 7.7 ist die kostenoptimale Verteilung von Wartezeit für den Ablauf aus Abbildung 7.5 dargestellt.

### 7.3.3 Ressourcenbezogene Kostengrößen

Bei der Analyse der Kosten, die bei Ausführung eines Geschäftsprozeß-Entwurfs zur Bearbeitung von Cases entstehen, sind Kosten, die durch den Verbrauch oder die Inanspruchnahme von Ressourcen verursacht werden, von großem Interesse (vgl. Abschnitt 2.2.3). Wie bei der Ermittlung von ressourcenbezogenen Zeitgrößen können bei der Ermittlung ressourcenbezogener Kostengrößen verschiedene Bezugsgrößen zugrunde gelegt werden. In Analogie zur Ermittlung der Nutzungsdauern in Abschnitt 7.2.4 werden in diesem Abschnitt die *Nutzungskosten* sowohl für Ressourcen-Klassen als auch für einzelne Ressourcen, bezogen auf einen Ablauf oder einzelne Aktivitäten, ermittelt.

Dazu wird in Abbildung 7.8 der Ablauf aus Abbildung 7.5 um Ressourcen erweitert.

#### *Beispiel*

In Abbildung 7.8 ist basierend auf dem Geschäftsprozeßmodell in Abbildung 5.15 die Nutzung von Ressourcen der Klasse *Mitarbeiter* (MA) berücksichtigt worden.

Die Kostenbewertungen  $k(b_{Res})$  der Bedingungen  $b_{Res}$ , die auf Stellen  $s_{Res}$  abgebildet werden, entsprechen dem Anteil der durch Nutzung von Ressourcen verursachten Kosten an den gesamten Aktivitätskosten des zugehörigen Aktivitäts-Ereignisses.

□

Bei der Modellierung der Nutzung von Ressourcen mittels interner Stellen lassen sich die Kosten der Nutzung über die Kostenbewertungen der Bedingungen ermitteln, die auf interne Stellen  $s_{Res}(\tau_A)$  zur Modellierung der Ressourcennutzung durch eine Aktivitäts-Transition  $\tau_A$  abgebildet werden. Für den Fall, daß Ressourcen in einem Ablauf mehrfach



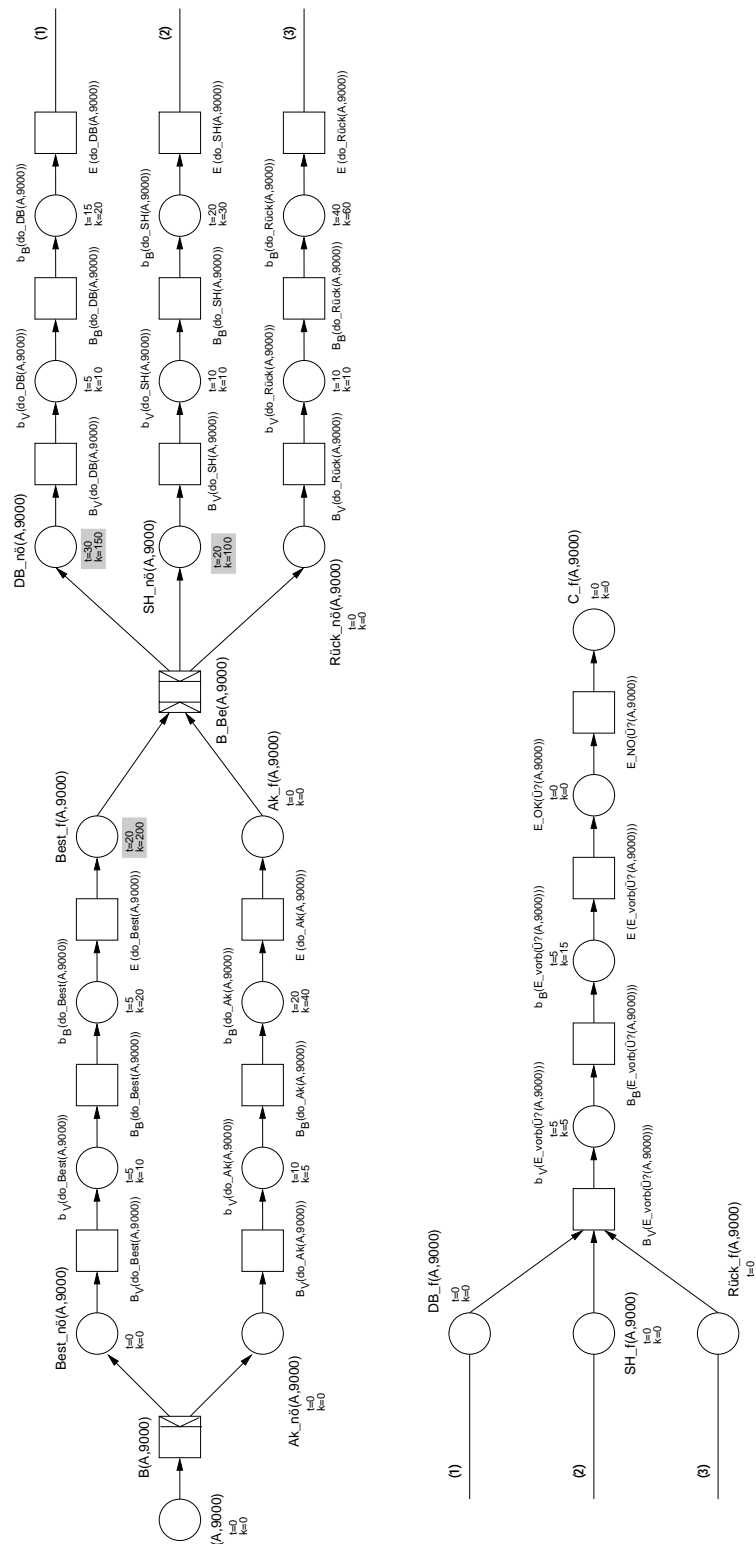


Abbildung 7.7: Kostenoptimale Verteilung von Wartezeit für den Ablauf aus Abbildung 7.5.

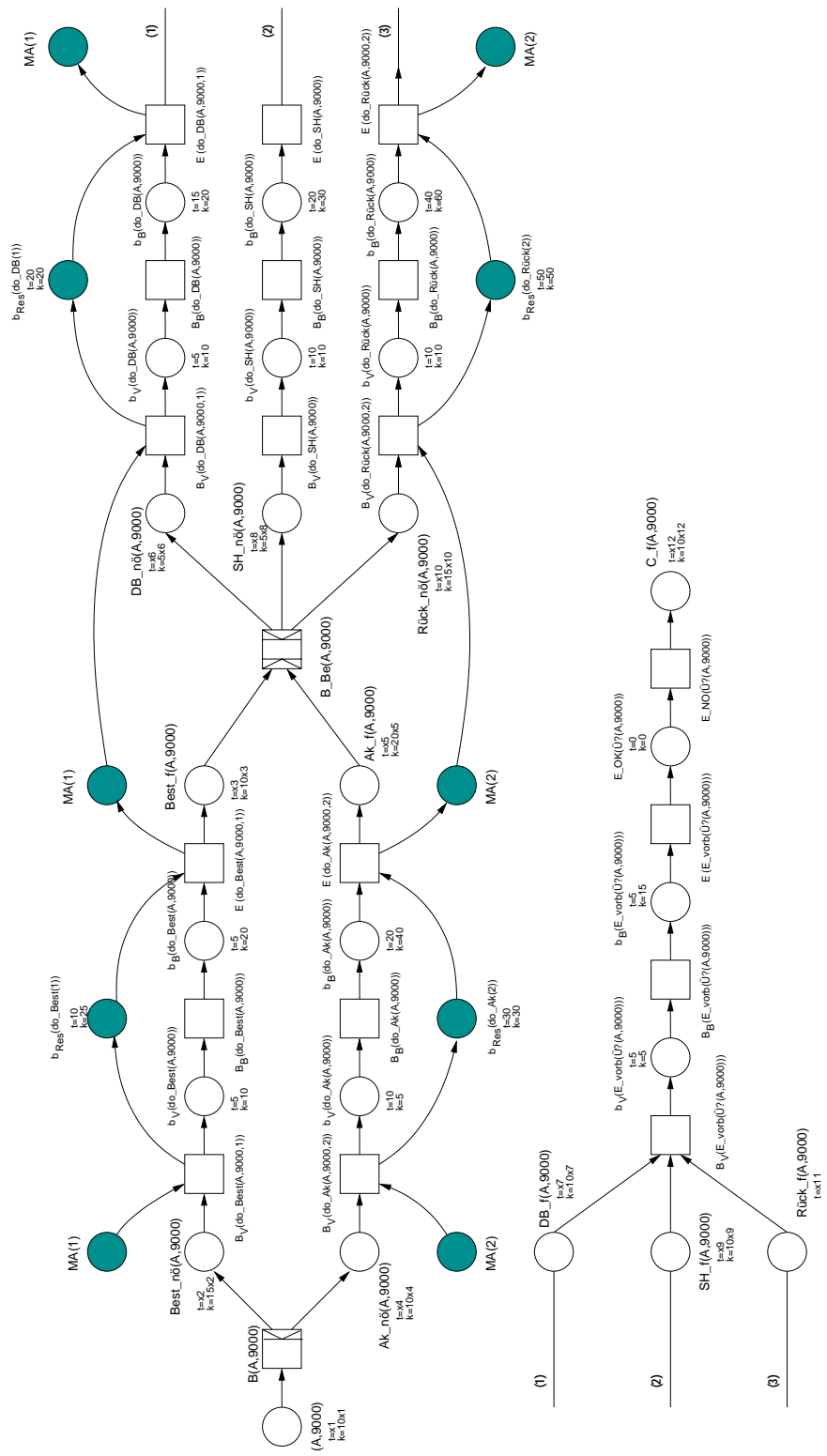


Abbildung 7.8: Kostenbewerteter Ablauf aus Abbildung 7.5 mit Nutzung von Ressourcen.

genutzt werden, ergeben sich die Nutzungskosten als Summe der Kostenbewertungen aller Bedingungen, die auf die entsprechenden internen Stellen abgebildet werden.

Sei mit  $A = ((B, E, K, A_1, A_2, \alpha, \beta, \lambda))$  ein halbgeordneter Ablauf eines *VIPbusiness-*Netzes  $N = (S, R, T, T_A, \{T_{AS}, T_{AJ}, T_{OS}, T_{OJ}\}, F, \mathcal{D}, m_0, \mathcal{P}, \Phi, NS)$  gegeben. Sei  $R \subset S$  die Menge der Ressourcen-Stellen. Sei  $T_{A_r}$  die Menge der Aktivitäts-Transitionen, die eine Ressource aus einer Ressourcen-Klasse  $r \in R$  nutzen. Damit ergeben sich die *Nutzungskosten*  $k_N(r)$  einer Ressourcen-Klasse  $r$  als:

$$k_N(r) = \sum_{\tau_A \in T_{A_r}} \left( \sum_{b \in B} k(b_{Res}) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \right)$$

Die *Ressourcenkosten*  $k_{Res}(A)$  eines Ablaufs als Summe der Nutzungskosten aller für die Ausführung der Aktivitäten eines Ablaufs benötigten Ressourcen ergeben sich als:

$$k_{Res}(A) = \sum_{r \in R} k_N(r)$$

### **Beispiel**

Für den Ablauf in Abbildung 7.8 ergeben sich als Nutzungskosten der Ressourcen-Klasse **MA**:

$$\begin{aligned} t_N(MA) &= k(b_{Res}(do\_Best(1))) + k(b_{Res}(do\_DB(1))) \\ &\quad + k(b_{Res}(do\_Ak(2))) + k(b_{Res}(do\_Rück(2))) \\ &= 25 + 20 + 30 + 50 = 125 \end{aligned}$$

□

Um ermitteln zu können, wie sich die Kosten der Nutzung einer Ressourcen-Klasse auf einzelne Ressourcen der Klasse verteilen, sind neben den Nutzungskosten der Ressourcen-Klasse auch die Nutzungskosten für einzelne Ressourcen von Interesse.

Sei  $d_r \in D_r$  eine Ressource beziehungsweise ein *Ressourcen-Typ* der Ressourcen-Klasse  $r$ . Dann ergeben sich die *Nutzungskosten*  $k_N(d_r)$  einer Ressource  $d_r$  als:

$$k_N(d_r) = \sum_{\tau_A \in T_{A_r}} \left( \sum_{b \in B} k(b_{Res}) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \text{ und } \lambda(b) = d_r \right)$$

### **Beispiel**

Für den Ablauf in Abbildung 7.4 ergeben sich als Nutzungskosten der Ressource **1**:

$$\begin{aligned} k_N(1) &= k(b_{Res}(do\_Best(1))) + k(b_{Res}(do\_DB(1))) \\ &= 25 + 20 = 45 \end{aligned}$$

Für die Ressource **2** ergibt sich:

$$\begin{aligned}
k_N(2) &= k(b_{Res}(do\_Ak(2))) + k(b_{Res}(do\_Rück(2))) \\
&= 30 + 50 = 80
\end{aligned}$$

□

Um eine genauere Vorstellung von der Kostenstruktur einzelner Aktivitäten zu bekommen, werden die mit der Nutzung von Ressourcen verbundenen Kosten für einzelne Aktivitäten ermittelt. Die Nutzungskosten können dabei wieder nach Ressourcen-Klassen oder einzelnen Ressourcen(-Typen) aufgegliedert werden. Letzteres ist nur sinnvoll, falls mit dem Einsatz der einzelnen Ressourcen unterschiedliche Kosten verbunden sind.

Mit den zu Beginn dieses Abschnitts eingeführten Notationen ergeben sich als Nutzungskosten einer Ressourcen-Klasse  $r$  durch eine Aktivität  $\tau_A$ :

$$k_N(r, \tau_A) = \sum_{b \in B} k(b_{Res}) \text{ mit } \alpha(b) = s_{Res}(\tau_A)$$

Entsprechend ergeben sich als Nutzungskosten einer Ressource  $d_r$  durch eine Aktivität  $\tau_A$ :

$$k_N(d_r, \tau_A) = \sum_{b \in B} k(b_{Res}) \text{ mit } \alpha(b) = s_{Res}(\tau_A) \text{ und } \lambda(b) = d_r$$

### **Beispiel**

Exemplarisch sei die Aktivität `do_DB` für den Ablauf in Abbildung 7.8 und das zugehörige Geschäftsprozeßmodell aus Abbildung 5.15 betrachtet. Als Nutzungskosten der Ressourcen-Klasse MA durch `do_DB` ergeben sich:

$$\begin{aligned}
k_N(MA, do\_DB) &= k(b_{Res}(do\_DB(1))) \\
&= 20
\end{aligned}$$

Für die Nutzungskosten der Mitarbeiter vom Typ 1 durch `do_DB` ergeben sich:

$$\begin{aligned}
k_N(1, do\_DB) &= k(b_{Res}(do\_DB(1))) \\
&= 20
\end{aligned}$$

Die Angabe der Nutzungskosten für andere Ressourcen erübrigt sich, da durch Ausführung der Aktivität für den gegebenen Ablauf ausschließlich Mitarbeiter vom Typ 1 genutzt wurden.

□

### **7.3.4 Ermittlung der Gesamtkosten**

Mit der Ermittlung der kostenoptimalen Verteilung echter Wartezeit in Abschnitt 7.3.2 sind alle Kostengrößen für einen Ablauf bekannt. Damit kann nun die Summe aller Kosten

für einen Ablauf eines Geschäftsprozeßmodells ermittelt werden. Da die Kostenbewertungen der für den Verbrauch von Ressourcen eingeführten internen Bedingungen  $b_{Res}$  lediglich den Anteil der ressourcenbezogenen Kosten an den Aktivitätskosten angeben, ergeben sich die *Gesamtkosten*  $k_{Ges}(A)$  eines halbgeordneten Ablaufs  $A$  eines *VIPbusiness*-Netztes  $N$  als Summe der Kostenbewertungen aller Bedingungen des Ablaufs, die nicht auf eine interne Stelle  $s_{Res}(\tau_A)$  einer Aktivitäts-Transitionen  $\tau_A$  von  $N$  abgebildet werden:

$$k_{Ges}(A) = \sum_{b \in B} k(b) \text{ mit: es existiert kein } \tau_A \in T_A \text{ mit } \alpha(b) = s_{Res}(\tau_A)$$

Falls die beiden Größen schon bekannt sind, können die Gesamtkosten eines Ablaufs auch als Summe der Aktivitätskosten  $k_A(A)$  und der echten Wartekosten  $k_{W_e}(A)$  berechnet werden:

$$k_{Ges}(A) = k_A(A) + k_{W_e}(A)$$

### Beispiel

Für den Beispiel-Ablauf in Abbildung 7.8 ergeben sich als Gesamtkosten:

$$k_{Ges}(A) = 685$$

□

## 7.3.5 Abgeleitete Kostengrößen

Wie schon in Abschnitt 7.2.5 können die in den vorangegangenen Abschnitten ermittelten Kenngrößen für Kosten als Ausgangspunkt zur Ermittlung weiterer Kostengrößen genutzt werden. Auch hier existieren prinzipiell beliebig viele Kenngrößen, die untersucht werden können. In diesem Abschnitt werden zunächst wieder eine Reihe von *Verhältnisgrößen* eingeführt. Anschließend wird gezeigt, wie mittels einer Tabelle mit einzelnen Kostengrößen eine systematische Aufbereitung der Kostenstruktur eines Geschäftsprozeß-Entwurfs durchgeführt werden kann.

### Verhältnisgrößen

Im Bereich Kosten werden die folgenden Verhältnisgrößen betrachtet:

- (Echte) Wartekosten vs. Aktivitätskosten:  $\frac{k_{W_e}(A)}{k_A(A)}$

Echte Wartekosten können beispielsweise durch Kosten für Lagerung von Material oder Produkten entstehen. Es kann sich dabei aber auch um Kosten für nicht genutzte (und damit möglicherweise veraltete) Informationen oder Opportunitätskosten für nicht zufriedene Kunden handeln. Ebenso wie echte Wartezeit sind echte Wartekosten eine zu minimierende Größe. Daher sollte das Verhältnis von echten Wartekosten zu Aktivitätszeit im Idealfall gegen 0 gehen.

- Verhältnisse der Bestandteile der Aktivitätskosten:

Als Beispiel sei hier das Verhältnis der Vorbereitungskosten zu den Bearbeitungskosten genannt:  $\frac{k_V(A)}{k_B(A)}$ .

Diese Kenngröße stellt einen wertvollen Hinweis in Bezug auf eine sinnvolle Zuordnung der Ressourcen zu Cases dar. Fallen die Kosten für vorbereitende Tätigkeiten im Vergleich zu den Kosten für die eigentliche Ausführung der Aktivität hoch aus, kann es sich lohnen, dieselbe Ressource mehrere Aktivitäten für einen Case ausführen zu lassen, um den Aufwand für vorbereitende Arbeiten zu reduzieren (z.B. Beschaffungszeit für Akten, Einlesezeit etc.).

- Ressourcenkosten vs. Aktivitätskosten:  $\frac{k_{Res}(A)}{k_A(A)}$

Ressourcenkosten sind Kosten, die durch den direkten Verbrauch an Ressourcen durch eine (Teil-)Aktivität entstehen. Die Differenz zwischen Kosten einer Aktivität und den zugehörigen Ressourcenkosten stellt in der Regel einen Anteil der Gemeinkosten dar, der auf die jeweilige Aktivität umgeschlagen wird (vgl. Abschnitt 2.2.3). Geht das Verhältnis der Ressourcenkosten zu Aktivitätskosten gegen 1, deutet dies auf allgemein geringe Gemeinkosten oder einen geringen Anteil an zurechenbaren Gemeinkosten für die ausgeführten Aktivitäten hin.

### *Beispiel*

Für den Ablauf in Abbildung 7.4 ergibt sich als Verhältnis von echten Wartekosten zu Aktivitätskosten:

$$\frac{k_{W_e}(A)}{t_A(A)} = \frac{450}{235} = 1.91$$

Dieser Wert ist weit vom idealen Ergebnis 0 entfernt, die echten Wartekosten sind fast doppelt so hoch wie die Aktivitätskosten. Die Ursache liegt neben der Menge der auftretenden echten Wartezeit in den hohen variablen Kostenbewertungen, die für echte Wartezeit angesetzt wurden.

Für Verhältnis der Vorbereitungskosten zu Bearbeitungskosten ergibt sich:

$$\frac{k_V(A)}{t_B(A)} = \frac{50}{185} = 0.27$$

Die Vorbereitungskosten entsprechen also 27% der Bearbeitungskosten. Es dürfte sich daher lohnen, eine Reduzierung der Vorbereitungszeit und den damit verbundenen Kosten anzustreben, beispielsweise über konsequente Zuordnung mehrerer für einen Case auszuführende Aktivitäten zu derselben Ressource.

Für Verhältnis der Ressourcenkosten zu Aktivitätskosten ergibt sich:

$$\frac{k_{Res}(A)}{t_A(A)} = \frac{125}{235} = 0.53$$

Dieses Ergebnis deutet auf einen hohen Gemeinkosten-Anteil in den Aktivitätskosten hin, ist jedoch insofern kein repräsentatives Ergebnis, als daß aus Gründen der graphischen

Übersichtlichkeit für das Berechnen der Schadenshöhe sowie die Entscheidung über eine detaillierte Überprüfung der Bearbeitung keine Ressourcen berücksichtigt wurden. □

### Detaillierte Analyse einzelner Kostenwerte

Um einen detaillierten Einblick in die Kostenstruktur eines Ablaufs zu erhalten, müssen die einzelnen Aktivitäten des Geschäftsprozesse betrachtet werden. Dies geschieht mit Hilfe einer für jeden Ablauf  $A$  aufgestellten Tabelle, in der für jede Aktivität  $\tau_A$  die folgenden (zeitabhängigen) Kostengrößen eingetragen werden:

- Transportkosten  $k_T(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Rüstkosten  $k_R(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- (*alternativ*) Vorbereitungskosten  $k_V(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Bearbeitungskosten  $k_B(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- (*optional*) fertigungsbedingte Wartekosten  $k_{W_f}(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Aktivitätskosten  $k_A(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Ressourcenkosten  $k_{Res}(t(\tau_A))$  multipliziert mit der Anzahl der Ausführungen von  $\tau_A$  für  $A$ ,
- Anteil der Ressourcenkosten  $k_{Res}(t(\tau_A))$  an den Aktivitätskosten  $k_A(t(\tau_A))$  der Aktivität,
- Anteil der  $\tau_A$  zurechenbaren Aktivitätskosten an den Aktivitätskosten  $k_A(A)$  des Ablaufs,
- Anteil der  $\tau_A$  zurechenbaren Ressourcenkosten an den Ressourcenkosten  $k_{Res}(A)$  des Ablaufs,
- Anteil der  $\tau_A$  zurechenbaren Aktivitätskosten an den Gesamtkosten  $k_{Ges}(A)$  des Ablaufs.

Eine wichtige Anwendung dieser Tabelle ist die Ermittlung von kostentreibenden Aktivitäten (vgl. Abschnitt 2.2.3). Dabei handelt es sich um die Aktivitäten, die den größten Anteil an den Gesamtkosten des Ablaufs haben. Die Ermittlung dieser Aktivitäten ist ein wichtiger Ansatzpunkt für eine Verbesserung des Geschäftsprozeß-Entwurfs unter kostenorientierter Gesichtspunkten.

### Beispiel

In Tabelle 7.1 sind für alle Aktivitäten des zum Ablauf in Abbildung 7.4 gehörenden Geschäftsprozeßmodells die einzelnen Kostengrößen eingetragen worden.

Es zeigt sich, daß die Rückfrage beim Kunden nicht nur die meiste Zeit in Anspruch nimmt (vgl. Abschnitt 7.2.5), sondern auch die größten Kosten verursacht. Für den Fall, daß die Leistungsbewertung des Geschäftsprozeß-Entwurfs nicht den Erwartungen entspricht, bietet diese Aktivität das größte Potential zur Reduzierung von Aktivitätskosten. Ein Vergleich der Aktivitätskosten der unterschiedlichen Aktivitäten zeigt, daß die Aktivitätszeit sowie ihre Bestandteile ansonsten einigermaßen gleichmäßig auf alle Aktivitäten verteilt sind. Für diesen Ablauf existiert also keine einzelne kostentreibende Aktivität.

	Eingang bestätigen	Akte anlegen	Datenbank einsehen	Schadenshöhe ermitteln	Rückfrage	Überprüfung?
$k_V(\tau_A)$	10	5	10	10	10	5
$k_B(\tau_A)$	20	40	20	30	60	15
$k_A(\tau_A)$	30	45	30	40	70	20
$k_{Res}(\tau_A)$	25	30	20	0	50	0
$\frac{k_{Res}(\tau_A)}{k_A(\tau_A)}$	0.83	0.66	0.66	0	0.71	0
$\%k_A(A)$	12.8	19.1	12.8	17.0	29.8	8.5
$\%k_{Res}(A)$	20.0	24.0	16.0	0	40.0	0
$\%k_{Ges}(A)$	4.4	6.6	4.4	5.8	10.2	3.0

Tabelle 7.2: Detaillierte Analyse der Kostengrößen für Aktivitäten.

□

Mit der Ermittlung der kostenoptimalen Verteilung von Wartezeit in Abschnitt 7.3.2 ist eine sinnvolle Zuordnung der echten Wartezeit zu den Bedingungen des Ablaufs möglich. Ausgehend von dieser Zuordnung kann nun durch die Erstellung einer Tabelle (vgl. dazu die Ausführungen in Abschnitt 7.2.5) eine detaillierte Untersuchung der echten Wartezeit vorgenommen werden. Da mit der Zuordnung der echten Wartezeit auch die echten Wartekosten ermittelt werden können, wird die Tabelle an dieser Stelle um die folgenden Daten für jede externe Stelle  $s$  ergänzt:

- die Summe der echten Wartekosten, die auf einer externen Stelle angefallen sind, als Summe der echten Wartekosten  $k_{W_e}(b)$  für alle Bedingungen mit  $\alpha(b) = s$ ,
- Anteil der  $s$  zurechenbaren Wartekosten an den echten Wartekosten  $k_{W_e}(A)$  des Ablaufs,
- Anteil der  $\tau_A$  zurechenbaren Wartekosten an den Gesamtkosten  $t_{D_{min}}(A)$  des Ablaufs.

Auch hier gibt die Analyse der einzelnen, insbesondere der minimalen und maximalen, Werte aufschlußreiche Hinweise für die Gestaltung des zugrundeliegenden Geschäftsprozeß-Entwurfs, vor allem im Hinblick auf eine Reduzierung der minimalen Durchlaufzeit und der damit verbundenen Reduzierung der echten Wartekosten.



## Beispiel

In Tabelle 7.3 sind für alle externen Stellen<sup>15</sup> des zum Ablauf in Abbildung 7.7 gehörenden Geschäftsprozeßmodells die einzelnen Werte für echte Wartezeit und den damit verbundenen echten Wartekosten eingetragen worden.

Es zeigt sich, daß die Wartezeit und die entsprechenden Wartekosten auf drei externe Stellen verteilt sind. Während die Wartezeit zum größten Teil auf der Stelle **Datenbank nötig** anfällt (42.8%), verursacht das Warten auf der Stelle **Bestätigung fertig** den höchsten Kostenanteil (44.4%). Für beide Stellen kann jetzt geprüft werden, ob sich durch Umgestaltung des Geschäftsprozeß-Entwurfs die auftretende Wartezeit und die damit verbundenen Kosten reduzieren lassen. Für die Stelle **Bestätigung fertig** sollte zudem geprüft werden, worin der Grund für die offensichtlich sehr hohen relativen Wartekosten liegt. □

## 7.4 Weitere Methoden zur Ressourcenanalyse

Mit der Ermittlung ressourcenbezogener Zeit- und Kostengrößen in den vorangegangenen Abschnitten sind bereits einige wichtige Analysemöglichkeiten für Ressourcen vorgestellt worden. In diesem Abschnitt werden zwei weitere Verfahren eingeführt, die eine zusätzliche Analyse des Einsatzes von Ressourcen zur Ausführung von Aktivitäten in Abläufen erlauben. Dabei werden in Abschnitt 7.4.1 die Auswirkungen der gemeinsamen Nutzung beschränkter Ressourcen durch Aktivitäten in Form von *Linearisierungs-Effekten* untersucht. In Abschnitt 7.4.2 wird ein Verfahren zur Planung des Einsatzes von Ressourcen durch Konfiguration spezieller Kontrollparameter ermöglicht.

### 7.4.1 Linearisierungen durch Ressourcen

In der Regel sind die Ressourcen, die zur Ausführung der Aktivitäten benötigt werden, nicht in unendlicher Menge vorhanden. In diesem Abschnitt wird ein Verfahren vorgestellt, mit dessen Hilfe die Effekte des Einsatzes solcher *beschränkter* Ressourcen untersucht werden können. Dabei werden die mit dem *VIPbusiness*-Ansatz verbundene explizite Modellierung von Ressourcen sowie die explizite graphische Darstellung des Verhaltens von Geschäftsprozeß-Entwürfen mittels halbgeordneter Abläufe und Kausalnetzen genutzt.

Ausgangspunkt der Analyse ist ein halbgeordneter Ablauf (beziehungsweise das Kausalnetz zu seiner graphischen Repräsentation), der durch Simulation des zugehörigen Geschäftsprozeßmodells unter Nicht-Berücksichtigung von Ressourcen generiert wurde<sup>16</sup>. Mittels Ergänzung des Kausalnetzes um Ressourcen-Stellen sowie eine Anfangsmarkierung erhält man ein markiertes High-Level Petrinetz, das wieder simuliert werden kann. Ergebnis dieser Simulation ist das Verhalten des Geschäftsprozeß-Entwurfs unter Berücksichtigung des Einsatzes von Ressourcen in Form neuer Abläufe, basierend auf dem bereits bekannten Ablauf ohne Ressourcen. Durch dieses Verhalten werden alle Möglichkeiten für den Einsatz der beschränkten Ressourcen repräsentiert.

---

<sup>15</sup>Ressourcen-Stellen wurden aus den bekannten Gründen nicht berücksichtigt (vgl. Abschnitt 7.2.2).

<sup>16</sup>Durch die Möglichkeit, Ressourcen „ein- und ausblenden“ zu können, ist die Durchführung der Simulation ohne Berücksichtigung der Ressourcen problemlos möglich (vgl. Abschnitt 5.3).

Case ange- kommen	Bestäti- gung nötig	Bestäti- gung fertig	Akte nötig	Akte fertig	Datenbank nötig	Datenbank fertig	Schadens- höhe nötig	Schadens- höhe fertig	Rück- frage nötig	Rück- frage fertig	Case fertig
$t_{W_e}(s)$	0	0	0	0	30	0	20	0	0	0	0
$\%t_{W_e}(A)$	0	0	0	0	42.8	0	28.6	0	0	0	0
$\%t_{D_{min}}(s)$	0	0	0	0	33.3	0	22.2	0	0	0	0
$k_{W_e}(s)$	0	0	0	0	150	0	100	0	0	0	0
$\%k_{W_e}(A)$	0	0	0	0	33.3	0	22.2	0	0	0	0
$\%k_{Ges}(A)$	0	0	0	0	21.9	0	14.6	0	0	0	0

Tabelle 7.3: Detaillierte Analyse echter Wartezeit und echter Wartekosten.

Beim Vergleich der neu generierten Abläufe (mit Ressourcen) mit dem ursprünglichen Ablauf (ohne Ressourcen) sind vor allem die Effekte von Linearisierungen von Interesse. *Linearisierungen* entstehen, wenn nebenläufig ausführbare Aktivitäten durch den konkurrierenden Zugriff auf eine beschränkte Ressource in eine sequentielle Ordnung gezwungen werden. Die Folge von Linearisierungen sind in der Regel höhere Durchlaufzeiten der betroffenen Abläufe sowie mehr echte Wartezeit und entsprechend höhere Wartekosten.

Die Analyse der Effekte des Einsatzes von Ressourcen wird für zwei Szenarien betrachtet:

1. 1-Case-Szenario

Es werden die Abläufe zur Bearbeitung einzelner Cases untersucht. Dabei ist die Beeinflussung der Nebenläufigkeit einzelner Aktivitäten durch den Einsatz beschränkter Ressourcen von Interesse.

2. Mehr-Case-Szenario

Es werden die Abläufe zur Bearbeitung mehrerer Cases untersucht. Dabei konkurrieren zusätzlich zu nebenläufigen Aktivitäten auch die einzelnen Cases um die Ressourcen, d.h., es muß nicht nur entschieden werden, welche Aktivität die Ressource erhält, sondern auch, welcher Case bearbeitet werden soll.

Für beide Szenarien wird das Vorgehen im Folgenden anhand eines Beispiels erläutert.

**1-Case-Szenario**

Bei der Analyse von 1-Case-Szenarien wird die Auswirkung des Einsatzes beschränkter Ressourcen auf die Bearbeitung einzelner Cases untersucht. Ausgangspunkt der Analyse ist ein Ablauf, der für ein Geschäftsprozeßmodell unter Nicht-Berücksichtigung von Ressourcen generiert worden ist.

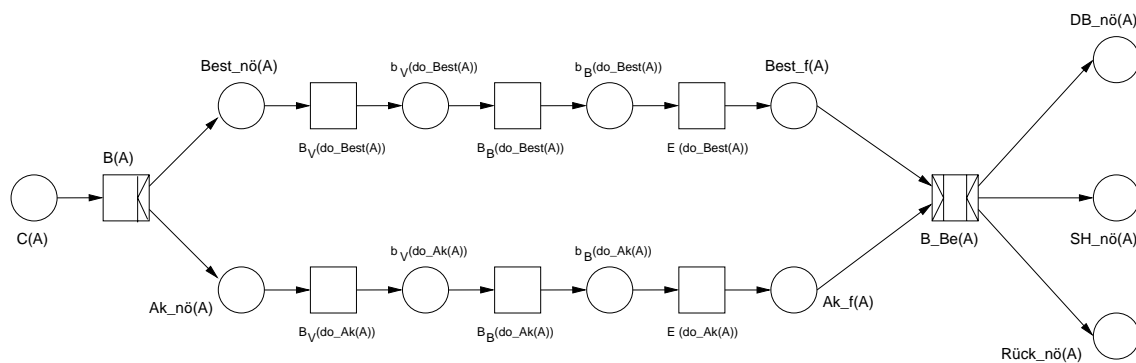


Abbildung 7.9: Ausgangsablauf für die Linearisierungen von Ressourcen.

**Beispiel**

Abbildung 7.9 zeigt einen Teil des Ablaufs aus Abbildung 7.8 zur Vorbereitung der Bearbeitung einer Schadensmeldung. Wie schon in den vorangegangenen Beispielen erläutert,

können die Bestätigung des Eingangs der Schadensmeldung und das Anlegen der Akte nebenläufig erfolgen. Um das Beispiel möglichst übersichtlich zu halten, wurde bei den Marken nur noch der Identifikator für den zu bearbeitenden Case verwendet, der Wert des Schadens wurde weggelassen. Da über eine Kontrolle der Bearbeitung nicht zu entscheiden ist (der entsprechende Teil des Geschäftsprozeßmodells wird nicht betrachtet), wird diese Information für das Routing der Cases nicht benötigt.

□

Dieser Ablauf wird um Ressourcen-Stellen nach den Vorgaben des zugehörigen Geschäftsprozeßmodells ergänzt, d.h., die Ressourcen-Stellen werden hinzugefügt und über entsprechende Kanten mit den Aktivitäts-Ereignissen beziehungsweise deren Teil-Ereignissen verbunden. Die Ressourcen-Stellen erhalten als Anfangsmarkierung die anfangs verfügbaren Ressourcen der modellierten Ressourcen-Klasse. Zusätzlich werden die Aktivitäts-Ereignisse um die interne Stelle zur Modellierung der Ressourcen-Nutzung ergänzt.

Darüber hinaus werden die Bedingungen des minimalen Schnitts des Ablaufs mit einer Anfangsmarkierung versehen. Dabei reicht die Verwendung nicht unterscheidbarer Marken aus.

Als Ergebnis erhält man ein markiertes Petrinetz<sup>17</sup>, das als Input für einen Simulator geeignet ist. Auf eine Umbenennung der Netzelemente wurde verzichtet, auch wenn diese jetzt nicht mehr als Bedingungen und Ereignisse, sondern als Stellen und Transitionen interpretiert werden.

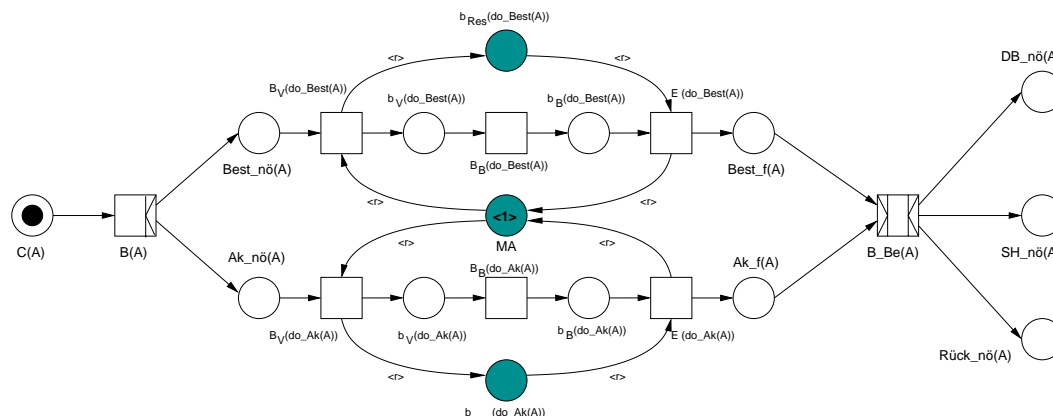


Abbildung 7.10: Ablauf mit Ressourcen und Anfangsmarkierung.

### Beispiel

In Abbildung 7.10 ist der Ablauf aus Abbildung 7.9 um Ressourcen-Stellen. Die Stellen C(A) und MA sind mit den Marken • beziehungsweise <1> versehen worden.

□

<sup>17</sup>Für den Fall, daß zur Modellierung der anfangs verfügbaren Ressourcen unterscheidbare Marken verwendet wurden, handelt es sich dabei wieder um ein höheres Petrinetz.

Der Ablauf mit Anfangsmarkierung und zusätzlichen Stellen wird als markiertes Petrinetz interpretiert und mittels der bekannten *VIPsim*-Konzepte simuliert (vgl. Abschnitt 6.2). Das Ergebnis ist eine Menge von Abläufen, die alle Möglichkeiten für die Allokation der beschränkten Ressourcen repräsentieren. Für den Fall, daß die Beschränkung der Ressourcen zu einer Beschränkung der Nebenläufigkeit bei der Ausführung von Aktivitäten führt, lassen sich Linearisierungs-Effekte beobachten, die je nach Allokation der Ressourcen unterschiedlich ausfallen können.

Auf diese Weise läßt sich untersuchen, ob der Einsatz beschränkter Ressourcen Auswirkungen auf die Abläufe des Geschäftsprozeßmodells hat. Für den Fall auftretender Linearisierungen muß entschieden werden, ob diese hingenommen werden können oder gegebenenfalls durch eine neue Zuordnung von Ressourcen zu Aktivitäten im Geschäftsprozeßmodell oder eine Aufstockung der anfangs verfügbaren Ressourcen der einzelnen Ressourcen-Klassen zu vermeiden sind.

Die neu generierten Abläufe können zudem mit allen in den vorangegangenen Abschnitten vorgestellten Analyseverfahren untersucht werden. Lassen sich Linearisierungen nicht vermeiden (oder wird ihr Auftreten als unkritisch erachtet), kann durch Analyse der unterschiedlichen Abläufe untersucht werden, für welchen Ablauf die Allokation von Ressourcen bezüglich der relevanten Merkmale zur Leistungsbewertung, beispielsweise der Durchlaufzeit, am besten abschneidet. Ausgehend von dieser Information können für den praktischen Einsatz des Geschäftsprozeß-Entwurfs Steuerungs-Mechanismen für die Allokation der Ressourcen generiert werden.

### **Beispiel**

Abbildung 7.11 zeigt einen der neuen Abläufe für den um Ressourcen ergänzten Ablauf aus Abbildung 7.10. Da nur ein Mitarbeiter zur Verfügung steht, müssen die Bestätigung des Eingangs und das Anlegen der Akte sequentiell erfolgen, es kommt also zu einer Linearisierung mit entsprechender Verlängerung der Durchlaufzeit.

Insgesamt gibt es zwei neue Abläufe, resultierend aus den beiden Möglichkeiten die zwei vormals nebenläufigen Aktivitäten nacheinander auszuführen.

Um zu verdeutlichen, daß es sich bei dem Kausalnetz um einen Ablauf des um Ressourcen ergänzten (und anschließend als markiertes Netz interpretierten) Ablauf aus Abbildung 7.10 handelt, wurden die Bedingungen und Ereignisse mit einem „<sup>l</sup>“ in der Bezeichnung versehen (beispielsweise  $C^l(A)$  oder  $b_V^l(do\_Best(A))$ ).

□

### **Mehr-Case-Szenario**

Werden bei der Bearbeitung mehrerer Cases keine Ressourcen berücksichtigt, erfolgt die Bearbeitung der Cases nebenläufig, da die zur Modellierung der Aktivitäten benutzten Aktivitäts-Transitionen in verschiedenen Modi nebenläufig ausgeführt werden können. Das Ergebnis sind Abläufe, die wie eine Ansammlung der Abläufe zur Bearbeitung einzelner Cases aussehen (vgl. dazu auch Abbildung 7.12). Werden bei der Bearbeitung mehrerer

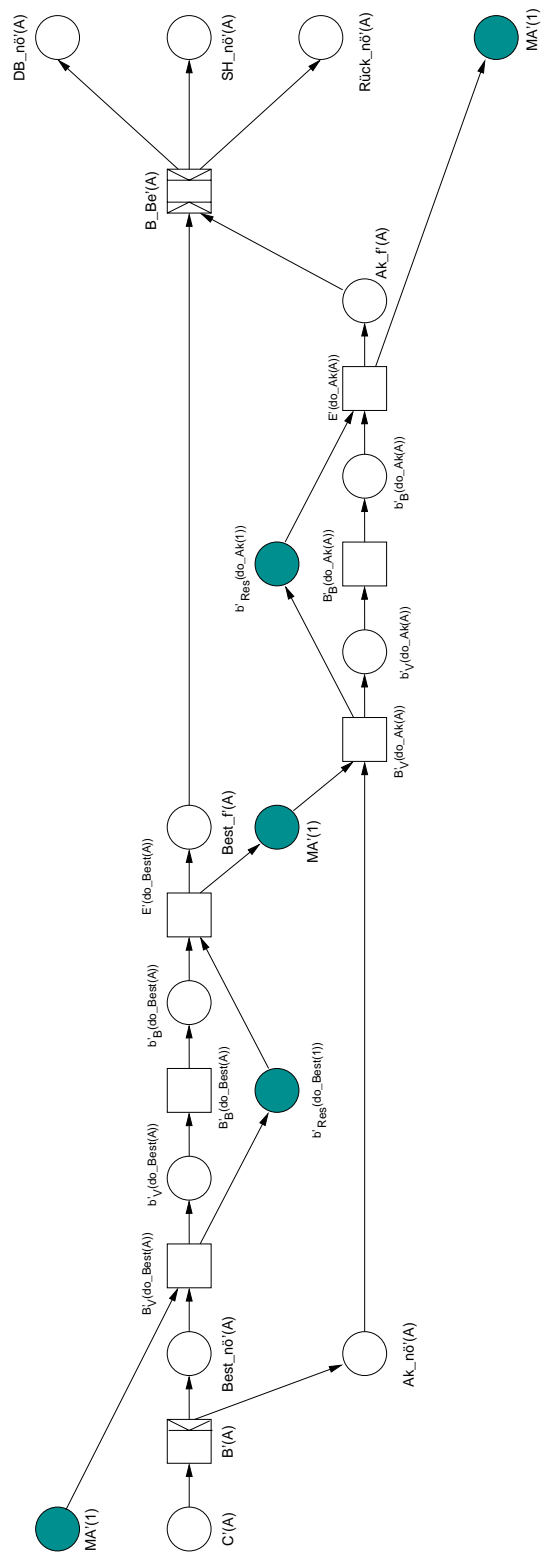


Abbildung 7.11: Linearisierte Version des Ablaufs aus Abbildung 7.10.

Cases beschränkte Ressourcen berücksichtigt, so wird die vom 1-Case-Szenario bekannte Konkurrenz nebenläufig ausführbarer Aktivitäten durch die Konkurrenz der verschiedenen zu bearbeitenden Cases verstärkt. Durch die gemeinsame Abhängigkeit der Cases von beschränkten Ressourcen findet also eine Interaktion statt, die Cases beeinflussen sich gegenseitig bei ihrer Bearbeitung. Die explizite Darstellung eventuell auftretender Ressourcen-Engpässe in Form von Linearisierungen erleichtert die Analyse der Ursachen solcher Engpässe erheblich. In diesem Punkt unterscheidet sich der *VIPbusiness*-Ansatz von „klassischen“ Ansätzen zur Leistungsbewertung wie beispielsweise Warteschlangen-Modellen [Kle75], bei denen nur die Effekte von Ressourcen-Engpässen über Kenngrößen wie Durchlaufzeit oder Auslastungsgrad, nicht jedoch die Ursachen dieser Engpässe ermittelt werden können.

Ausgangspunkt ist wieder ein Ablauf zur Bearbeitung der entsprechenden Cases gemäß Geschäftsprozeßmodell unter Nicht-Berücksichtigung von Ressourcen.

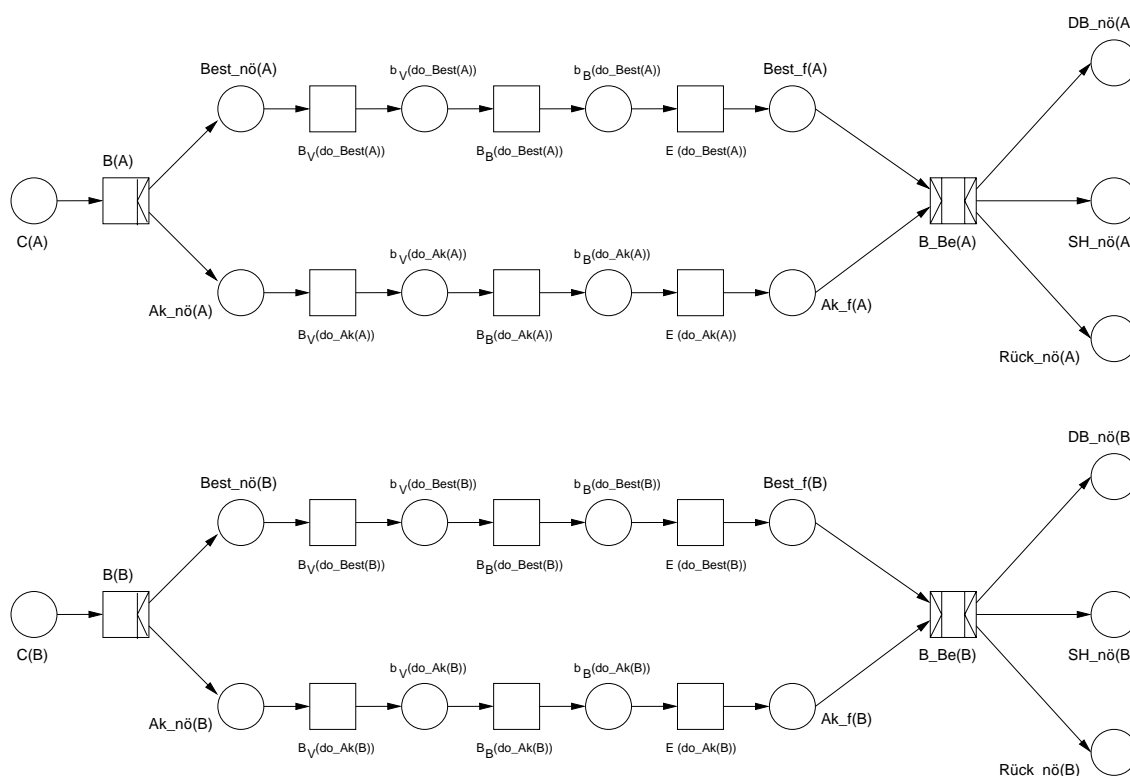


Abbildung 7.12: Bearbeitung mehrerer Cases ohne Ressourcen.

### Beispiel

Abbildung 7.12 zeigt einen Ablauf zur Bearbeitung zweier Cases A und B. Dabei handelt es sich im Prinzip um die gemeinsame Darstellung der Abläufe zur Bearbeitung der einzelnen Cases.

□

Dieser Ablauf wird um Ressourcen-Stellen ergänzt, die einzelnen Aktivitäts-Ereignisse

erhalten zusätzliche Bedingungen für die Nutzung von Ressourcen. Die Bedingungen des minimalen Schnitts werden mit nicht unterscheidbaren Marken versehen, die Ressourcen-Stelle erhält als Markierung die anfangs verfügbaren Ressourcen. Der so ergänzte Ablauf kann wieder als markiertes Petrinetz interpretiert werden und ist somit ausführbar.

### Beispiel

In Abbildung 7.13 ist der Ablauf aus Abbildung 7.12 um Ressourcen-Stellen, interne Bedingungen der Aktivitäts-Ereignisse sowie eine Anfangsmarkierung ergänzt worden. □

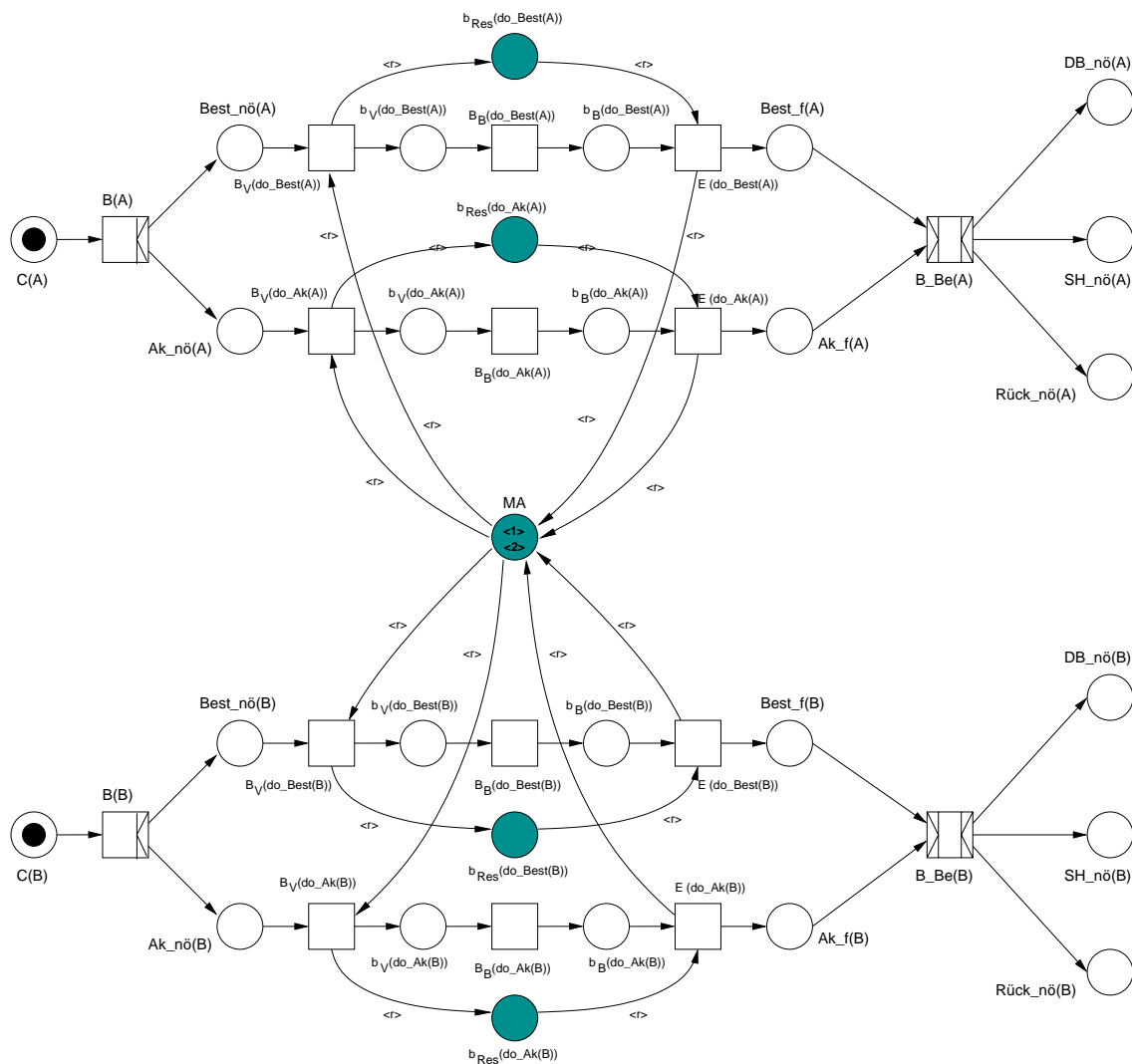


Abbildung 7.13: Ablauf aus Abbildung 7.12 mit Ressourcen und Anfangsmarkierung.

Durch Simulation des ergänzten Ablaufs wird eine Menge neuer Abläufe generiert. Diese Menge repräsentiert alle Möglichkeiten, die beschränkten Ressourcen zur Bearbeitung der Cases einzusetzen. Für den Fall, daß die Beschränkung der Ressourcen zu Engpässen



führt, lassen sich wieder Linearisierungs-Effekte beobachten. Durch diese Effekte lassen sich gegebenenfalls die für die neuen Abläufe mittels Anwendung der in den vorangegangenen Abschnitten eingeführten Analyseverfahren erhaltenen Kenngrößen erklären. Auf diese Weise können beispielsweise die Ursachen für erhöhte Durchlaufzeiten auf einfache Weise ermittelt werden.

### **Beispiel**

In den Abbildungen 7.14 und 7.15 sind zwei der neuen Abläufe des ergänzten Ablaufs aus Abbildung 7.13 dargestellt.

Beim Ablauf in Abbildung 7.14 wird je einer der beiden Cases komplett von einem der beiden zur Verfügung stehenden Mitarbeiter bearbeitet, es findet damit keine Interaktion der Cases statt. Die Linearisierungs-Effekte beschränken sich auf die sequentielle Ausführung der ursprünglich nebenläufigen ausführbaren Eingangsbestätigung und dem Anlegen der Akte.

Beim Ablauf in Abbildung 7.15 wird von jedem Mitarbeiter je eine Aktivität für einen der beiden Cases ausgeführt. Auf diese Weise kommt es zu einer Interaktion der Cases über die gemeinsam genutzten Ressourcen.

Um zu verdeutlichen, daß es sich bei dem Kausalnetz um einen Ablauf des um Ressourcen ergänzten Ablaufs aus Abbildung 7.10 handelt, werden die Bedingungen und Ereignisse wieder mit einem „'“ in der Bezeichnung versehen. □

Es sei an dieser Stelle angemerkt, daß die linearisierten Abläufe vergleichsweise lang und unübersichtlich werden können. Bei einem hohen Anteil nebenläufig ausführbarer Aktivitäten, die ausschließlich über beschränkte Ressourcen miteinander verknüpft sind, lassen sich auf diese Weise die Effekte von Ressourcen-Engpässen sehr schön beobachten.

## **7.4.2 Konfiguration von Kontrollparametern**

Wie in Abschnitt 5.2 bereits erläutert, sind die verschiedenen möglichen Abläufe eines Geschäftsprozeß-Modells das Ergebnis von Routing-Entscheidungen in Form von *OR*-Verzweigungen.

Dabei können zwei Arten von Auswahlentscheidungen unterschieden werden:

- Auswahlentscheidungen, die nicht beeinflusst werden können,
- Auswahlentscheidungen, die durch *Kontrollparameter* beeinflusst werden können.

Auswahlentscheidungen, die nicht beeinflussbar sind, können von externen Faktoren wie dem Verhalten von Kunden oder internen Faktoren wie Fehler von Mitarbeitern abhängen. Die andere Art von Auswahlentscheidungen hängt dagegen (teilweise) von *Kontrollparametern* ab, deren Werte im Prinzip frei gewählt werden können. So wird beispielsweise in unserem Versicherungs-Geschäftsprozeß-Modell für Schadensfälle mit einem Schadenswert über 10000 eine zusätzliche Überprüfung der Bearbeitung des Schadensfalls vorgenommen. Der Wert 10000 ist dabei beliebig gewählt. Dennoch hängt die tatsächliche

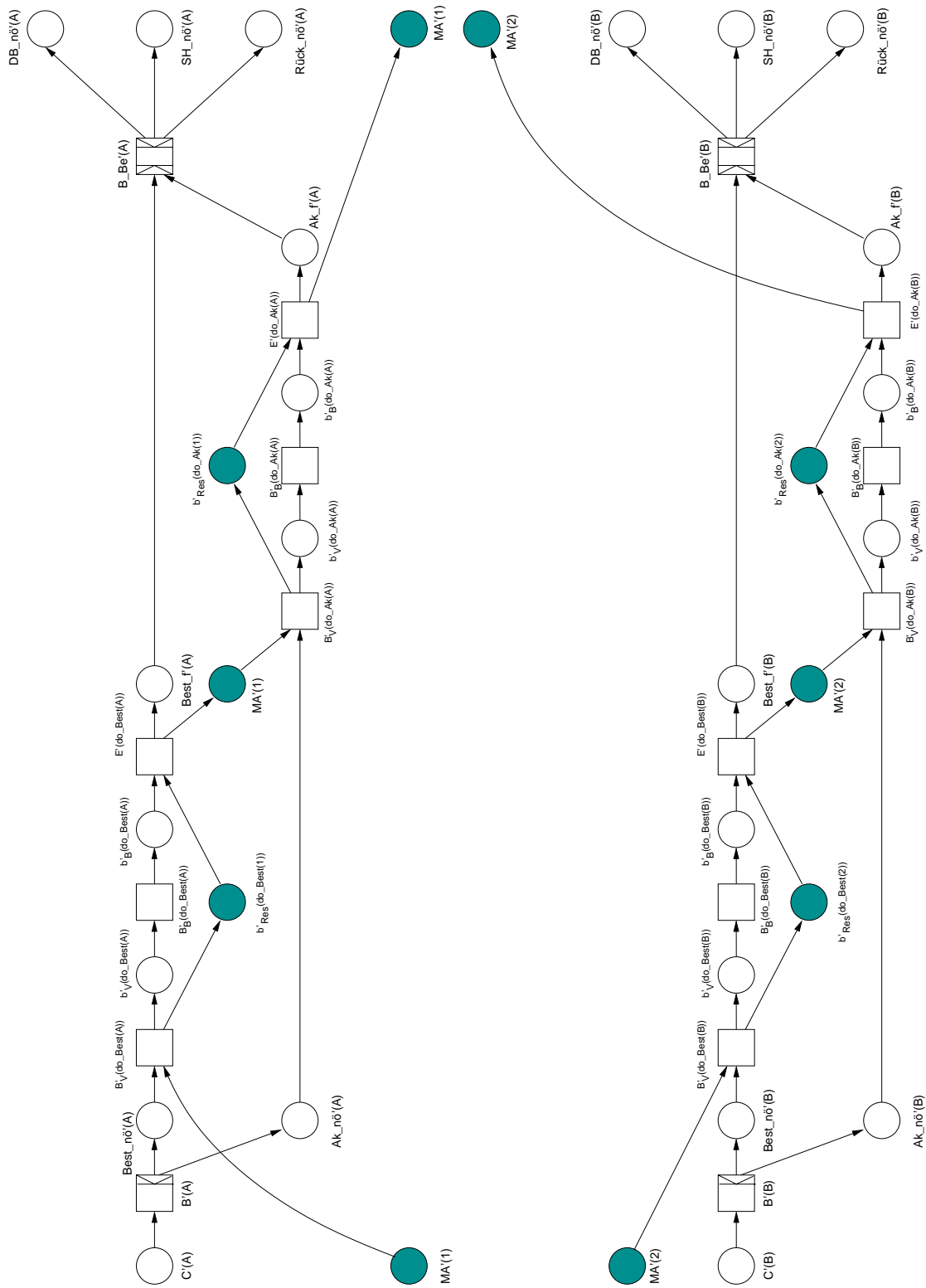


Abbildung 7.14: Linearisierte Version des Ablaufs aus Abbildung 7.13 (I).

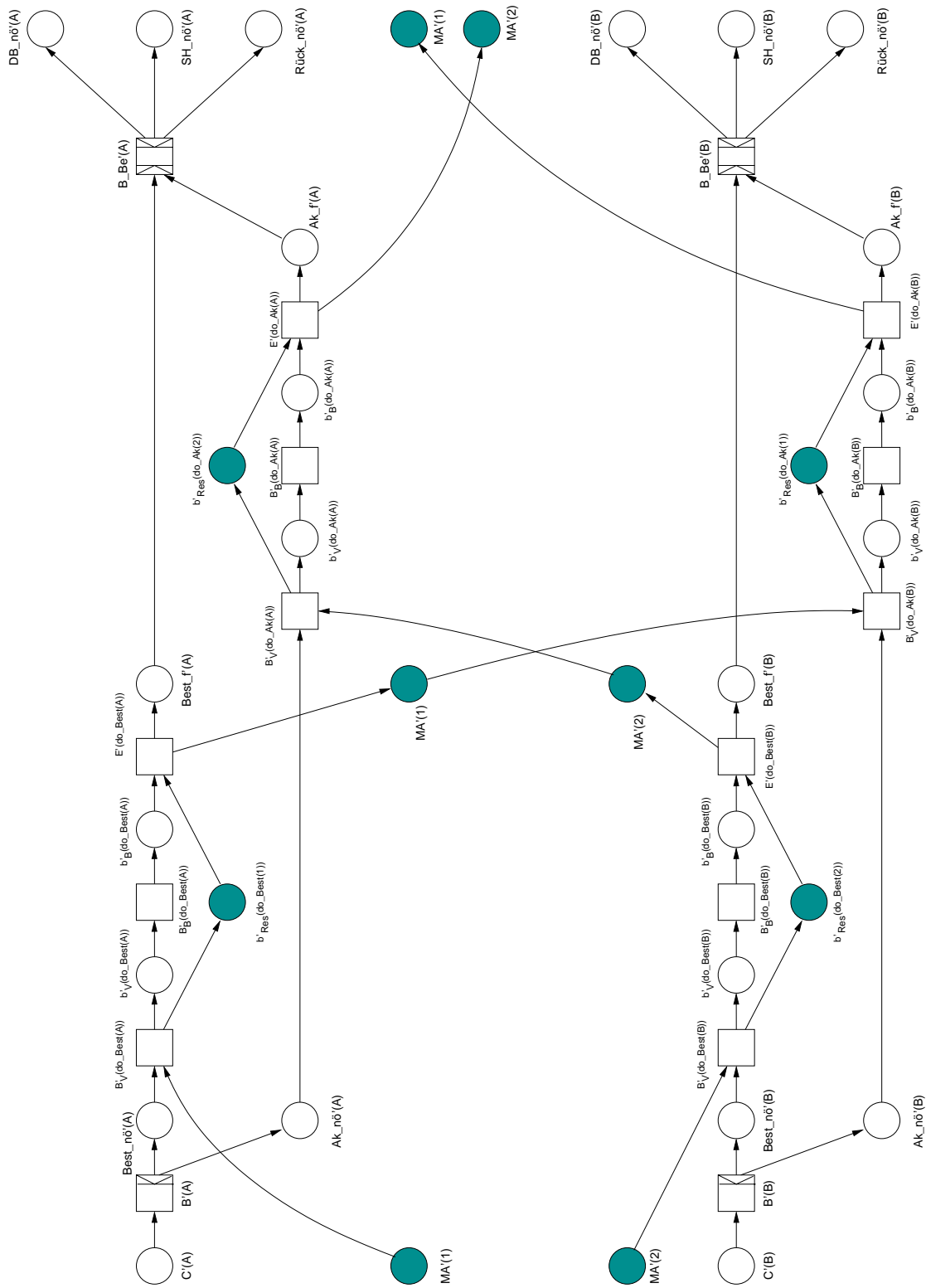


Abbildung 7.15: Linearisierte Version des Ablaufs aus Abbildung 7.13 (II).

Auswahlentscheidung letzten Endes von der wirklichen Schadenshöhe bei der Bearbeitung entsprechender Cases ab. Anstelle des Kriteriums *Höhe des Schadenswertes* kann die Routing-Entscheidung jedoch auch durch Festlegung *prozentualer Wahrscheinlichkeiten* für das Treffen der jeweiligen Entscheidung erfolgen. Durch dieses Vorgehen wird eine Unabhängigkeit der Auswahlentscheidung von anderen Faktoren erreicht. Typische Anwendungs-Szenarien für dieses Vorgehen finden sich vor allem im Rahmen des *Total Quality Management (TQM)*<sup>18</sup>, wo mit Hilfe von Stichproben die Qualität erzeugter Produkte überprüft wird. Dabei entsteht in der Regel ein Entscheidungsproblem zwischen Aufwand (höhere Auslastung von Ressourcen, höhere Durchlaufzeiten,...) und Nutzen (höhere Sicherheit bezüglich der Qualität) der zusätzlichen Kontroll-Aktivitäten. Etwas allgemeiner läßt sich festhalten, daß mittels Kontrollparametern beeinflusst werden kann, wie ein Case im Rahmen eines Geschäftsprozeß-Modells gehandhabt wird. Daher ist die optimale Konfiguration der Kontrollparameter insbesondere bei Geschäftsprozeß-Modellen, die als Grundlage für Systeme mit hohem Durchsatz dienen, von großer Bedeutung. Bereits in frühen Phasen des Entwurfs entsprechender Geschäftsprozesse erzielte diesbezügliche Ergebnisse können beispielsweise für die Konfiguration von Workflow Management-Systemen (und dort insbesondere der Workflow-Engine) von großem Nutzen sein [Obe96]. In diesem Abschnitt wird daher ein Ansatz vorgestellt, der es ermöglicht, die Kontrollparameter unter Berücksichtigung von anderen Leistungsparametern in Abhängigkeit von verfügbaren Ressourcen festzulegen.

Ziel der Konfiguration der Kontrollparameter ist das Erreichen eines Systemzustandes, in dem die Bearbeitung einer bestimmten Anzahl von Cases pro Zeiteinheit möglich ist. Die Anzahl der Cases, die zu bearbeiten sind, wird mittels einer *Ankunftsrate*  $\lambda > 0$  modelliert. In den meisten Fällen wird der Ankunftsprozeß als Poisson-Prozeß angenommen, d.h. die Zwischenankunftszeit ist negativ exponentiell verteilt. Andere Verteilungen sind jedoch ebenso denkbar.

Da letzten Endes vor allem der mögliche *Durchsatz* und damit das *steady-state behaviour* [AE01] des Geschäftsprozeß-Modells von Interesse ist, spielt die Anfangsmarkierung lediglich zur Generierung aller möglichen Abläufe eine Rolle.

Zur Illustration des Ansatzes führen wir zunächst ein sehr übersichtliches Geschäftsprozeß-Modell ein:

### **Beispiel**

Abbildung 7.16 zeigt ein Geschäftsprozeß-Modell als *VIPbusiness*-Netz mit Kontrollparametern. Das Geschäftsprozeß-Modell enthält drei Auswahlentscheidungen. Dabei ist eine Auswahlentscheidung (zwischen D und F) bereits mit festen Wahrscheinlichkeiten belegt. Die übrigen beiden Auswahlentscheidungen (zwischen G und I beziehungsweise zwischen J und K) sind mittels der Kontrollparameter  $cp_1$  beziehungsweise  $cp_2$  parametrisiert.

Die Anfangsmarkierung  $m_0$  erlaubt die Generierung aller möglichen Abläufe des Geschäftsprozeß-Modells. Als Ankunftsrate wurde  $\lambda = 10$  gewählt.

□

---

<sup>18</sup>vgl. Abschnitt 2.3.1.

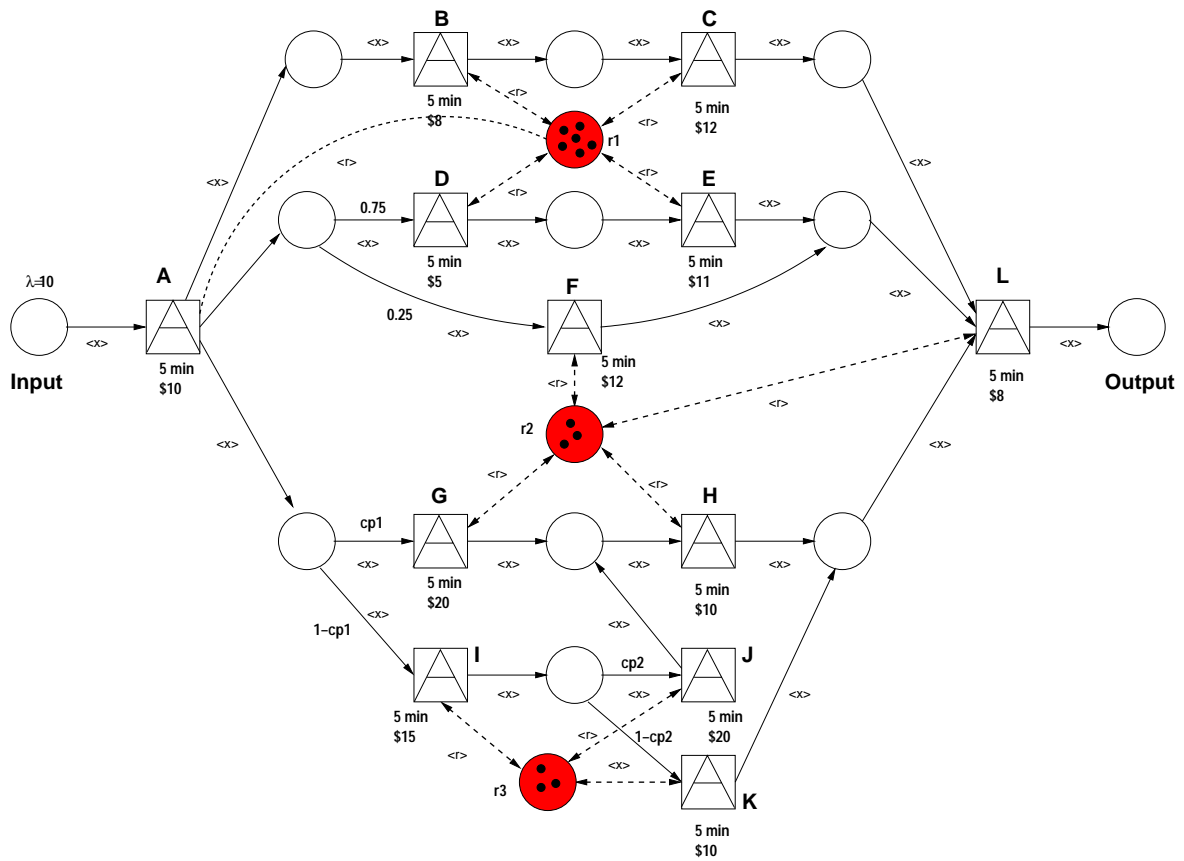


Abbildung 7.16: *VIPbusiness*-Netz mit Kontrollparametern und Ankunftsrate.

Auswahlentscheidungen können auch ohne Kontrollparameter modelliert werden (damit entsprechen sie der klassischen *Konflikt-Situation*). In diesem Fall werden alle Auswahlentscheidungen als *gleich* wahrscheinlich angenommen. Da die Wahrscheinlichkeit von Auswahlentscheidungen im allgemeinen schwierig zu berechnen ist, wird an dieser Stelle vereinfachend angenommen, daß die betrachteten Geschäftsprozeß-Modelle die *Free-Choice*-Eigenschaft [DE95] erfüllen.

Aus Gründen der Übersichtlichkeit führen wir noch die folgenden abkürzenden Schreibweisen für bereits bekannte Sachverhalte ein (vgl. Abschnitt 5.3):

- $nr : R \rightarrow \mathbb{N}$  bezeichnet die *Anzahl der verfügbaren Ressourcen* einer Ressourcen-Klasse. Damit entspricht  $nr(r)$  der Markierung  $m(r)$  einer Ressourcen-Klasse  $r$ .
- $ra : \tau_A \rightarrow R$  bezeichnet die Allokation einer Ressource zu einer Aktivität (modelliert durch eine Aktivitäts-Transition). Mit Hilfe von  $ra$  kann die zugehörige Flußrelation  $F_R = \{(t, r) \in T \times R \mid ra(t) = r\} \cup \{(r, t) \in R \times T \mid ra(t) = r\}$  angegeben werden.

Der Ansatz zur Konfiguration der Kontrollparameter besteht insgesamt aus neun Schritten:

1. Der Geschäftsprozeß wird als *VIPbusiness*-Netz modelliert.
2. Es werden alle relevanten Abläufe generiert<sup>19</sup>.  
Ressourcen werden dabei nicht berücksichtigt. Die Menge der generierten Abläufe sei als  $Q$  bezeichnet, die Anzahl der generierten Abläufe als  $l$ .
3. Für jeden generierten Ablauf wird die Wahrscheinlichkeit berechnet.  
Die Wahrscheinlichkeit eines Ablaufs  $q \in Q$  hängt von Kontrollparametern sowie externen Faktoren ab. Externe Faktoren sind nicht beeinflussbar und werden daher wie eingangs dieses Abschnitts beschrieben mittels fester Wahrscheinlichkeiten modelliert. Die Werte der Kontrollparameter sind dagegen innerhalb ihres Wertebereichs frei wählbar. Seien  $cp_1, \dots, cp_n$  die Kontrollparameter. Dann kann die Wahrscheinlichkeit eines Ablaufs  $q = (B_q, E_q, K_q)$  in Abhängigkeit von  $cp_1, \dots, cp_n$  mit  $w_q(cp_1, \dots, cp_n)$  angegeben werden.
4. Für jeden Ablauf wird unter Annahme unendlicher Ressourcen und der Verwendung des in Abschnitt 7.2.3 eingeführten Verfahrens die Durchlaufzeit berechnet.  
Die berechneten Werte werden im letzten Schritt mit der ressourcenbeschränkten Lösung verglichen. Alternativ können an dieser Stelle natürlich auch irgendwelche anderen Werte für die Durchlaufzeit als Vergleichswerte postuliert werden.
5. Für jede Ressourcen-Klassen  $r$  wird die parametrisierte Ressourcen-Auslastung  $\rho_r$  berechnet:

$$\rho_r(cp_1, \dots, cp_n) = \frac{\lambda}{nr(r)} \sum_{i=1}^l \left( w_i(cp_1, \dots, cp_n) \sum_{e \in E_i \mid ra(\beta(e))=r} pt(\beta(e)) \right)$$

---

<sup>19</sup>Zur Generierung von Analyse-relevanten Teilmengen der möglichen halbgeordneten Abläufe vgl. auch Abschnitt 6.2.

Dabei ist:

- $nr(r)$  die Anzahl der Ressourcen der Ressourcen-Klasse  $r$ .
  - $ra(\beta(e))$  ist die Zuweisung der Ressource zur Aktivität  $\beta(e)$ . An dieser Stelle sei daran erinnert, daß Ressourcen Aktivitäts-Transitionen zugewiesen werden, bei  $e$  handelt es sich also um ein Aktivitäts-Ereignis.
6. Für jede Ressourcen-Klasse  $r \in R$  wird die (Neben-)Bedingung  $\rho_r(cp_1, \dots, cp_n) \leq 1$  formuliert.

Auf diese Weise wird sichergestellt, daß die Ressourcenauslastung nicht über 100 % liegt. Als Ergebnis dieses Schrittes erhalten wir eine Menge von Ungleichungen in Abhängigkeit der Kontrollparameter. Zusätzlich können noch andere Nebenbedingungen formuliert werden, beispielsweise eine obere Grenze für die Durchlaufzeit oder Kostenbeschränkungen. Zusätzlich zu den Nebenbedingungen kann eine Zielfunktion in Abhängigkeit von relevanten Leistungsparametern formuliert werden, beispielsweise eine Minimierung der Kosten oder eine Minimierung der Durchlaufzeit.

7. Das aus dem vorherigen Schritt resultierende Ungleichungssystem wird gelöst.

Falls aufgrund in Konflikt stehender (Neben-)Bedingungen keine Lösung existiert, müssen die Annahmen geprüft und die Bedingungen gegebenenfalls neu aufgestellt werden. Falls mehrere Lösungen existieren, kann eine ausgewählt werden.

8. Mit Hilfe einer auf der Ressourcen-Auslastung basierenden Heuristik wird eine Annäherung der durchschnittlichen Durchlaufzeit berechnet.

Die Heuristik berücksichtigt neben der bekannten Aktivitätszeit die durch Nichtverfügbarkeit von Ressourcen induzierte Wartezeit:

$$ft(t) = (W_{\rho_r, pt_r, nr(r)}) \cdot \left( \frac{c_\lambda^2 + c_r^2}{2} \right) + pt(t)$$

Dabei ist

- $r = ra(t)$  die der Aktivität  $t$  zugeordnete Ressourcen-Klasse.
- $\rho_r$  ist die Ressourcenauslastung nach Berücksichtigung der in Schritt 6 berechneten Lösung.
- $c_\lambda = \lambda\sigma_\lambda$  ist der Variations-Koeffizient des Ankunftsprozesses ( $\sigma_\lambda^2$  ist die Varianz).
- $c_r = \frac{\sigma_r}{pt_r}$  ist der Variations-Koeffizient der Aktivitätszeiten ( $\sigma_r^2$  ist die Varianz).
- $pt_r$  ist die durchschnittliche Aktivitätszeit (die Berechnung erfolgt auf die gleiche Art und Weise wie für  $\rho_r$ ).
- $W_{\rho, pt, s}$  ist die durchschnittliche Wartezeit in einer  $M|M|s$ -Warteschlange mit  $s = nr(r)$  Schaltern, einer Auslastung  $\rho = \rho_r$  und einer durchschnittlichen Aktivitätszeit  $pt = pt_r$ .

9. Vergleiche die berechneten Durchlaufzeiten mit den in Schritt 3 berechneten Werten.

Falls die Diskrepanz zu hoch und das Ergebnis damit unbefriedigend ausfällt, müssen die Bedingungen sowie die Zielfunktion in Schritt 4 angepaßt und die darauffolgenden Schritte wiederholt werden.

□

Die Heuristik in *Schritt 8* benutzt die bekannte Approximation für  $G|G|s$ -Wartesysteme [Kle75]. Dabei handelt es sich um Systeme mit beliebig verteilter Zwischenankunftszeit ( $G$ ), beliebig verteilter Bedienzeit ( $G$ ) und  $s$  Bedienungsschaltern. Die Approximation basiert auf der Formel für die durchschnittliche Wartezeit in  $M|M|s$ -Wartesystemen [Kle75, NM93]. Es wird dabei davon ausgegangen, daß der Variations-Koeffizient des Ankunftsprozesses einer gegebenen Aktivität dem Variations-Koeffizienten des gesamten Ankunftsprozesses entspricht. Dies trifft im allgemeinen nicht zu [BCMP75, BY86, Dij93, GP98]. Für unsere Zwecke ist es jedoch eine akzeptable Annäherung, die eine detaillierte Analyse der Ankunfts- und Abgangsprozesse innerhalb des Geschäftsprozeß-Modells erlaubt.

Prinzipiell wäre es möglich, die in *Schritt 8* mittels der Heuristik berechneten Durchlaufzeiten dem Ungleichungs-System in *Schritt 7* hinzuzufügen. Da die Wartezeit jedoch von den Kontrollparametern  $cp_1, \dots, cp_n$  abhängt, ist die Durchlaufzeit als Ganzes ebenfalls eine von diesen Kontrollparametern abhängige Größe. Beispielsweise für den Fall von Iteration ist nicht einfach möglich, die Durchlaufzeit als von  $cp_1, \dots, cp_n$  abhängige Größe zu formulieren. Daher berechnen wir in *Schritt 7* zunächst Werte für die Kontrollparameter, um in *Schritt 8* mittels der Heuristik Schätzwerte für die Durchlaufzeit unter Berücksichtigung des Einsatzes von Ressourcen zu erhalten.

Die in *Schritt 6* formulierten Nebenbedingungen und Zielfunktionen sind im allgemeinen nicht linear, meistens handelt es sich um polynomiale Ungleichungen. Techniken der linearen Programmierung reichen daher nur in seltenen Fällen aus. Um zu einer Lösung zu gelangen, kann jedoch auf Standard-Programmpakete wie *Mathematica* (Wolfram Research), *MathLab* (MathWorks), *Maple* (Waterloo Maple) oder *MathCad* (MathSoft) zurückgegriffen werden. Für die Beispiele in diesem Abschnitt wurde *Mathematica* benutzt.

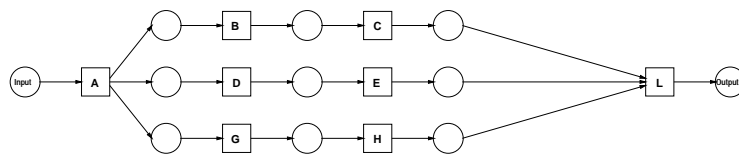
### **Beispiel**

Ausgangspunkt unseres Beispiels ist das Geschäftsprozeß-Modell in Abbildung 7.16 mit Kontrollparametern und Ankunftsrate (*Schritt 1*).

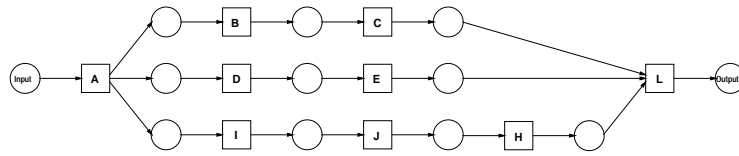
Die Simulation ohne Berücksichtigung der Ressourcen erzeugt 6 Abläufe (vgl. Abbildung 7.17) (*Schritt 2*).

Im Hinblick auf die in Abschnitt 6.2 gemachten Aussagen bezüglich der Effizienz kausaler Semantik bei der Beschreibung des Verhaltens von Systemen mit einem hohen Grad an Nebenläufigkeit sei darauf hingewiesen, daß zur Beschreibung des Verhaltens dieses einfachen Geschäftsprozeß-Modells mittels sequentieller Semantik 570 Schaltfolgen angegeben werden müssen.

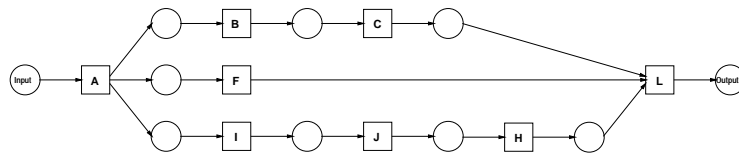




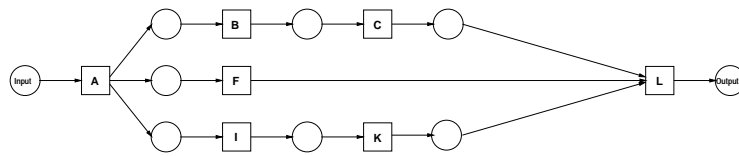
(a) Ablauf 1



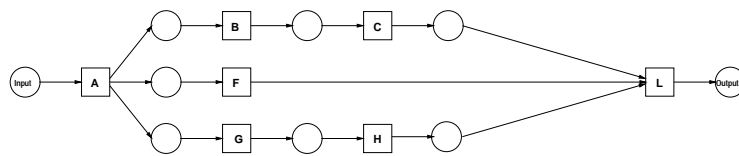
(b) Ablauf 2



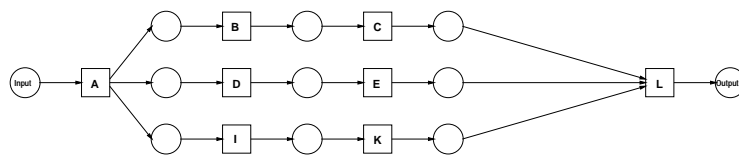
(c) Ablauf 3



(d) Ablauf 4



(e) Ablauf 5



(f) Ablauf 6

Abbildung 7.17: Abläufe des Geschäftsprozess-Modells aus Abbildung 7.16.

In Abhängigkeit der beiden Kontrollparameter  $cp_1$  und  $cp_2$  kann jetzt die Wahrscheinlichkeit  $w_q(cp_1, cp_2)$  für jeden Ablauf angegeben werden (*Schritt 3*):

$$\begin{aligned}
(\text{Ablauf 1}) \quad & w_1(cp_1, cp_2) = 0.75cp_1 \\
(\text{Ablauf 2}) \quad & w_2(cp_1, cp_2) = 0.75(1 - cp_1)cp_2 \\
(\text{Ablauf 3}) \quad & w_3(cp_1, cp_2) = 0.25(1 - cp_1)cp_2 \\
(\text{Ablauf 4}) \quad & w_4(cp_1, cp_2) = 0.25(1 - cp_1)(1 - cp_2) \\
(\text{Ablauf 5}) \quad & w_5(cp_1, cp_2) = 0.25cp_1 \\
(\text{Ablauf 6}) \quad & w_6(cp_1, cp_2) = 0.75(1 - cp_1)(1 - cp_2)
\end{aligned}$$

Für jeden Ablauf wird unter Annahme unendlicher Ressourcen die (minimale) Durchlaufzeit  $t_{D_{min}}(q)$  berechnet<sup>20</sup> (*Schritt 4*):

$$\begin{aligned}
(\text{Ablauf 1}) \quad & t_{D_{min}}(1) = 20min = 0.33h \\
(\text{Ablauf 2}) \quad & t_{D_{min}}(2) = 25min = 0.42h \\
(\text{Ablauf 3}) \quad & t_{D_{min}}(3) = 25min = 0.42h \\
(\text{Ablauf 4}) \quad & t_{D_{min}}(4) = 20min = 0.33h \\
(\text{Ablauf 5}) \quad & t_{D_{min}}(5) = 20min = 0.33h \\
(\text{Ablauf 6}) \quad & t_{D_{min}}(6) = 20min = 0.33h
\end{aligned}$$

Für alle 3 Ressourcen-Klassen  $r_1, r_2, r_3$  wird die parametrisierte Ressourcen-Auslastung  $\rho_{r_i}$  berechnet (*Schritt 5*):

$$\begin{aligned}
\rho_{r_1}(cp_1, cp_2) &= 0.625 \\
\rho_{r_2}(cp_1, cp_2) &= 0.347 + 0.556cp_1 + 0.278cp_2 - 0.278cp_1cp_2 \\
\rho_{r_3}(cp_1, cp_2) &= 0.556 - 0.556cp_1
\end{aligned}$$

Im nächsten Schritt wird das Ungleichungssystem aufgestellt (*Schritt 6*):

Für die Ressourcen-Klassen fordern wir, daß der Grad der Auslastung  $< 0.65$  sein soll. Für  $r_1$  und  $r_3$  ist dies in jedem Fall erfüllt, für  $r_2$  formulieren wir die Bedingung:

$$\begin{aligned}
0.347 + 0.556cp_1 + 0.278cp_2 - 0.278cp_1cp_2 &< 0.65 \\
\Leftrightarrow 2cp_1 + cp_2 - cp_1cp_2 &< 1.09
\end{aligned}$$

Abbildung 7.18 zeigt den Grad der Auslastung der Ressourcen-Klasse  $r_2$  in Abhängigkeit der Parameter  $cp_1$  und  $cp_2$ : Für den Bereich rechts von der Linie ist der Grad der Auslastung  $> 0.65$ . Damit ist klar, welche Werte für  $cp_1$  und  $cp_2$  gewählt werden können. Falls  $cp_1 > 0.55$ , übersteigt der Grad der Ressourcen-Auslastung von  $r_2$  den Wert 0.65. Falls  $cp_1 < 0.09$ , ist  $cp_2$  frei wählbar.

<sup>20</sup>Es sei an dieser Stelle darauf hingewiesen, daß die Zeitbewertungen der Aktivitäts-Transitionen in Abbildung 7.16 in *Minuten* angegeben sind, während für den weiteren Verlauf dieses Beispiels die Einheit *Stunden* gewählt wurde.

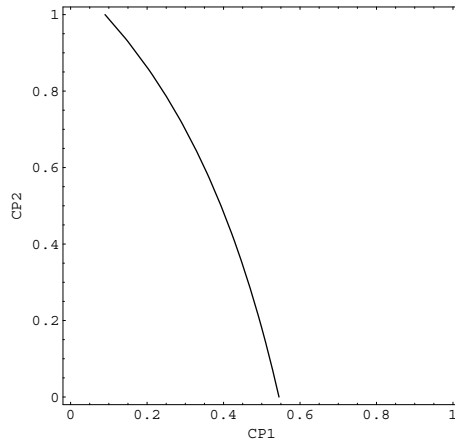


Abbildung 7.18: Der Grad der Auslastung der Ressourcen-Klasse  $r_2$  in Abhängigkeit von  $cp_1$  und  $cp_2$ .

Als Zielfunktion wählen wir die durchschnittlichen Gesamtkosten  $atc$ , die bei der Bearbeitung eines Cases entstehen. Diese können als Summe über alle Abläufe in Abhängigkeit von den Kontrollparametern formuliert werden:

$$atc = \sum_{i=1}^l (cost(i))(w_i)$$

Da für jeden Ablauf die Kosten und die Wahrscheinlichkeit bekannt sind, erhalten wir:

$$atc = 78 + 5cp_1 + 20cp_2 - 20cp_1cp_2$$

Das Ungleichungssystem wird gelöst (*Schritt 7*):

Der Plot in Abbildung 7.19 zeigt die durchschnittlichen Kosten pro Case in Abhängigkeit der Parameter  $cp_1$  und  $cp_2$ : die Farbe *weiß* entspricht dabei maximalen, *schwarz* minimalen Kosten. Die Kosten werden also minimiert, wenn  $cp_1$  und  $cp_2$  auf den Wert 0 gesetzt werden. Damit haben lediglich Ablauf 4 (BC/F/IK) und Ablauf 6 (BC/DE/IK) positive Wahrscheinlichkeiten. Für den Grad der Ressourcen-Auslastung ergibt sich damit:

$$\rho_{r_1}(cp_1, cp_2) = 0.625$$

$$\rho_{r_2}(cp_1, cp_2) = 0.347$$

$$\rho_{r_3}(cp_1, cp_2) = 0.556$$

Damit können im nächsten Schritt die Durchlaufzeiten der Abläufe unter Berücksichtigung beschränkter Ressourcen untersucht werden (*Schritt 8*).

Für den Ankunftsprozeß wird eine Poisson-Verteilung angenommen, daher gilt für den

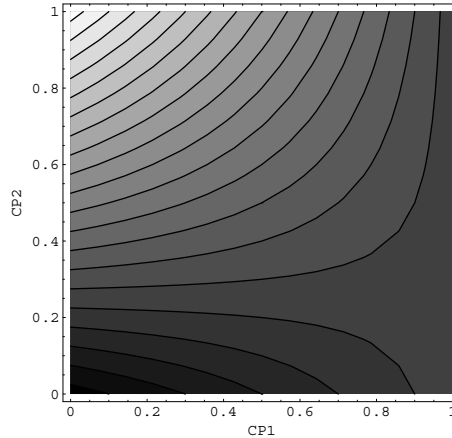


Abbildung 7.19: Die durchschnittlichen Kosten pro Case in Abhängigkeit der Parameter  $cp_1$  und  $cp_2$ : die Farbe *weiß* entspricht dabei maximalen, *schwarz* minimalen Kosten.

Variationskoeffizienten des Ankunftsprozesses  $c_\lambda = 1$ . Für den Variationskoeffizienten der Aktivitätszeiten wählen wir den Wert 0.5, damit erhalten wir  $c_{r_1} = c_{r_2} = c_{r_3} = 0.5$ .

Für die Ressourcen-Klasse  $r_1$  gilt:  $\rho_{r_1} = 0.625$ ,  $pt_{r_1} = 5/60$  und  $nr(r_1) = 6$ . Mittels des bekannten Algorithmus zur Ermittlung durchschnittlicher Wartezeiten in  $M|M|s$ -Wartesystemen [Kle75] erhalten wir:  $W_{\rho_{r_1}, pt_{r_1}, nr(r_1)} = 0.0084$  (Stunden).

Für die Ressourcen-Klasse  $r_2$  mit  $\rho_{r_2} = 0.347$ ,  $pt_{r_2} = 5/60$  und  $nr(r_2) = 3$  erhalten wir:  $W_{\rho_{r_2}, pt_{r_2}, nr(r_2)} = 0.0043$  (Stunden).

Für die Ressourcen-Klasse  $r_3$  mit  $\rho_{r_3} = 0.556$ ,  $pt_{r_3} = 5/60$  und  $nr(r_3) = 3$  erhalten wir:  $W_{\rho_{r_3}, pt_{r_3}, nr(r_3)} = 0.0188$  (Stunden).

Die durchschnittliche Wartezeit, die bei einer Bearbeitung einer Aktivität durch eine Ressource aus  $r_1$  anfällt, ist  $0.0084 * 0.625 = 0.0053$ . Dabei ist  $0.625 = \frac{1^2 + 0.5^2}{2}$  der Kompensationsfaktor der Heuristik. Damit beträgt die durchschnittliche Dauer, die die Ausführung einer Aktivität für einen Case in Anspruch nimmt, 0.0886 Stunden (ungefähr 5.32 Minuten).

Die durchschnittliche Wartezeit, die bei einer Bearbeitung einer Aktivität durch eine Ressource aus  $r_2$  anfällt, ist  $0.0043 * 0.625 = 0.0027$ . Die durchschnittliche Dauer einer Aktivität ist 0.0860 Stunden (ungefähr 5.16 Minuten).

Mit  $r_3$  ergibt sich entsprechend  $0.0188 * 0.625 = 0.0117$  als Wartezeit und 0.0951 Stunden als durchschnittliche Dauer von Aktivitäten (ungefähr 5.70 Minuten).

Mit diesen Werten läßt sich nun die Durchlaufzeit der Abläufe unter Berücksichtigung dees Einsatzes beschränkter Ressourcen berechnen (*Schritt 9*):

Für die Durchlaufzeit der Abläufe 3 und 6 (die beiden Abläufe, die nach der Lösung des

Ungleichungssystem in Schritt 6 noch mit positiven Wahrscheinlichkeiten übrig geblieben sind) ergibt sich als Durchlaufzeit  $5 + 5.70 + 5.70 + 5 = 21.4$  Minuten. Damit entsteht im Durchschnitt nur etwa 1.4 Minuten Wartezeit pro Case. Der Geschäftsprozeß-Entwurf wird mit den veranschlagten Ressourcen also den formulierten Leistungsanforderungen gerecht.

□

## 7.5 Konfigurationen und Szenarien

Nachdem in den vorangegangenen Abschnitten einzelne Verfahren zur Analyse von Geschäftsprozeß-Entwürfen im Vordergrund standen, werden in diesem Abschnitt mit *Konfigurationen* und *Szenarien* auf diesen Verfahren aufsetzende Konzepte zur systematischen Leistungsbewertung von Geschäftsprozeß-Entwürfen eingeführt.

Wie in Abschnitt 3.1.4 bereits erläutert, muß für die Zeit- und Kostenbewertungen der Aktivitäten im Geschäftsprozeßmodell auf Schätz- und Sollwerte zurückgegriffen werden. Um zu verhindern, daß bei der Leistungsbewertung eines Geschäftsprozeß-Entwurfs aufgrund fehlerhafter oder unrealistischer Schätz- und Sollwerte ein falscher Eindruck bezüglich der Leistungsfähigkeit entsteht, sollten die in den vorangegangenen Abschnitten eingeführten Analysemethoden für möglichst viele verschiedene Schätz- und Sollwerte durchgeführt werden. Zusätzlich sollte untersucht werden, welche Auswirkungen die Variation der Bewertungen auf die Leistungsbewertung des modellierten Geschäftsprozesses hat. Mit dem Konzept der *Konfigurationen* für Geschäftsprozeßmodelle wird in Abschnitt 7.5.1 ein entsprechender Ansatz vorgestellt.

Bisher konzentrierten sich die Analyseverfahren im wesentlichen auf die Untersuchung einzelner Abläufe. In Abschnitt 7.5.2 wird erläutert, wie mittels *Szenarien* ein umfassender Eindruck von der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs unter Berücksichtigung mehrerer oder sämtlicher Abläufe gewonnen werden kann.

### 7.5.1 Konfigurationen

Die Aussagekraft der Ergebnisse, die durch Anwendung der in den vorangegangenen Abschnitten vorgestellten Verfahren zur Leistungsbewertung erzielt werden können, ist letzten Endes abhängig von der Güte der verwendeten Schätz- und Sollwerte für die Aktivitätszeiten und -kosten der einzelnen Aktivitäten. Um das Risiko einer falschen Beurteilung der Leistungsfähigkeit eines Geschäftsprozeß-Entwurfs durch zweifelhafte Schätz- und Sollwerte möglichst gering zu halten, ist es sinnvoll, die verschiedenen Verfahren zur Ermittlung leistungsbezogener Kenngrößen für eine Vielzahl verschiedener Werte durchzuführen. Leider ist dieses sicherlich sinnvolle Vorgehen bei der Nutzung der bekannten Ansätze und Werkzeuge für den Geschäftsprozeß-Entwurf mit erheblichem Aufwand verbunden (vgl. Abschnitt 3.1): Für jede Kombination von Werten, die untersucht werden soll, muß das erstellte Geschäftsprozeßmodell komplett neu simuliert werden - ein Vorgang, der durchaus einiges an Zeit und Rechenaufwand verursachen kann. Das liegt daran, daß bei diesen Ansätzen das aufgrund der kausalen Beziehungen der Aktivitäten mögliche Verhalten durch die Einführung der Zeit- und Kostenkonzepte beeinflusst wird (vgl. Abschnitt 5.4). Im Extremfall muß also für die Änderung eines einzigen Werts für eine Aktivität die

komplette Simulation wiederholt werden. Zu diesem Mehraufwand für zusätzliche Simulationen (und dem damit verbundenen Zeitaufwand des Anwenders, der ständig auf die Ergebnisse von Simulationsläufen wartet) kommt noch das Problem der Vergleichbarkeit des während der Simulation generierten Verhaltens hinzu (vgl. Abschnitt 3.1.3). Da dieses Verhalten von den verwendeten Zeit- und Kostenwerten beeinflusst wird, kann es sein, daß das generierte Verhalten für verschiedene Schätz- und Sollwerte so unterschiedlich ist, daß eine klare Analyse des Effekts der unterschiedlichen Zeit- und Kostenwerte nicht mehr möglich ist.

Ausgehend von dieser Problematik ist für den *VIPbusiness*-Ansatz ein Zeit- und Kostenkonzept entwickelt worden, das eine klare Trennung zwischen generiertem Verhalten (in Form der Menge der Abläufe eines Geschäftsprozeßmodells) und den verwendeten Zeit- und Kostenwerten erlaubt. Auf diese Weise ist es möglich, die einmal generierten Abläufe für verschiedene Schätz- und Sollwerte zu untersuchen, ohne daß *Schritt 2* des *VIPbusiness*-Ansatzes wiederholt werden muß.

In den Abschnitten 5.4 und 5.5 wurde erläutert, wie den einzelnen Aktivitäten des Geschäftsprozesses Zeitbewertungen sowie zeitabhängige Kostenfunktionen zugeordnet werden können. Diese Konzepte werden nun um die Möglichkeit der Zuweisung von Konfigurationen zu Geschäftsprozeßmodellen erweitert. Eine *Konfiguration* ist eine Menge von Zeitwerten und zugehörigen Kostenfunktionen, die den Aktivitätszeiten und -kosten (inklusive der entsprechenden Teilgrößen) der einzelnen Aktivitäten beziehungsweise der Aktivitäts-Transitionen und ihrer internen Stellen entspricht. Durch Zuweisung einer Konfiguration zu einem *VIPbusiness-Netz* erhält also jede Aktivitäts-Transition des Netzes (beziehungsweise die internen Stellen) je einen Zeitwert sowie eine (zeitabhängige) Kostenfunktion. Jedem Geschäftsprozeßmodell können beliebig viele Konfigurationen zugeordnet werden.

Echte Wartezeit findet in den Konfigurationen keine Berücksichtigung, da sie erst nach Zuweisung einer Konfiguration zu einem Ablauf mittels der in den Abschnitten 7.2.2 und 7.3.2 vorgestellten Verfahren ermittelt und zugeordnet werden kann. Daher erhalten externe Stellen durch Konfigurationen statt fixer Werte variable Zeitgrößen sowie zeitabhängige Kostenfunktionen.

Bisher ist Aktivitätszeit mittels fester Zeitgrößen modelliert worden. Um eine noch größere Bandbreite verschiedener Schätz- und Sollwerte abdecken zu können, bietet es sich an, diese fixen Werte durch Zufallsfunktionen zu ersetzen. Auf diese Weise lassen sich beispielsweise die Schwankungen modellieren, der die Bearbeitungszeit von Cases aufgrund der unterschiedlichen Fähigkeiten oder Tagesformen der Mitarbeiter unterliegen können. Falls mindestens einer der Aktivitäten statt eines fixen Zeitwerts eine Zufallsfunktion zugeordnet wurde, können für diese Konfiguration verschiedene Mengen von Zufallswerten erzeugt und gespeichert werden. Auf diese Weise wird das Testen unterschiedlicher Werte erheblich erleichtert. Eine Konfiguration, die Zufallsfunktionen enthält, wird als *stochastische Konfiguration* bezeichnet [DE01].

Ausgehend von der Zuweisung einer Konfiguration zu einem Geschäftsprozeßmodell können mittels der generierten Abläufe sowie der in den vorangegangenen Abschnitten vorgestellten Verfahren alle relevanten Kenngrößen auf einfache Weise ermittelt werden.

Durch die Verwendung mehrerer Konfigurationen und (bei der Verwendung stochastischer Konfigurationen) beliebig vieler Mengen von Zufallszahlen als Ausgangspunkt für die Leistungsbewertung des Geschäftsprozeß-Entwurfs läßt sich die Qualität der Analyse-Ergebnisse erheblich verbessern. Da die Verfahren der Leistungsbewertung im Vergleich zur Simulation wenig Zeit und Rechenaufwand in Anspruch nehmen, können beim Entwurfsprozeß Situationen vermieden werden, in denen Design-Entscheidungen aufgrund weniger und möglicherweise falscher oder unrealistischer Schätz- und Sollwerte getroffen werden müssen.

### **Beispiel**

Als Beispiel wird die Durchlaufzeit für verschiedene Konfigurationen untersucht. In Tabelle 7.4 sind dazu in jeder Reihe für alle Aktivitäten des zum Ablauf in Abbildung 7.4 gehörenden Geschäftsprozeßmodells die einzelnen Zeitbewertungen der jeweiligen Konfiguration eingetragen worden. Diesen Werten wird die zugehörige Durchlaufzeit hinzugefügt. Bei Ermittlung der Durchlaufzeiten wurde auf den bereits bekannten Ablauf zurückgegriffen, es mußte lediglich das in Abschnitt 7.2.2 eingeführte Verfahren zur Ermittlung echter Wartezeit angewendet werden.

Es zeigt sich, daß die Durchlaufzeit für verschiedene Konfigurationen relativ konstant zwischen 70 und 95 liegt.

Es muß nun entschieden werden, ob die verschiedenen Durchlaufzeiten akzeptabel sind oder nicht. Gegebenenfalls ist zu überprüfen, ob die in der jeweiligen Konfiguration verwendeten Werte realistisch sind.

□

Die Unterstützung der Untersuchung von Abläufen mittels Konfigurationen durch das *VIPtool* wird in Abschnitt 8.2 anhand eines Fallbeispiels beschrieben.

### **7.5.2 Szenarien**

In diesem Abschnitt wird ein Ansatz zur systematischen Untersuchung eines Geschäftsprozeß-Entwurfs bezüglich einer oder mehrerer Kenngrößen zur Leistungsbewertung vorgestellt.

Ausgangspunkt ist die erfolgte Durchführung von *Schritt 2* des *VIPbusiness*-Ansatzes, d.h., der Geschäftsprozeß-Entwurf ist als *VIPbusiness*-Netz modelliert und die zugehörigen Abläufe mittels *VIPsim* generiert worden (vgl. Abschnitt 3.1.4).

Die Grundidee des Vorgehens besteht darin, die Menge der Abläufe bezüglich der relevanten Kenngrößen durch sukzessive Anwendung der folgenden *Szenarien* zu analysieren:

- 1 Ablauf / 1 Konfiguration (*Szenario 1*)
- 1 Ablauf / alle Konfigurationen (*Szenario 2*)
- alle Abläufe / 1 Konfiguration (*Szenario 3*)
- alle Abläufe / alle Konfigurationen (*Szenario 4*)

	Eingang bestätigen	Akte anlegen	Datenbank einschauen	Schadenshöhe ermitteln	Rückfrage	Überprü- fung?	Durchlaufzeit
Konfiguration 1	10	30	20	30	50	10	90
Konfiguration 2	20	20	10	40	40	10	70
Konfiguration 3	15	25	20	35	55	5	85
Konfiguration 4	20	30	15	30	45	10	85
Konfiguration 5	25	20	15	40	50	10	85
Konfiguration 6	20	15	20	35	40	15	75
Konfiguration 7	15	30	20	30	45	10	85
Konfiguration 8	10	25	15	35	55	15	95
Konfiguration 9	25	35	15	40	55	5	95
Konfiguration 10	20	30	20	30	50	10	90

Tabelle 7.4: Durchlaufzeit für Konfigurationen des Ablaufs aus Abbildung 7.2.



Die Szenarien müssen nicht notwendigerweise in dieser Reihenfolge untersucht werden. Die Art ihrer Nutzung wird vielmehr von den Anforderungen und den Möglichkeiten des Anwenders abhängen.

Die Untersuchung aller Abläufe für eine Konfiguration (*Szenario 3*) vermittelt in der Regel einen soliden ersten Eindruck von der Leistungsfähigkeit des Geschäftsprozeß-Entwurfs für verschiedene Cases und die zugehörigen Routing-Entscheidungen. Große Unterschiede bezüglich der untersuchten Kenngrößen, beispielsweise der Durchlaufzeit, liefern Anhaltspunkte für erste detaillierte Analysen im Hinblick auf möglicherweise notwendige Veränderungen des Geschäftsprozeß-Entwurfs.

Interessante Interpretationsmöglichkeiten ergeben sich für den Fall, daß ein Ablauf für eine oder mehrere Kenngrößen Werte aufweist, die stark von denen der anderen Abläufe abweicht, beispielsweise wird für einen der Abläufe eine besonders niedrige Durchlaufzeit ermittelt. Falls sich herausstellt, daß es sich bei diesem Ablauf um eine besonders leistungsfähige Variante handelt, kann dieser Ablauf als „Vorbild“ genutzt werden, d.h., man sollte versuchen, möglichst viele Cases durch Ausführung der Aktivitäten gemäß der Vorgaben des Ablaufs zu bearbeiten. Umgekehrt kann der niedrige Wert für die Durchlaufzeit aber auch daher resultieren, daß bei der Bearbeitung der Cases wichtige Aktivitäten einfach ausgelassen wurden, die Cases also nicht korrekt bearbeitet worden sind und es sich folglich um einen fehlerhaften Ablauf handelt. Für diesen Fall muß der Geschäftsprozeß-Entwurf dahingehend modifiziert werden, daß dieser Ablauf nicht mehr vorkommen kann, es müssen also gegebenenfalls Routing-Entscheidungen angepaßt werden.

Für den positiven Fall einer besonders leistungsfähigen Variante bietet es sich an, die Gründe für diese besondere Leistungsfähigkeit durch Analyse von *Szenario 1* und *Szenario 2* zu untersuchen. Die Möglichkeit, den Ablauf für einzelne Konfigurationen zu untersuchen, erlaubt beispielsweise die Ermittlung von *best case* oder *worst case*-Annahmen bezüglich der untersuchten Kenngrößen. Die Untersuchung des Ablaufs für verschiedene Konfigurationen vermittelt einen Eindruck davon, wie die Leistungsfähigkeit des Ablaufs von den jeweils verwendeten Schätz- und Sollwerten abhängt.

Für den Fall, daß ein Ablauf konstant gute Ergebnisse für verschiedene Konfigurationen liefert, lohnt sich die Ermittlung der Gründe für die besondere Leistungsfähigkeit. Solche Gründe können beispielsweise ein hoher Anteil nebenläufig ausführbarer Aktivitäten oder die konsequente Nicht-Ausführung eigentlich überflüssiger Aktivitäten sein. Die so gewonnenen Informationen sind wertvolle Anhaltspunkte für eine qualitativ hochwertige Gestaltung eines Geschäftsprozesses. Es sollte untersucht werden, ob es nicht möglich ist, mehrere oder gar alle Cases mittels Ausführung der Aktivitäten nach Vorgabe dieses Ablaufs zu bearbeiten.

Falls keiner der untersuchten Abläufe den gesteckten Zielen bezüglich der Leistungsfähigkeit, beispielsweise eine maximal akzeptable Durchlaufzeit, gerecht wird, muß der Geschäftsprozeß-Entwurf unter Umständen komplett neu gestaltet werden.

Ein Fallbeispiel zur Untersuchung eines Geschäftsprozeß-Entwurfs mittels Szenarien findet sich Abschnitt 8.2.



# Kapitel 8

## Arbeiten mit dem *VIPtool*

Ausgehend von der Entwicklung des Anforderungsprofils in Kapitel 3 geht es in den folgenden Abschnitten um die Umsetzung dieses Profils durch eine Weiterentwicklung des *VIPtools* und die damit erreichte Werkzeugunterstützung der in den vorangegangenen Kapiteln eingeführten *VIPbusiness*-Konzepte.

In Abschnitt 8.1 werden dazu neben einigen technischen Details zur Implementierung die Unterstützung der einzelnen Schritte des *VIPbusiness*-Ansatzes sowie die Verknüpfung des *VIPtools* mit anderen Werkzeugen erläutert. In Abschnitt 8.2 wird anhand einer kleinen Fallstudie demonstriert, wie der Entwurf von Geschäftsprozessen mit Hilfe des *VIPbusiness*-Ansatzes durch das *VIPtool* unterstützt wird.

### 8.1 Das Werkzeug

Das *VIPtool* basiert auf einem Software-Prototyp, der ursprünglich zur Validierung der im Rahmen des *VIP*-Projekts entwickelten Simulations-Konzepte entwickelt wurde (vgl. Abschnitt 6.2). Die ersten Prototypen wurde in der Zwischenzeit erheblich erweitert<sup>1</sup>.

#### 8.1.1 Technische Details und Architektur

Als Ausgangsbasis für die Implementierung des *VIPtools* wurde die objektorientierte, interpretierte Skriptsprache *Python* gewählt [Lut96, Con01]. Diese Sprache eignet sich besonders gut zur Erstellung von prototypischer Software. Da Interpreter für die Betriebssysteme *Linux*, *Windows* und *Sun/Solaris* verfügbar sind, kann das *VIPtool* ohne großen Mehraufwand parallel für diese Betriebssysteme entwickelt werden.

Die Objektorientierung ermöglicht einen modularen Aufbau des *VIPtools*. Dadurch können zum einen neue Funktionalitäten in Form neuer Module auf einfache Weise hinzugefügt werden, zum anderen können laufzeitkritische Komponenten wie die Verfahren zur Simulation oder zur Berechnung der Darstellung generierter Abläufe in performanten Sprachen wie *C* implementiert und über funktionale Schnittstellen (die von *Python* zur Verfügung gestellt werden) als eigenständige Module eingebunden werden.

---

<sup>1</sup>Eine aktuelle Version des *VIPtools* kann über die folgende URL bezogen werden: [www.informatik.ku-eichstaett.de](http://www.informatik.ku-eichstaett.de).

Die graphische Benutzeroberfläche wurde mit Hilfe von *Tkinter* [Gra00] realisiert, einer von *Python* bereitgestellten Schnittstelle zu der weit verbreiteten und ebenfalls plattform-unabhängigen graphischen Toolkit-Bibliothek *Tcl/Tk* [Ous94].

Eine ausführliche Beschreibung der Basis-Module inklusive Klassendiagrammen der verwendeten *Python*-Objekte sowie eine Dokumentation des *VIPsim*-Moduls mit Beschreibung der verwendeten Verfahren und Algorithmen findet sich in [Fre01].

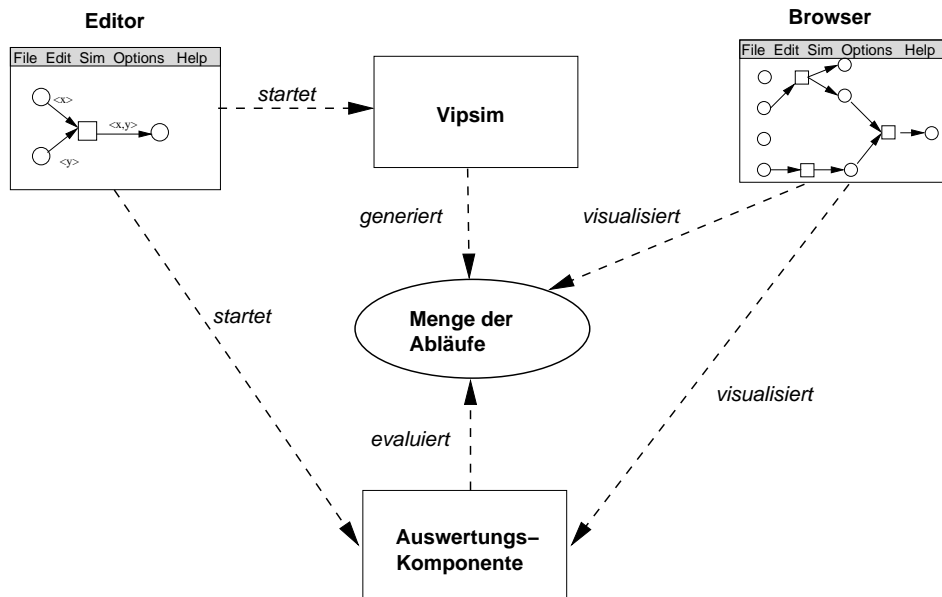


Abbildung 8.1: Architektur des *VIPtools*.

Abbildung 8.1 zeigt die grundlegende Architektur des *VIPtools*. Im Wesentlichen setzt sich das Werkzeug aus vier Komponenten zusammen, von denen zwei, die *Editor*- und die *Browser*-Komponente, über eine graphische Oberfläche zur direkten Kommunikation mit dem Anwender verfügen. Welche Funktionalität die einzelnen Komponenten im Rahmen der Unterstützung des *VIPbusiness*-Ansatzes erbringen, wird im folgenden Abschnitt beschrieben.

### 8.1.2 Unterstützung des *VIPbusiness*-Ansatzes durch das *VIPtool*

Zur Unterstützung des *VIPbusiness*-Ansatzes wurden die bereits existierenden Module des *VIPtools* um die in Kapitel 3 und den nachfolgenden Kapiteln vorgestellten *VIPbusiness*-Konzepte erweitert. In diesem Abschnitt werden die entsprechenden Erweiterungen und die dadurch möglichen Anwendungen des *VIPtools* für den Entwurf von Geschäftsprozessen im Rahmen des 3-stufigen *VIPbusiness*-Ansatzes kurz erläutert.

## Modellierung

Die Erstellung der Geschäftsprozeßmodelle als *VIPbusiness*-Netze erfolgt mit Hilfe der *Editor*-Komponente. Diese stellt sämtliche zur graphischen Modellierung benötigten Funktionalitäten zur Verfügung. Dabei werden alle in den Kapiteln 4 und 5 vorgestellten Bausteine zur Modellierung mit der Klasse der *VIPbusiness*-Netze unterstützt. Abbildung 8.2 zeigt die *Editor*-Komponente mit einem Geschäftsprozeßmodell und geöffnetem Detail-Fenster zur Modellierung der internen Stellen und Transitionen einer Aktivitäts-Transition. Im Prinzip können beliebig viele Geschäftsprozeßmodelle unabhängig voneinander editiert werden.

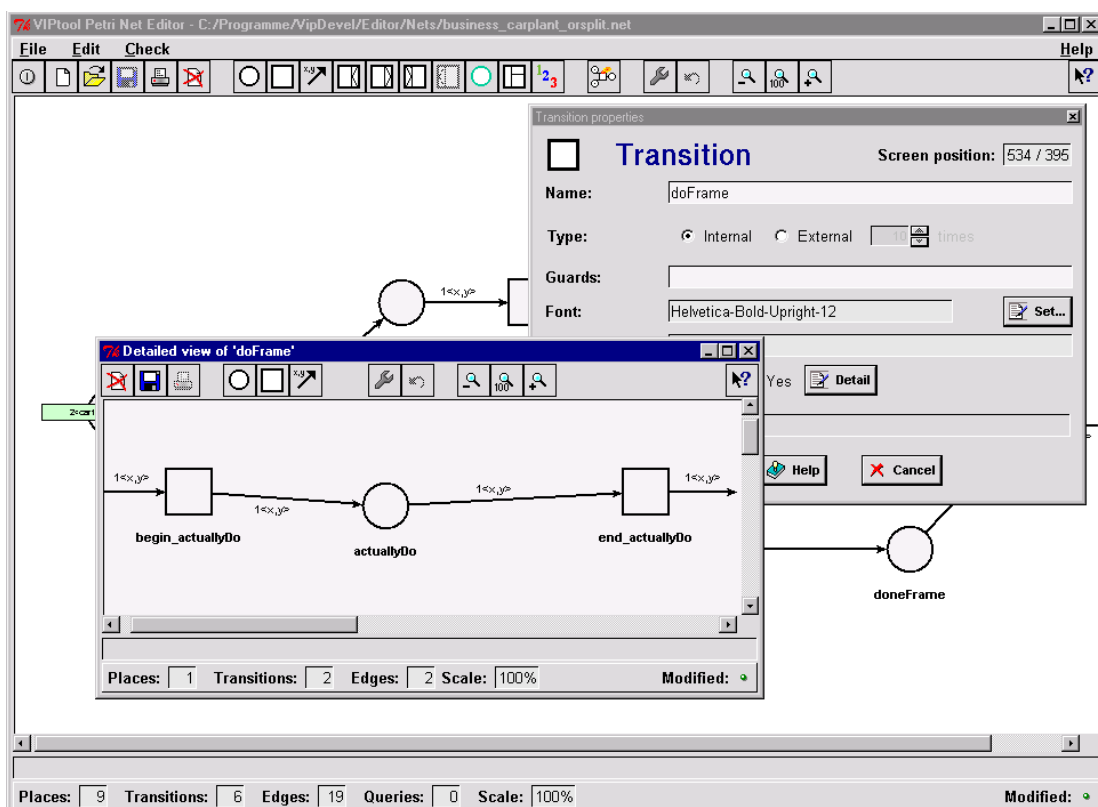


Abbildung 8.2: *Editor*-Komponente des *VIPtools*, Modellierung einer Aktivitäts-Transition.

## Simulation

Die Generierung der halbgeordneten Abläufe des Geschäftsprozeßmodells geschieht mit Hilfe der *VIPsim*-Komponente. Diese greift bei der Generierung der Abläufe auf die in Kapitel 6 beschriebenen *VIP*-Simulations-Konzepte (vgl. auch [Fre01]) sowie die entsprechenden *VIPbusiness*-Erweiterungen zurück. Die so erzeugte Menge der Abläufe wird für die nachfolgenden Analyse-Schritte gespeichert.

Um die Simulation der Geschäftsprozeßmodelle möglichst effizient zu gestalten, können die von *VIPsim* benutzten Techniken wie *Cut-Offs* (vgl. [Esp94, McM92, Fre96] sowie Abschnitt 6.2) über ein graphisches Menü gesteuert werden (vgl. Abbildung 8.3). Eine detaillierte Beschreibung aller konfigurierbaren Simulations-Parameter findet sich in [Fre97a, Fre01].

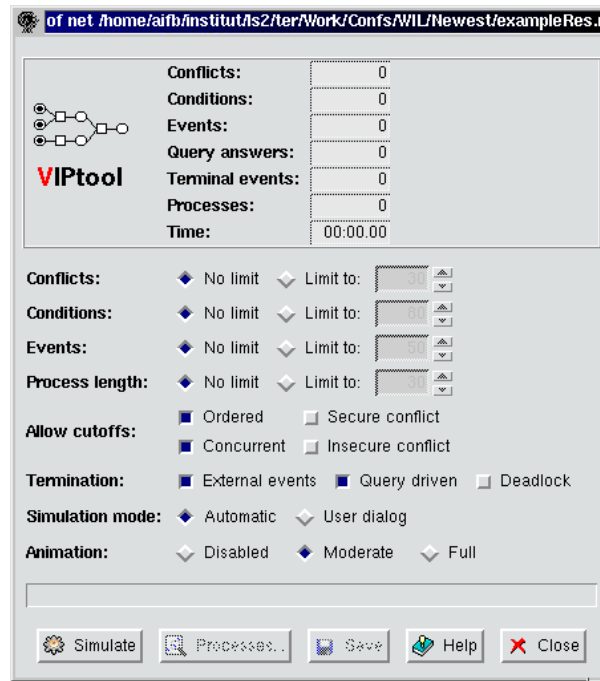


Abbildung 8.3: Menü zur Konfiguration der Simulations-Parameter.

## Analyse

Dem Analyse-Konzept des *VIPbusiness*-Ansatzes entsprechend werden bei der Analyse der Geschäftsprozeßmodelle die drei Bereiche *Validierung*, *Verifikation* und *Leistungsbeurteilung* berücksichtigt (vgl. Abschnitt 3.1.3).

Wesentlicher Punkt bei der *Validierung* der Geschäftsprozeßmodelle ist die explizite graphische Darstellung des möglichen Verhaltens. Die *Browser*-Komponente des *VIPtools* ermöglicht die Darstellung sowie die systematische Durchsicht der erzeugten halbgeordneten Abläufe in Form von Kausalnetzen (vgl. Abbildung 8.4).

Für Verfahren zur Überprüfung grundlegender Eigenschaften wie *Soundness* im Rahmen der *Verifikation* wird im wesentlichen auf die Funktionalität des mit dem *VIPtool* über eine funktionale Schnittstelle verknüpften Analyse-Werkzeuges *Woflan* zurückgegriffen (vgl. dazu auch die Ausführungen im nachfolgenden Abschnitt 8.1.3). Darüber hinaus kann die *restriktive Korrektheit* der Abläufe, also die Erfüllung der mit Hilfe von Anfrage-Transitionen formulierten Business Rules, im Browser überprüft werden. Verletzungen von Business Rules werden hier graphisch angezeigt.

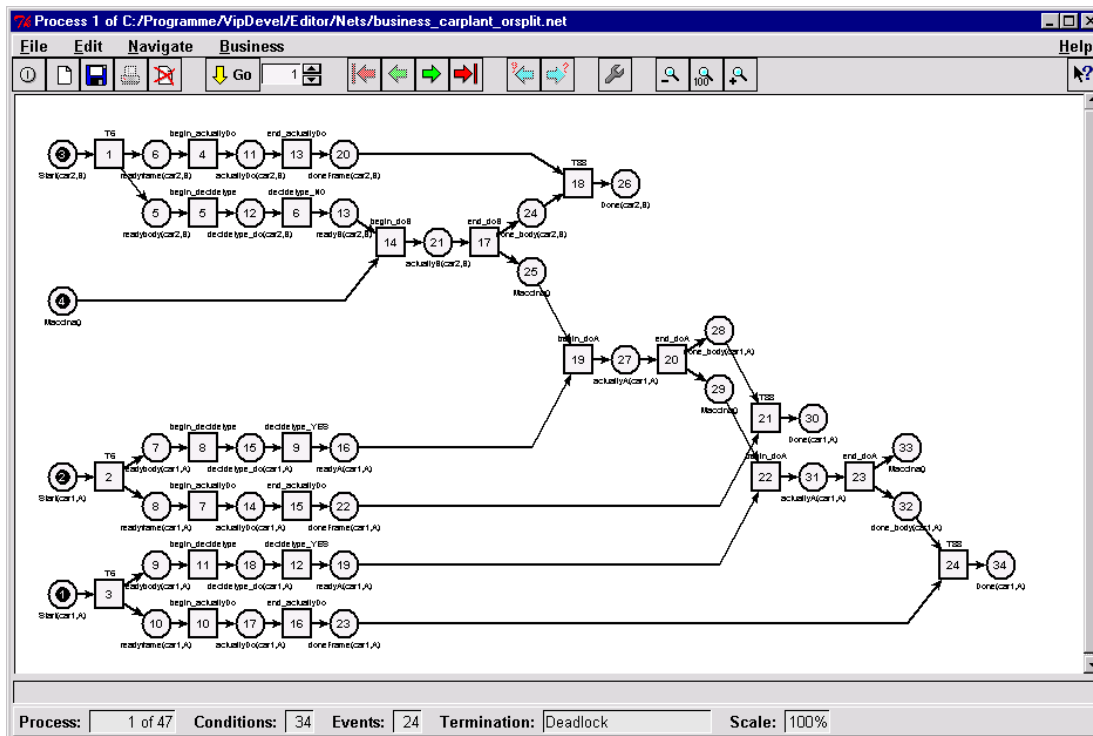


Abbildung 8.4: Browser für die generierten Abläufe.

Die Verfahren zur *Leistungsbewertung* sind in der *Auswertungs*-Komponente zusammengefaßt. Als wesentlicher Input dienen dabei die Informationen aus der Editor-Komponente wie die verfügbaren Ressourcen oder die verwendeten Konfigurationen für Zeit- und Kostenwerte. Abbildung 8.5 zeigt die Oberfläche zum Erstellen, Editieren und Verwalten von Konfigurationen. Der Aufruf der einzelnen Verfahren kann wahlweise über die Menüs der *Editor*- oder der *Browser*-Komponente erfolgen. Dabei werden insbesondere auch systematische Analysen über mehrere oder alle für ein Geschäftsprozeßmodell existierenden Zeit- und Kosten-Konfigurationen unterstützt.

Zur graphischen Aufbereitung können die Analyse-Ergebnisse über eine Schnittstelle in gängige Programme zur Tabellen-Kalkulation exportiert werden (vgl. den nachfolgenden Abschnitt 8.1.3 sowie die Abbildungen 8.6 und 8.10).

### 8.1.3 Verknüpfung mit anderen Werkzeugen

Aufgrund des modularen Aufbaus des *VIPtools* ist es möglich, wahlweise Teile oder die gesamte Funktionalität mit anderen Werkzeugen zu verknüpfen beziehungsweise die Funktionalität anderer Werkzeuge für die mit Hilfe des *VIPtools* erstellten Geschäftsprozeßmodelle nutzbar zu machen. So können beispielsweise Teile der *VIPsim*-Komponente sowie des *Browsers* als Plug-Ins für den an der Humboldt-Universität zu Berlin entwickelten *Petri-netz-Kern* genutzt werden [DK96, KSW97a, KSW97b, Erw98].

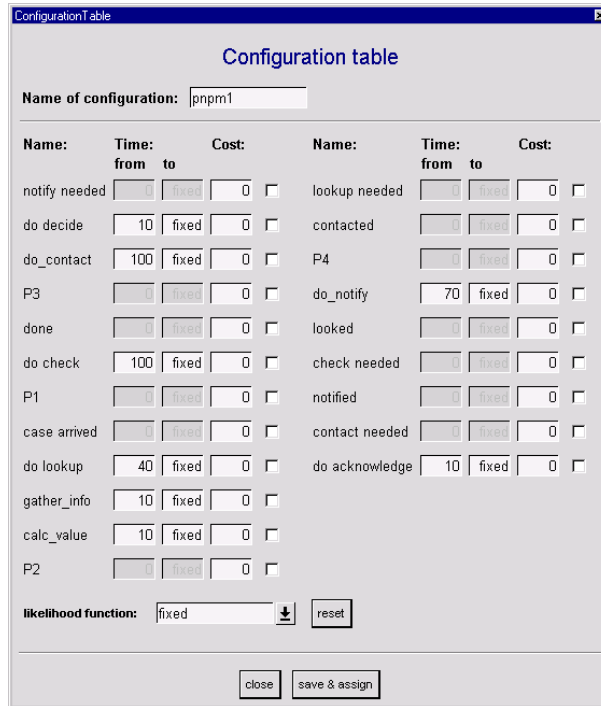


Abbildung 8.5: Oberfläche zum Erstellen und Editieren von Konfigurationen.

Im Hinblick auf die Überprüfung der *Konsistenz* von Geschäftsprozeß-Modellen ist die Verknüpfung des *VIPTools* mit dem an der TU Eindhoven entwickelten Analysewerkzeug *Woflan* von besonderem Interesse. Bei *Woflan* (*Workflow analyzer*) handelt es sich um ein Werkzeug zur Analyse von Geschäftsprozeß-Spezifikationen in Form sogenannter *Workflow Nets* [Aal97, Aal98d]. Der Schwerpunkt der Analyse liegt auf Verfahren aus dem Bereich der Verifikation<sup>2</sup>. *Woflan* wurde ursprünglich zur Analyse von Modellen entwickelt, die als Grundlage für den Entwurf von Steuerungen für *Workflow Management Systeme (WfMS)* verwendet werden können. In den letzten Jahren ist es konsequent zu einem vielfältig einsetzbaren Werkzeug zur Verifikation von Petrinetz-basierten Geschäftsprozeßmodellen erweitert worden [HVA97, AHV97, Aal99, VBA01].

Durch die Integration von *Woflan* und dem *VIPTool* ist es möglich, die mit Hilfe des *VIPTools* entworfenen Geschäftsprozeßmodelle bezüglich einiger grundlegender wünschenswerter Eigenschaften wie *Soundness* zu untersuchen (vgl. Abschnitt 7.1.1). Durch die Nutzung dieser Analyse-Möglichkeiten direkt nach der Erstellung der Geschäftsprozeßmodelle kann vermieden werden, daß die teilweise sehr aufwendigen Simulationen für grundsätzlich fehlerbehaftete Geschäftsprozeßmodelle durchgeführt werden. Darüber hinaus ist es möglich, Modelle zu untersuchen, die mit Hilfe von Produkten aus dem Workflow-Bereich wie *Staffware* [Sta97], *COSA* [SL96], *Protos* [Pal97] oder *METEOR* [LSKM00] erstellt wurden. Abbildung 8.6 zeigt die Architektur des *VIPTool/Woflan*-Toolkits. Dabei nimmt

<sup>2</sup>Eine aktuelle Version Woflan kann über den Link <http://www.win.tue.nl/~woflan/download.htm> heruntergeladen werden.



*Woflan* eine Schnittstellenfunktion zu wichtigen Workflow-Produkten ein und dient zur Verifikation von Geschäftsprozeß-Entwürfen. Das *VIPtool* erlaubt die Erstellung der entsprechenden Geschäftsprozeßmodelle und ermöglicht zugleich vor allem deren Validierung und Leistungsbewertung.

Zusätzlich zu den oben beschriebenen Verknüpfungen ist für das *VIPtool* eine Schnittstelle für den Export von Ergebnissen der Leistungsbewertung zu Programmen aus dem Bereich der Tabellenkalkulation realisiert worden, beispielsweise zu *Excel* der Firma *Microsoft* (vgl. dazu auch das Beispiel in Abschnitt 8.2).

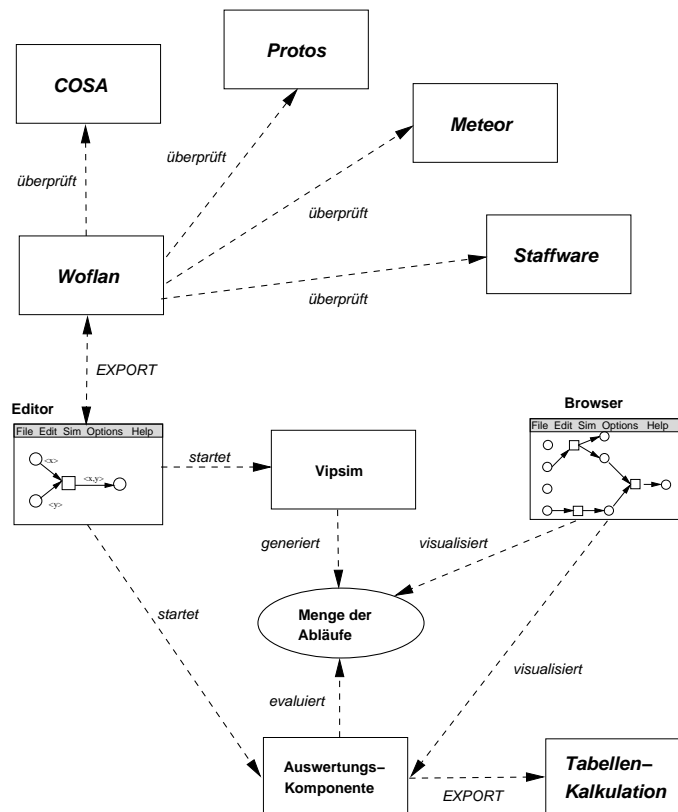


Abbildung 8.6: Architektur des *VIPtool/Woflan*-Toolkits [AE01].

## 8.2 Eine kleine Fallstudie

In diesem Abschnitt wird mit Hilfe eines kleinen Beispiels gezeigt, wie die in dieser Arbeit entwickelten *VIPbusiness*-Konzepte, unterstützt durch das *VIPtool*, für den Entwurf von Geschäftsprozessen genutzt werden können. Bei dem Beispiel handelt es sich um eine vereinfachte Variation des schon in den vorangegangenen Kapiteln benutzten Geschäftsprozesses für die Bearbeitung von Schadensmeldungen in einer Versicherung. Nach einer kurzen Beschreibung des Geschäftsprozesses und Erstellung eines ersten Entwurfs wird gezeigt, wie die Verfahren zur Simulation und Analyse für eine Bewertung der Qualität dieses Entwurfs genutzt werden können.

## Das Beispiel

Ausgangspunkt ist die Betrachtung einer Abteilung zur Bearbeitung von Schadensmeldungen in einer Versicherung. Ein neuer Case zur Bearbeitung wird durch das Eintreffen einer Schadensmeldung generiert. Der Eingang der Schadensmeldung wird bestätigt, davon unabhängig wird eine Akte zur Dokumentation der Bearbeitung angelegt. Nach diesen Vorbereitungen kann mit der eigentlichen Bearbeitung der Schadensmeldung begonnen werden. Die folgenden Aktivitäten müssen ausgeführt werden: die Gegenseite muß über den Vorgang informiert werden, der Kunde wird für detailliertere Informationen kontaktiert. Zusätzlich müssen einige Daten überprüft werden (beispielsweise ob der Kunde überhaupt versichert ist, die Schadenshistorie des Kunden etc.). Sofern keine gemeinsam genutzten Ressourcen benötigt werden, können diese drei Aktivitäten unabhängig voneinander ausgeführt werden. In Abhängigkeit von der Höhe des Schadens kann eine zusätzliche Überprüfung der Bearbeitung nötig sein, dies ist ab einer Schadenshöhe von 10000 der Fall. Nach der Bearbeitung beziehungsweise einer eventuellen zusätzlichen Überprüfung wird entschieden, ob die Versicherung für den Schaden aufkommt oder nicht. Für unser Fallbeispiel betrachten wir allerdings ausschließlich die Bearbeitung der Schadensmeldungen, die abschließende Entscheidung über die Zahlung wird nicht berücksichtigt.

## Modellierung

Dem 3-stufigen Ansatz folgend (vgl. Kapitel 3) wird ein erster Entwurf des Geschäftsprozesses als *VIPbusiness*-Netz erstellt. Dies geschieht mit Hilfe der Editor-Komponente des *VIPtools*, die alle benötigten Funktionalitäten zur Verfügung stellt (vgl. Abschnitt 8.1.2). Abbildung 8.7 zeigt die Editor-Komponente des *VIPtools* mit einem ersten Entwurf des Geschäftsprozesses.

An dieser Stelle könnte das Geschäftsprozeßmodell nach *Woflan* exportiert werden, um einige grundlegende Eigenschaften wie die *Soundness* des Entwurfs überprüfen zu können (vgl. Abschnitt 7.1). Auf diese Weise lassen sich bereits eine Reihe möglicher Fehler des Entwurfs beseitigen, bevor aufwendige Simulationen durchgeführt werden. Für das hier betrachtete Beispiel ist der Nachweis dieser grundlegenden Eigenschaften vergleichsweise einfach, weshalb wir uns direkt denjenigen Analysemöglichkeiten zuwenden, die auf einer Simulation des Verhaltens basieren.

## Erste Simulation und Analyse

In einem ersten Schritt wird das Geschäftsprozeßmodell mit einer Anfangsmarkierung versehen. In Abbildung 8.7 ist die Stelle `Case arrived` mit der Marke  $\langle c1, 11000 \rangle$  versehen worden, wobei *c1* als Identifikator für den Case dient und 11000 die Höhe der Schadensmeldung darstellt. Die Höhe der Schadensmeldung wird bei der Bearbeitung als Information zur Entscheidung über eine zusätzliche Überprüfung der Bearbeitung benötigt. Diese Information ist eigentlich erst nach der Bearbeitung des Case bekannt. Da wir weder externe Datenquellen noch zufällig generierte Werte als Informationen in unserem Modell berücksichtigen, muß jede für die Bearbeitung von Cases benötigte Information, insbesondere für Routing-Entscheidungen, schon im Vorfeld der Simulation in Form entsprechender Marken modelliert werden. Dieser Nachteil wird ausgeglichen, indem systematisch

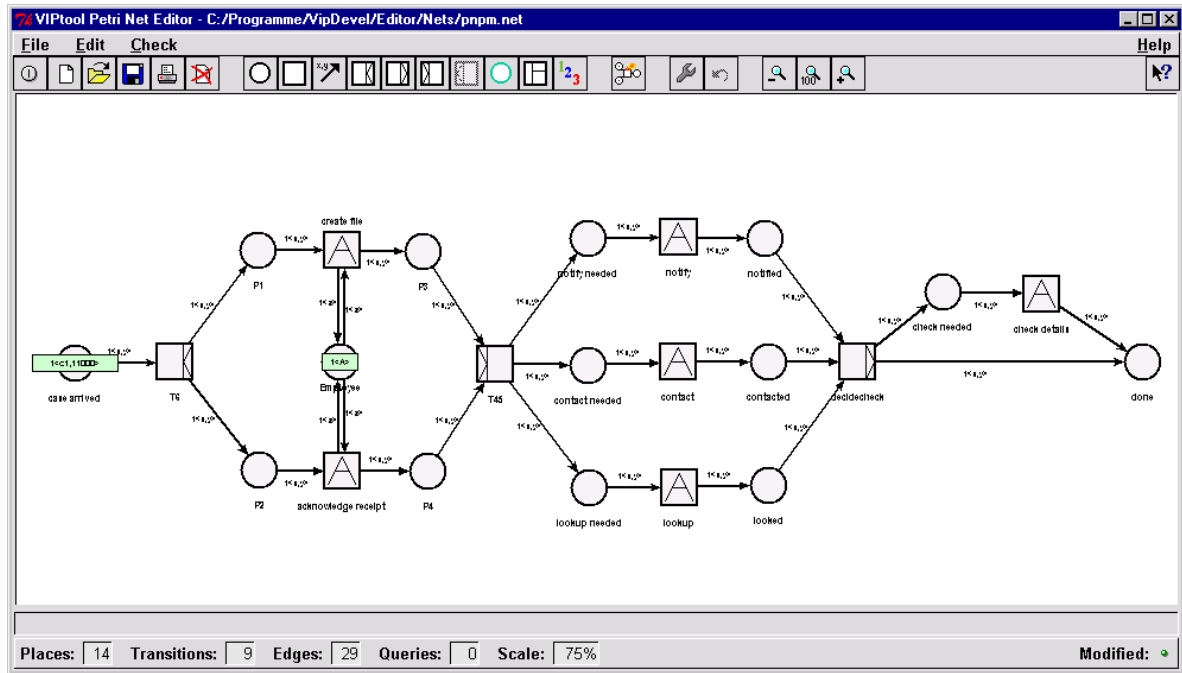


Abbildung 8.7: Ein erster Entwurf des Geschäftsprozesses.

verschiedenen Anfangsmarkierungen generiert werden. Dabei werden sowohl sämtliche erwarteten Ausprägungen der später zu bearbeitenden echten Cases als auch alle bei der Bearbeitung möglichen Routing-Entscheidungen berücksichtigt. Auf diese Weise wird sichergestellt, daß alle möglichen Pfade im Geschäftsprozeßmodell durchlaufen werden, beziehungsweise alle bei der Bearbeitung eines einzelnen Case möglichen halbgeordneten Abläufe generiert werden.

Im vorliegenden Beispiel gibt es nur eine Routing-Entscheidung: falls die Höhe des Schadens den Wert 10000 übersteigt, muß eine zusätzliche Überprüfung der Bearbeitung vorgenommen werden. Durch Hinzufügen einer zweiten Marke  $\langle c2, 5000 \rangle$  erhalten wir beide möglichen Ergebnisse der Routing-Entscheidung.

Für kleine Beispiele ist es in der Regel sehr einfach, die Anzahl der während der Simulation generierten Abläufe und die dabei getroffenen Routing-Entscheidungen vorherzusagen. Bei größeren Geschäftsprozeßmodellen wird es jedoch sehr schwierig, den Überblick über alle zulässigen Kombinationen von Routing-Entscheidungen zu behalten und damit die Generierung von fehlerhaften Abläufen zu vermeiden. Durch die graphische Darstellung von Abläufen mit Hilfe der Browser-Komponente des *VIPtools* werden unerwünschte Kombinationen von Routing-Konstrukten und damit verbundene Routing-Entscheidungen sofort sichtbar.

Mit der *VIPsim*-Komponente des *VIPtools* werden zu jedem Case halbgeordnete Abläufe erzeugt, die einer möglichen Bearbeitung des Case entsprechen. Abbildung 8.8 zeigt den Browser mit dem Ablauf für die Bearbeitung des Case  $\langle c1, 11000 \rangle$ , dargestellt als

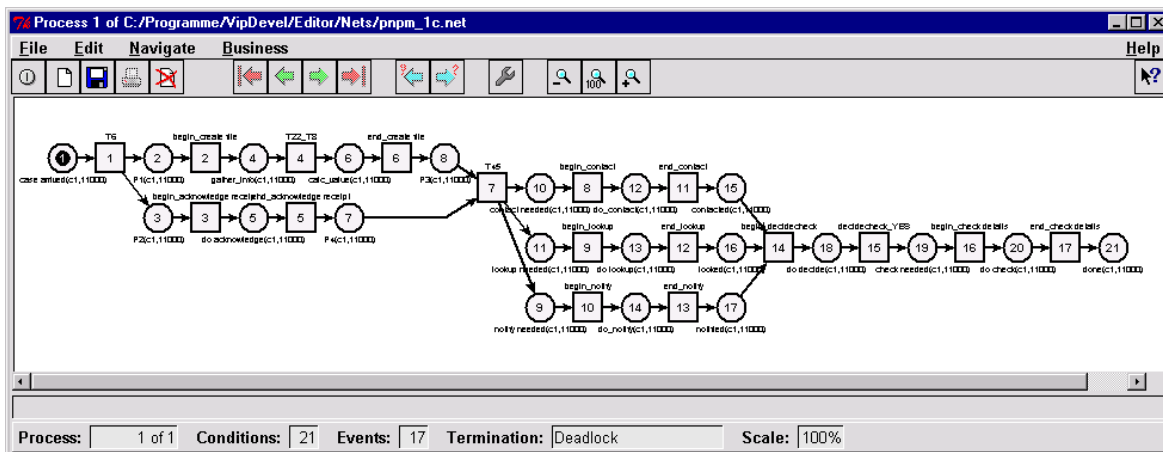


Abbildung 8.8: Ein Ablauf des Geschäftsprozeßmodells aus Abbildung 8.7.

Kausalnetz. Ressourcen werden dabei erst einmal nicht berücksichtigt. Daher wird für die Ausführung der Aktivitäten **Create File** und **Acknowledge Receipt** zuständige Ressource **Employee** zunächst ignoriert. Da der konkurrierende Zugriff auf geteilte Ressourcen die einzige Form der Interaktion von Cases darstellt, hätten wir auch die Marken  $\langle c1, 11000 \rangle + \langle c2, 5000 \rangle$  als Anfangsmarkierung benutzen können. Das Ergebnis wäre ein Ablauf zur (nebenläufigen) Bearbeitung beider Cases gewesen. Wir werden diese Anfangsmarkierung später verwenden, wenn die Bearbeitung mehrerer Cases unter Berücksichtigung der dazu benötigten Ressourcen betrachtet wird. Im Moment begnügen wir uns damit, für alle relevanten Ausprägungen möglicher zu bearbeitender Cases zu zeigen, daß ihre Bearbeitung durch Ausführung der Aktivitäten gemäß des erstellten Geschäftsprozeßmodells möglich ist. Dies wird dadurch erreicht, daß für jeden Case ein Ablauf erzeugt wird, dessen minimaler (maximaler) Schnitt den markierten Input-Stellen (Output-Stellen) des Geschäftsprozeßmodells entspricht, in unserem Beispiel die Stellen **Case Arrived** und **Done**.

### Berücksichtigung von Zeit

Nachdem überprüft worden ist, daß alle Arten von Cases zumindest einzeln in sinnvoller Weise bearbeitet werden können, soll nun die Durchlaufzeit als eine der wichtigsten Kennzahlen bei der Leistungsbewertung eines Geschäftsprozeß-Entwurfs untersucht werden. Da das betrachtete Beispiel keine Entscheidungen enthält, die nicht durch die im Vorfeld festzulegenden Routing-Informationen (siehe oben) determiniert wäre, existiert genau ein Ablauf. Daher wird das Modell zunächst um Ressourcen erweitert, um möglicherweise auftretende Linearisierungs-Effekte zu untersuchen.

### Berücksichtigung von Ressourcen: Simulation und Analyse

Wie in Abschnitt 7.4 bereits erläutert, kann es bei konkurrierendem Zugriff auf knappe Ressourcen zu Linearisierungen halbgeordneter Abläufe kommen. Dabei müssen vormalis

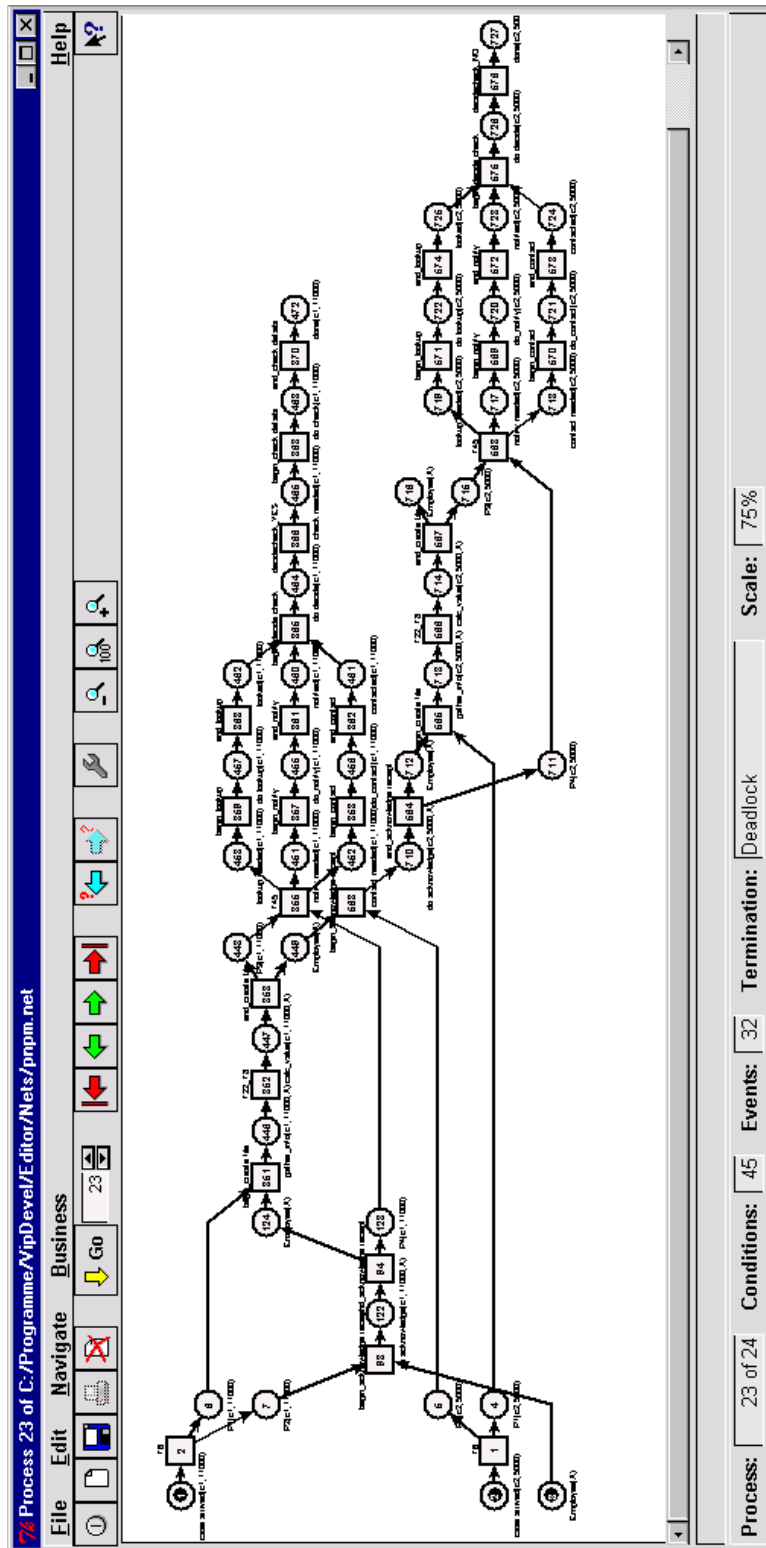


Abbildung 8.9: Linearisierungen durch gemeinsam genutzte Ressourcen.

nebenläufig ausführbare Aktivitäten aufgrund gemeinsam genutzter Ressourcen sequentiell ausgeführt werden. In unserem Beispiel berücksichtigen wir nun die Ressource **Employee**. Um einen Eindruck von dem Verhalten des Geschäftsprozeßmodells unter ungünstigen Umständen zu erhalten, wird angenommen, daß nur ein einzigen Mitarbeiter *A* verfügbar ist.

Durch die kombinatorischen Möglichkeiten der Allokation des Mitarbeiters *A* zu den verschiedenen Aktivitäten erhöht sich die Anzahl der generierten Abläufe von 1 auf 24. Darüber hinaus sind die Abläufe erheblich länger geworden. Abbildung 8.9 zeigt die Effekte des konkurrierenden Ressourcen-Zugriffs. Damit wird deutlich, wie der *VIPbusiness*-Ansatz zu einer detaillierten Analyse des möglichen Verhaltens von Geschäftsprozeßmodellen eingesetzt werden kann, insbesondere beim Einsatz beschränkter Ressourcen. Diese Möglichkeiten sind vor allem für die Identifizierung und Untersuchung von *worst case*-Szenarios von großer Bedeutung.

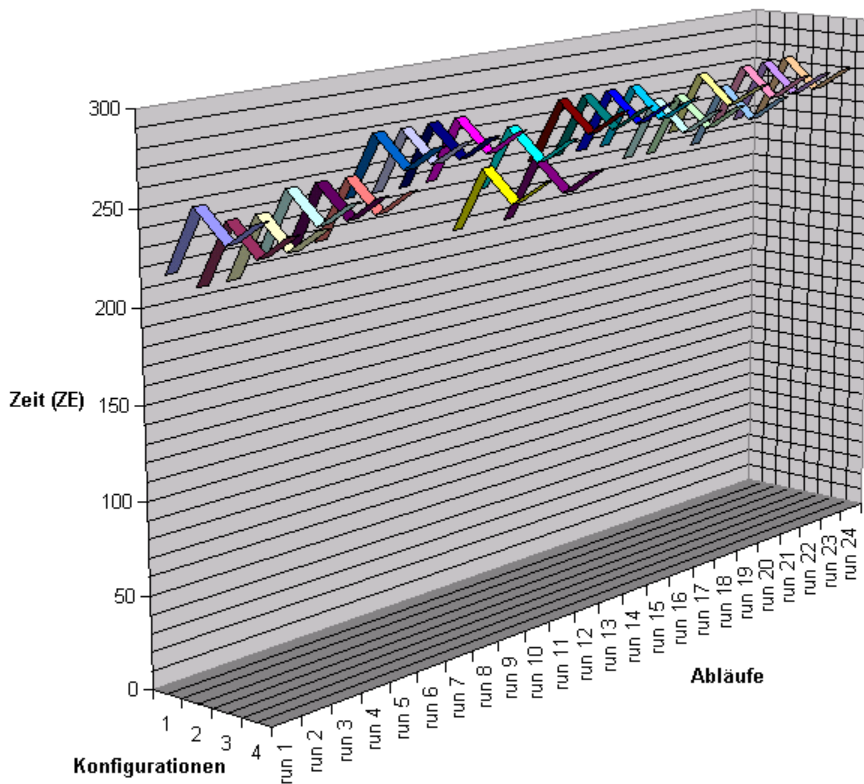


Abbildung 8.10: Durchlaufzeiten für alle Abläufe und verschiedene Konfigurationen.

In Kombination mit dem in Abschnitt 7.5.1 vorgestellten Konzept (multipler) *Konfigurationen* für Aktivitätszeiten können jetzt alle möglichen relevanten Kennzahlen für verschiedene *Szenarien* durch Anwendung der einzelnen in Kapitel 7 beschriebenen Analyse-

verfahren untersucht werden. Abbildung 8.10 zeigt die Durchlaufzeiten für alle 24 Abläufe und vier Mengen von Zufallszahlen, die für eine Konfiguration mit stochastischen Aktivitätszeiten generiert wurden. Alle 24 Abläufe zeigen ähnliche Ergebnisse bezüglich der ermittelten Durchlaufzeiten. Es gibt also offensichtlich keinen „besten“ Ablauf, der eine Vorbildfunktion für die Bearbeitung der Cases haben könnte. Angesichts des eher geringen Variationsgrades unseres kleinen Beispiels ist dieses Ergebnis nicht besonders überraschend.

Trotzdem können auch bei diesem Beispiel eine Reihe von Punkten gefunden werden, die eine genauere Untersuchung verdienen. Da bei der Ermittlung der Durchlaufzeit der Abläufe für Konfiguration 1 durchweg die besten und für Konfiguration 2 konstant die schlechtesten Werte erzielt werden, lohnt es sich, einen genauen Blick auf die Werte zu werfen, die im Rahmen der Erzeugung der Zufallszahlen generiert wurden. In einem zweiten Schritt sollte abgeschätzt werden, welche der erzeugten Werte die realistischeren sein könnten. Beispielsweise für den Fall, daß für die Bearbeitung von Cases eine Durchlaufzeit unter 250 garantiert werden soll, wäre Konfiguration 2 als realistisches Szenario ein Problem. Darüber hinaus kann es sinnvoll sein, den Grund für die geringfügig besseren Durchlaufzeiten einiger Abläufe (beispielsweise die Abläufe 1, 11 und 13) zu ermitteln. Die so ermittelten Durchlaufzeiten bieten eine solide Grundlage zur Festlegung eines Service Levels, beispielsweise in Form einer maximalen Zeitspanne zur Bearbeitung eines Schadensfalls, die den Kunden bei der Bearbeitung echter Cases zugesichert werden kann.





## Kapitel 9

# Zusammenfassung und Ausblick

Mit der zunehmenden Verknüpfung von Geschäftsprozessen und Informationssystemen sind die Ansprüche an die Qualität von Geschäftsprozeß-Entwürfen aufgrund ihrer Schnittstellenfunktion bei der Durchführung entsprechender Integrations-Projekte stark gestiegen. Als Ergebnis früher, anwendungsorientierter Projektphasen bilden Geschäftsprozeß-Entwürfe die Grundlage für den Beginn der Umsetzungs- und Implementierungsaufgaben, die in den späteren, systemorientierten Projektphasen im Vordergrund stehen (vgl. Abbildung 1.1 in Kapitel 1). Dabei haben in den Entwurfsphasen getroffene Fehlentscheidungen bezüglich der Gestaltung der Geschäftsprozesse schwerwiegende Auswirkungen auf die Qualität der implementierten Prozesse. Schlecht funktionierende Systeme und hohe Anpassungskosten sind die Folge. Aus diesem Kontext ergibt sich eine Reihe von Anforderungen an die beim Entwurf von Geschäftsprozessen verwendeten Formalismen, Methoden und Werkzeuge.

### Ergebnisse

In dieser Arbeit wurde mit *VIPbusiness* ein werkzeugunterstützter Ansatz für den Entwurf von qualitativ hochwertigen Geschäftsprozessen vorgestellt. Ausgehend von einem auf den Aspekten *Korrektheit*, *Konsistenz* und *Effizienz* basierenden Qualitätsbegriff wurde ein 3-stufiges Vorgehen entwickelt, das neben der Erstellung von Geschäftsprozeßmodellen deren Simulation sowie Analyse bezüglich aller relevanten Qualitätsmerkmale erlaubt.

Zur *Modellierung* von Geschäftsprozessen wurde mit *VIPbusiness*-Netzen eine neue Klasse höherer Petrinetze entwickelt, die eine Reihe speziell auf den Entwurf von Geschäftsprozessen ausgerichteter Bausteine zur Verfügung stellt. Da sich alle in dieser Arbeit eingeführten Erweiterungen auf herkömmliche Petrinetz-Konzepte zurückführen lassen, verfügen *VIPbusiness*-Netze über eine formale Semantik, die eine *Simulation* der erstellten Geschäftsprozeßmodelle ermöglicht. Das im Rahmen der Simulation in Form von halbgeordneten Abläufen generierte Verhalten des Geschäftsprozeßmodells bildet die Grundlage für eine systematische Analyse des Geschäftsprozeß-Entwurfs bezüglich aller relevanten Qualitätsmerkmale. Da die dazu entwickelten Verfahren aus den Bereichen *Validierung*, *Verifikation* und *Leistungsbewertung* ohne Ausnahme auf die erstellten Geschäftsprozeßmodelle beziehungsweise auf das im Rahmen der Simulation generierte Verhalten angewendet werden können, kann mit dem *VIPbusiness*-Ansatz eine integrierte Analyse aller Aspekte der Qualität eines Geschäftsprozeß-Entwurfs, mit *einem* Modell und *einem* Werkzeug, durchgeführt werden.

Die wenigsten Geschäftsprozesse sind vollständig strukturiert. Daher wurde in dieser Arbeit ein besonderes Augenmerk auf die Unterstützung des Entwurfs von Geschäftsprozessen gerichtet, die neben strukturierten auch nicht-strukturierte Teile enthalten. Durch entsprechende Erweiterungen, insbesondere der Klasse der *VIPbusiness*-Netze, wurde erreicht, daß *hybride Spezifikationen* nicht vollständig strukturierter Geschäftsprozesse erstellt und ohne Einschränkungen gegenüber vollständig strukturierten Geschäftsprozessen simuliert sowie analysiert werden können.

Der Anspruch, alle in dieser Arbeit entwickelten Formalismen und Methoden durch Werkzeuge zu unterstützen, bildete den Ausgangspunkt für eine Untersuchung existierender Werkzeuge bezüglich ihrer Eignung zur Umsetzung des *VIPbusiness*-Ansatzes. Basierend auf den Ergebnissen dieser Untersuchung wurde mit der Implementierung entsprechender *VIPbusiness*-Erweiterungen für das *VIPtool* ein eigener Werkzeug-Prototyp für den Entwurf von Geschäftsprozessen realisiert.

### **Mögliche weitere Schritte**

Ansatzpunkte für eine Weiterentwicklung der in dieser Arbeit entwickelten Konzepte resultieren in erster Linie aus der Anwendung des *VIPbusiness*-Ansatzes für „echte“ Geschäftsprozesse.

Zwar sind die in dieser Arbeit vorgestellten Konzepte für einen Geschäftsprozeß-Entwurf in Experimenten durch Studenten untersucht worden [VBA01], eine breite Erprobung des *VIPbusiness*-Ansatzes in der *Industrie* steht jedoch noch aus. Grundsätzlich problematisch ist dabei die in der Praxis immer noch verbreitete Nicht-Akzeptanz von Werkzeugen zur Modellierung und Analyse von Geschäftsprozessen. Hauptpunkt der Kritik ist hier das als nicht lohnend empfundene Verhältnis von Aufwand und Nutzen des Einsatzes entsprechender Techniken und Werkzeuge. Stand der Dinge ist daher häufig eine ausschließliche Dokumentation von Geschäftsprozessen beispielsweise im Rahmen einer *ISO*-Zertifizierung. Durch eine Verknüpfung des *VIPbusiness*-Ansatzes sowie des *VIPtools* mit den dabei zur Anwendung kommenden Methoden und Werkzeugen sollte es möglich sein, die Bereitschaft möglicher Anwender zu einer Nutzung der in dieser Arbeit vorgestellten Konzepte deutlich zu erhöhen. Im Extremfall kommen die *VIPbusiness*-Methoden dabei für den Anwender völlig transparent zum Einsatz. Die Möglichkeit vorausgesetzt, die jeweils verwendeten Spezifikationen ganz oder in relevanten Teilen in *VIPbusiness*-Netze überführen zu können, läßt sich vermeiden, daß der Anwender zur Nutzung der Analyse-Möglichkeiten des *VIPbusiness*-Ansatzes einen neuen Modellierungs-Formalismus sowie den Umgang mit neuen Werkzeugen erlernen muß. Durch den Mehrwert der Entwurfs-Unterstützung durch die *VIPbusiness*-Konzepte kann so die Akzeptanz für den praktischen Einsatz deutlich erhöht werden.

Beim praktischen Einsatz des *VIPbusiness*-Ansatzes wird sich ein Bedarf an zusätzlichen Analyse-Möglichkeiten ergeben, beispielsweise im Bereich der Leistungsbewertung. Aufgrund des modularen Aufbaus des *VIPtools* können neu entwickelte Verfahren, beispielsweise aus dem Bereich der Kostenrechnung, ohne großen Aufwand in die bereits existierende Umgebung eingebettet werden. Eine Erweiterung der *Editor*-Komponente um zusätzliche Modellierungs-Bausteine kann ebenfalls problemlos realisiert werden.

Weitere interessante Ansatzpunkte ergeben sich aus dem Einsatz der in dieser Arbeit erstmals zur Verfügung gestellten ausführbaren (hybriden) Spezifikationen nicht vollständig strukturierter Geschäftsprozesse als Steuerung für Workflow Management-Systeme oder andere Komponenten in Informationssystemen. Aus den Besonderheiten dieser Spezifikationen und der daraus resultierenden relativen Freiheit des Anwenders im Umgang mit Informationssystemen ergeben sich eine Reihe von Anwendungs-Szenarien (in [DE00] umrissen), deren Erprobung in der Praxis lohnend erscheint.



## Anhang A

# Liste der anfänglich berücksichtigten Werkzeuge für den Entwurf von Geschäftsprozessen

Name	Workflow	Simulation	Engine
AWD and Workflow Analyzer	Ja	Ja	Nein
BONAPART	Ja	Ja	Nein
Business Object Modelling Workbench	Ja	Ja	Nein
EPM Suite	Ja	Ja	Nein
METIS	Ja	Ja	Nein
PAVONE Group Flow	Ja	Ja	Nein
SPARKS G2	Ja	Ja	Nein
Struct Ware	Ja	Ja	Nein
Workflow Analyzer	Ja	Ja	Nein
Workflow BPR	Ja	Ja	Nein
Beyond Mail	Ja	Nein	Ja
COI-Business Flow	Ja	Nein	Ja
COSA	Ja	Nein	Ja
Flowmark	Ja	Nein	Ja
FYI	Ja	Nein	Ja
IBIsys	Ja	Nein	Ja
Linkworks	Ja	Nein	Ja
OPENworkflow	Ja	Nein	Ja
Plexus Flo Ware	Ja	Nein	Ja
Viewstar	Ja	Nein	Ja
1View Workflow	Ja	Nein	Nein
ABSI-Docs	Ja	Nein	Nein
Action Request System	Ja	Nein	Nein
Action Workflow Analyst	Ja	Nein	Nein
Action Workflow Application Builder	Ja	Nein	Nein
Action Workflow Enterprise Series	Ja	Nein	Nein

Name	Workflow	Simulation	Engine
Action Workflow Workflow Manager	Ja	Nein	Nein
ARIS	Ja	Nein	Nein
ATI Workflow Manager	Ja	Nein	Nein
Automated Work Distributor AWD	Ja	Nein	Nein
Cap Web-Flow	Ja	Nein	Nein
CMS Workflow	Ja	Nein	Nein
Computron Workflow	Ja	Nein	Nein
COOL	Ja	Nein	Nein
CSEWorkflow 5.0	Ja	Nein	Nein
Designer2000	Ja	Nein	Nein
Docu Flow	Ja	Nein	Nein
Document Manager	Ja	Nein	Nein
Documentrix Workmanager	Ja	Nein	Nein
EDI36	Ja	Nein	Nein
EDI38	Ja	Nein	Nein
EDI400	Ja	Nein	Nein
EDLe QMail	Ja	Nein	Nein
Engineering Workflow System	Ja	Nein	Nein
Ensemble	Ja	Nein	Nein
Enterprise Analyst	Ja	Nein	Nein
Entire Workflow	Ja	Nein	Nein
Extend BPR	Ja	Nein	Nein
Fabasoft Components	Ja	Nein	Nein
File Net Work Flow	Ja	Nein	Nein
Flo Ware	Ja	Nein	Nein
Flow Maker	Ja	Nein	Nein
Flow Man	Ja	Nein	Nein
Flow Works	Ja	Nein	Nein
FLOWBuilder	Ja	Nein	Nein
Form Flow	Ja	Nein	Nein
FORO	Ja	Nein	Nein
FYI Workflow	Ja	Nein	Nein
Group Wise	Ja	Nein	Nein
IBMBusiness Process Modeler	Ja	Nein	Nein
IBS Workflow Manager	Ja	Nein	Nein
Image Fast	Ja	Nein	Nein
Image Master	Ja	Nein	Nein
In Concert	Ja	Nein	Nein
Inter Office	Ja	Nein	Nein
Jet Form Server	Ja	Nein	Nein
Key Workgroup	Ja	Nein	Nein
Keyflow	Ja	Nein	Nein
Life FLOW	Ja	Nein	Nein
Link Works Team Links	Ja	Nein	Nein
Livelihood Intranet	Ja	Nein	Nein
MAVIM 3	Ja	Nein	Nein
Memo	Ja	Nein	Nein

<b>Name</b>	<b>Workflow</b>	<b>Simulation</b>	<b>Engine</b>
Message Driven processor MDP	Ja	Nein	Nein
Metaphase 2.0	Ja	Nein	Nein
Metaview FOLDERS	Ja	Nein	Nein
METEOR	Ja	Nein	Nein
Navigator 2000 DMS Navigator	Ja	Nein	Nein
Navigator 2000 Workflow	Ja	Nein	Nein
Nova Manage	Ja	Nein	Nein
OCTOFlow	Ja	Nein	Nein
ODMS	Ja	Nein	Nein
Office IQ	Ja	Nein	Nein
Open Image	Ja	Nein	Nein
Optix Workflow	Ja	Nein	Nein
Power Flow	Ja	Nein	Nein
Power Flow Team Flow Process Wise	Ja	Nein	Nein
Power Work	Ja	Nein	Nein
Process Flo	Ja	Nein	Nein
Process IT	Ja	Nein	Nein
Protos	Ja	Nein	Nein
radica	Ja	Nein	Nein
Regatta	Ja	Nein	Nein
Route Builder Omni Desk	Ja	Nein	Nein
SAP Business Workflow	Ja	Nein	Nein
Smart Flow 98	Ja	Nein	Nein
Smart Stream	Ja	Nein	Nein
The Vantive System	Ja	Nein	Nein
Ultimus	Ja	Nein	Nein
Viewstar Workbench	Ja	Nein	Nein
Win Work	Ja	Nein	Nein
Wizdom Works	Ja	Nein	Nein
Work Fast	Ja	Nein	Nein
Work Flow Analyzer	Ja	Nein	Nein
Work MAN	Ja	Nein	Nein
Work Party	Ja	Nein	Nein
Work Xpert	Ja	Nein	Nein
WORKlogik TM	Ja	Nein	Nein
World Wide Web Flow W4	Ja	Nein	Nein
XWorkflow	Ja	Nein	Nein
BPSimulator Template	Nein	Ja	Nein
Business Process Analyzer	Nein	Ja	Nein
Bwise Toolkit	Nein	Ja	Nein
CABRE -Witness	Nein	Ja	Nein
Cinderella SDL	Nein	Ja	Nein
CLEAR	Nein	Ja	Nein
Clear Process	Nein	Ja	Nein
Design CPN	Nein	Ja	Nein
Design Leverage	Nein	Ja	Nein
Dress Rehearsal	Nein	Ja	Nein

Name	Workflow	Simulation	Engine
First STEP	Nein	Ja	Nein
Flowcharter	Nein	Ja	Nein
GRADE	Nein	Ja	Nein
HITSoft BIZ	Nein	Ja	Nein
HOCUS	Nein	Ja	Nein
Ithink	Nein	Ja	Nein
i-think	Nein	Ja	Nein
Live Analyst	Nein	Ja	Nein
Micro SAINT	Nein	Ja	Nein
Object GEODE	Nein	Ja	Nein
Optima	Nein	Ja	Nein
Optima Express	Nein	Ja	Nein
Oracle Process Manager	Nein	Ja	Nein
PACE	Nein	Ja	Nein
Powersim	Nein	Ja	Nein
Pro Model	Nein	Ja	Nein
Pro Model 2.0	Nein	Ja	Nein
Process Charter	Nein	Ja	Nein
Prophesy	Nein	Ja	Nein
PROSIM Process Modeling Software	Nein	Ja	Nein
Quick CRC	Nein	Ja	Nein
RDD-100	Nein	Ja	Nein
SES/Workbench	Nein	Ja	Nein
SIMAN amp ARENA	Nein	Ja	Nein
SIMPROCESS	Nein	Ja	Nein
Soft Modeler Business	Nein	Ja	Nein
Statemate Magnum	Nein	Ja	Nein
Surveywin	Nein	Ja	Nein
Taylor II	Nein	Ja	Nein
TI BDF	Nein	Ja	Nein
Vectus	Nein	Ja	Nein
Vensim	Nein	Ja	Nein
Witness	Nein	Ja	Nein
ALL-IN-1/Team Route/Team Links	Nein	Nein	Ja
ArChief	Nein	Nein	Ja
AT&T ProcessIT	Nein	Nein	Ja
CMS/Workflow	Nein	Nein	Ja
DMS	Nein	Nein	Ja
DOX	Nein	Nein	Ja
Ecals/Ieals	Nein	Nein	Ja
FlowPATH	Nein	Nein	Ja
GroupFlow	Nein	Nein	Ja
Infomation Broker	Nein	Nein	Ja
INVOS	Nein	Nein	Ja
Keyfile	Nein	Nein	Ja
Leonberg VGS	Nein	Nein	Ja
LEU	Nein	Nein	Ja



<b>Name</b>	<b>Workflow</b>	<b>Simulation</b>	<b>Engine</b>
MIRES	Nein	Nein	Ja
multiDESK	Nein	Nein	Ja
multiDESK_Access	Nein	Nein	Ja
Plexus Flo Ware Desk	Nein	Nein	Ja
PRO/PHI	Nein	Nein	Ja
ProMinanD	Nein	Nein	Ja
Serie M	Nein	Nein	Ja
STAFFWARE	Nein	Nein	Ja
TC-Flow	Nein	Nein	Ja
TeamOFFICE	Nein	Nein	Ja
WorkFlo Business System	Nein	Nein	Ja
WORKFLOW	Nein	Nein	Ja
WORKFLOW SERVER	Nein	Nein	Ja
WorkParty	Nein	Nein	Ja



## Anhang B

# Kurzbeschreibung der untersuchten Werkzeuge für den Geschäftsprozeß-Entwurf

### **all Clear**

Firma: SPSS Inc.  
233 S. Wacker Drive, 11th Floor  
Chicago, Il 60606-6307  
U.S.A  
Tel. +1-312-651-3000  
Fax. +1-312-651-3668  
Homepage: <http://www.spss.com>

### *Besonderheiten:*

- Es ist möglich, wahlweise einen Ablauf oder ein Organisationsdiagramm zu modellieren und zu simulieren.
- Die Modelle können mittels Drag&Drop oder einer Skriptsprache erstellt werden. Ein zweigeteilter Arbeitsbereich ermöglicht es, mit beiden Methoden gleichzeitig und ergänzend zu arbeiten.
- Es können alle möglichen Wege sowie der kritische und der optimale Weg in einem Modell bestimmt werden.
- Graphische Symbole im Modell können ausgetauscht werden. Es können eigene Symbole erstellt werden.
- ODER-Verzweigungen werden durch Wahrscheinlichkeiten realisiert.

### **Aris Toolset**

Firma: IDS Scheer AG  
Altenkesseler Straße 17  
66115 Saarbrücken  
Deutschland  
Tel.: +49-681-210-0  
Fax.: +49-681-210-1000  
Homepage: <http://www.ids-scheer.de>

#### *Besonderheiten:*

- Aktivitäten können durch verschiedene graphische Symbole dargestellt werden.
- ODER-Verzweigungen werden mittels prozentualer Wahrscheinlichkeiten realisiert.
- Bei der Animation der Bearbeitung von Cases ist ein *halbautomatischer Modus* möglich. In diesem Modus wird der Benutzer lediglich nach dem zu wählenden Pfad bei Verzweigungen gefragt.
- Als nicht-quantitative Kennzahlen können die Anzahl der Medien- sowie die Anzahl der Organisationsbrüche angegeben werden.
- Es besteht die Möglichkeit, Prozesse als *Business Cases* zu modellieren und vom Programm automatisch in *ereignisgesteuerte Prozeßketten (EPKs)* umwandeln zu lassen.
- Eine Evaluation der Analysemöglichkeiten des Programms konnte nicht stattfinden, da bei der gelieferten Demoversion die Analyse nicht ausführbar war. Die Angaben in diesem Report stützen sich daher auf [IDS00]. Demnach scheint die Simulation einen Kalender zu benutzen. Es kann die Verarbeitung mehrerer Cases simuliert werden, für jeden Case wird dabei offenbar eine neue Instanz des Geschäftsprozeßmodells erzeugt.

### **Bonapart**

Firma: Proubis GmbH  
Alt-Moabit 96  
16559 Berlin  
Deutschland  
Tel.: +49-30-39929880  
Fax.: +49-30-39929888  
Homepage: <http://www.proubis.de>

#### *Besonderheiten:*

- Die Prozessmodelle lassen sich beliebig verfeinern.
- ODER-Verzweigungen können mittels prozentualer Wahrscheinlichkeiten oder in Abhängigkeit von Informationen realisiert werden.

- Die Werte für Aktivitätszeiten können mit Zufallswerten belegt werden. Es stehen verschiedene Wahrscheinlichkeitsverteilungen zur Verfügung.
- Für Ressourcen können einmalige und/oder periodische *Sperrzeiten*<sup>1</sup> modelliert werden.
- Für die Bearbeitung von Cases stehen verschiedene Strategien zur Verfügung: *LIFO*, *FIFO*, *kürzester Auftrag zuerst*, *längster Auftrag zuerst*.
- Benutzerdefinierte Analysereporte können erstellt werden.
- Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Aktivitäten können durch verschiedene graphische Symbole dargestellt werden.
- Wartende Cases und Cases in Bearbeitung können während der Simulation für einzelne Aktivitäten anhand von Diagrammen dynamisch verfolgt werden.

### **Corporate Modeler 99**

Firma: Casewise Ltd.

9-13 Swiss Terrace

London NW6 4RR

U.K.

Tel.: +44-207-722-4000

Fax.: +44-207-722-4004

Homepage: <http://www.casewise.com>

#### *Besonderheiten:*

- Die Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Die Ankunftsrate für Cases an der Quelle kann mittels verschiedener Verteilungsfunktionen realisiert werden.
- Alle quantitativen Parameter können mit Zufallswerten gemäß verschiedener Wahrscheinlichkeitsverteilungen belegt werden.
- ODER-Verzweigungen werden über prozentuale Wahrscheinlichkeiten realisiert.
- Diagramme können frei definiert und während der Simulation verfolgt werden.
- Das Erstellen von benutzerdefinierten Berichten (Reports) ist möglich.
- Die Verknüpfung von Geschäftsprozeß und Organisation kann über die Einblendung von *Swimlanes*<sup>2</sup> visualisiert werden.
- Prozeßmodelle lassen sich durch Subprozesse verfeinern.

---

<sup>1</sup>Sperrzeiten sind vor allem bei menschlichen Ressourcen von Bedeutung. Die entsprechende Ressource steht zu diesen Zeiten nicht zur Verfügung. Sperrzeiten werden beispielsweise zur Modellierung von Wochenenden, Schichtdauern oder Urlaubszeiten verwendet.

<sup>2</sup>Dabei wird der Verlauf des Geschäftsprozesses im Organisations-Modell des Unternehmens dargestellt. Auf diese Weise lassen sich beispielsweise Organisationsbrüche einfach identifizieren.

- Während der Simulation können die Längen der Warteschlangen an Aktivitäten angezeigt werden.
- Es existieren mehrere Module zur Erstellung von Modellen mit Hilfe verschiedener Formalismen (Datenflußdiagramm, Entity Relationship Diagramm, ...), diese sind jedoch nicht simulierbar.

### **Extend + BPR**

Firma: Imagine That Incorporated

6830 Via Del Oro, Suite 230

San Jose, CA 95119

U.S.A.

Tel.: +1-408-365-0305

Fax.: +1-408-629-1251

Homepage: <http://www.imaginethatinc.com>

#### *Besonderheiten:*

- Die Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- ODER-Verzweigungen werden durch *If-Anweisungen* (abgeleitet von der Programmiersprache C) realisiert.
- Die Modellierung erfolgt mit Hilfe verschiedener Blöcke mit vordefinierten Eingangs-, Ausgangs-, Steuer- und Ausgabeschnittstellen.
- Es besteht die Möglichkeit, eigene Blöcke mit Hilfe der Programmiersprache *Modl* zu erstellen. Dadurch ist es möglich, sehr unterschiedliche Aktivitäten zu implementieren.
- Ein *Ressourcenblock* ermöglicht die Integration verschiedener *Mitarbeiterpools* in das Modell. Auf diese Weise kann die Organisations-Dimension bedingt in das Geschäftsprozeßmodell integriert werden.
- Mit Hilfe von *Plotterblöcken*, die an beliebige andere Blöcke angehängt werden können, lassen sich die Werte von Variablen an jeder Stelle des Modells überwachen.
- Während der Simulation können Warteschlangenlänge und Füllstände dynamisch visualisiert werden.
- Die Quelle wird als *Generatorblock* bezeichnet. Mittels verschiedener Wahrscheinlichkeitsfunktionen können Cases erzeugt werden (bzw. Ereignisse, die der Ankunft eines Case entsprechen).
- Es können *Kontrollschalter* eingebaut werden, um die *Durchlässigkeit* von Verbindungen und somit das Systemverhalten zu beeinflussen.
- Warteschlangen für Aktivitäten können mittels vordefinierter Blöcke modelliert werden. Eine Warteschlange läßt sich innerhalb eines solchen Blocks mit verschiedenen Bearbeitungsregeln (*Prioritäten, FIFO, LIFO, ...*) verknüpfen.

### **First Step Designer 4**

Firma: Interfacing Technologies Corporation  
3333 Graham Blvd., Suite 101  
T.M.R ( Montreal ), Quebec, Canada H3 R1K1  
Canada  
Tel.: +1-514-737-7333  
Fax.: +1-514-737-0856  
Homepage: <http://www.interfacing.com>

#### *Besonderheiten:*

- Die Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Eine Animation während der Simulation findet nur im Organisationsdiagramm statt.
- Prozesse können durch Subprozesse verfeinert werden.
- Die ODER-Verzweigungen werden mittels prozentualer Wahrscheinlichkeiten realisiert.
- Die Dauer einer Aktivität kann mittels Zufallswerten gemäß einer Wahrscheinlichkeitsverteilung belegt werden.
- Während der Simulation können Prozeßdaten dynamisch in einem Diagramm verfolgt werden.
- Ergebnisreporte können benutzerdefiniert erstellt werden. Für die Darstellung von Ergebnissen sind verschiedene Diagrammartent wählbar (z.B. Tabelle, Balkendiagramm, Kuchendiagramm, ...).
- Für das Geschäftsprozeßmodell kann der kürzeste (der längste) sowie der günstigste (der teuerste) Weg bestimmt werden.
- Die Simulation erlaubt die Verwendung eines Kalenders, mit dessen Hilfe die Verfügbarkeitszeiten und Sperrzeiten von Ressourcen angegeben werden können.

### **Grade**

Firma: Infologistik GmbH  
Wagnerbreite 3  
D-83607 Holzkirchen  
Deutschland  
Tel.: +49-8024-3045-0  
Fax.: +49-8024-3045-25  
Homepage: <http://www.infologistik.com>

#### *Besonderheiten:*

- Während der Simulation wird die Länge der Warteschlangen für Geschäftsprozeß-Objekte angezeigt.

- Es können mehrere Geschäftsprozeßmodelle parallel simuliert werden.
- ODER-Verzweigungen können mittels Wahrscheinlichkeiten oder einer Formel realisiert werden.
- Prozesse können in mehrere Subprozesse unterteilt werden.
- Zur Modellierung von Daten können Entity-Relationship-Modelle erstellt, jedoch nicht simuliert und analysiert werden. Aus den ER-Modellen können automatisch Klassendiagramme generiert werden.
- Der kritische Weg eines Geschäftsprozeßmodells ist bestimmbar.
- Es existieren verschiedene Ansichten für das Geschäftsprozeßmodell. Dabei kann die Verknüpfung von Geschäftsprozeß und Organisation über die Einblendung von *Swimlanes* visualisiert werden.
- Präsentationen können automatisch erstellt werden.

### **Income**

Firma: Promatis AG

Badhausweg 5

76307 Karlsbad

Deutschland

Tel.: +49-7248-926-0

Fax.: +49-7248-926-119

Homepage: <http://www.promatis.de>

### *Besonderheiten:*

- Aktivitäten können mit graphischen Symbolen belegt werden.
- Income setzt auf einer Datenbank der Firma *Oracle* auf.
- Es können Kapazitätsgrenzen angegeben werden.
- ODER-Verzweigungen werden mittels prozentualer Wahrscheinlichkeiten realisiert.
- Für die Bearbeitung von Cases stehen verschiedene Strategien zur Auswahl (*FIFO*, *LIFO*, ...).
- Die Verfeinerung von Geschäftsprozeßmodellen wird unterstützt.
- Berichte (Reports) lassen sich benutzergeführt erstellen. Für die Präsentation der Analyse-Ergebnisse stehen verschiedene Diagrammartentypen zur Verfügung (Balkendiagramme, Tortendiagramme,...).
- Subprozesse können separat oder gleichzeitig simuliert werden.



### **Ivy Frame**

Firma: Ivy Team

Postfach

Alpenstrasse 9

CH-6304 Zug

Schweiz

Tel.: +41-41-710-80-20

Fax.: +41-41-710-80-60

Homepage: <http://www.ivyteam.com>

#### *Besonderheiten:*

- Prozesse können durch Subprozesse verfeinert werden.
- Aktivitäten können mit graphischen Symbolen belegt werden.
- Entscheidungen können mittels Wahrscheinlichkeiten oder Bedingungen formuliert werden.
- Durch eine sogenannte *E/A Sonde* ist es möglich, Werte während einer Simulation als Grafiken zu verfolgen.
- Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Während der Simulation kann die Auslastung der Ressourcen anhand von Füllständen verfolgt werden. Dabei werden Ressourcen als Objektspeicher modelliert.
- Während der Simulation wird die Länge der Warteschlange vor jeder Aktivität angezeigt.

### **Optima!**

Firma: Micrografx Incorporated

505 Millenium Drive

Allen, TX 75013

U.S.A.

Tel.: +1-214-495-4000

Fax.: +1-214-495-4073

Homepage: <http://www.micrografx.com>

#### *Besonderheiten:*

- ODER-Verknüpfungen können mittels Wahrscheinlichkeiten oder Bedingungen formuliert werden.
- Die Verknüpfung von Geschäftsprozeß und Organisation kann über die Einblendung von *Swimlanes* visualisiert werden. Dabei können den jeweiligen Organisationselementen Ressourcen zugeteilt werden. Die Auslastung der Ressourcen während der Simulation wird angezeigt.

- Für verschiedene Ressourcen lassen sich verschiedene Arbeitszeiten und Schichten definieren.

### **Scitor Process V3**

Firma: Scitor Corporation  
 Business Solution Group  
 256 Gibraltar Drive  
 Sunnyvale, CA 94085  
 U.S.A.  
 Tel. +1-408-745-8300  
 Fax. +1-408-745-8301  
 Homepage: <http://www.scitor.com>

#### *Besonderheiten:*

- Bei Modellierung können verschiedene Ebenen definiert und Prozeßmodelle auf diese Weise verfeinert werden.
- Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Funktionen können mit graphischen Symbolen belegt werden.
- Für Ressourcen können Kosten- und Zeitbeschränkungen festgelegt werden. Ressourcen werden den Funktionen zugeordnet.
- Ergebnisdaten können mittels Spreadsheets oder Tabellen dargestellt werden.
- ODER-Verknüpfungen können wahlweise mittels Wahrscheinlichkeiten, Bedingungen, längster oder kürzester Warteschlange realisiert werden.

### **Simprocess**

Firma: CACI Products Company  
 14151 Park Meadow Drive  
 Chantilly, VA 20151  
 U.S.A.  
 Tel.: +1-877-663-3746  
 Fax.: +1-703-679-3354  
 Homepage: <http://www.caciasl.com>

#### *Besonderheiten:*

- Geschäftsprozeßmodelle müssen Quellen und Senken enthalten.
- Mehrere Geschäftsprozesse können über *Synchronisationsbausteine* synchronisiert werden. Ein Synchronisationsbaustein hält ein Element solange auf, bis ein weiteres Element an einem zweiten Eingang die Sperre löst.
- Prozesse können auf verschiedenen Ebenen definiert werden. Dadurch wird Verfeinerung unterstützt.

- Eine ODER-Verknüpfung kann mittels Wahrscheinlichkeiten, Attributen, Entity-Typen und Prioritäten realisiert werden.
- Die Dauern von Funktionen können mit Zufallswerten und verschiedenen Wahrscheinlichkeitsverteilungen (z.B. Gleichverteilung, Normalverteilung, Dreiecksverteilung usw.) belegt werden.
- Den Aktivitäten des Ablaufs können unterschiedliche graphische Symbole zugewiesen werden.
- Ressourcen können definiert und verschiedenen Aktivitäten zugewiesen werden.
- Es ist möglich, benutzerdefinierte Berichte (Reports) zu erstellen.
- Die Ankunft von Cases an der Quelle kann gemäß verschiedener Wahrscheinlichkeitsverteilungen erfolgen.
- Werte von Variablen können während der Simulation mit Hilfe von Diagrammen überwacht werden.

### **Workflow BPR**

Firma: Holosofx Incorporated  
 100 N. Sepulveda, Suite 603, El  
 Segundo, CA 90245  
 U.S.A.  
 Tel.: +01-310-640-0101  
 Fax.: +01-310-640-0960  
 Homepage: <http://www.holosofx.com>

#### *Besonderheiten:*

- Für einen Geschäftsprozeß können mehrere Modelle erstellt und vergleichend analysiert werden.
- Alle möglichen Wege durch ein Geschäftsprozeßmodell können getrennt dargestellt und analysiert werden.
- Es wird zwischen externen und internen (in Bezug auf das Unternehmen) Aktivitäten unterschieden.
- Es steht eine Vielzahl vordefinierter Berichte (Reports) zur Auswahl, um das Systemverhalten zu analysieren. Darüber hinaus lassen sich benutzerdefinierte Berichte (Reports) erstellen.
- ODER-Verknüpfungen werden mittels prozentualer Wahrscheinlichkeiten realisiert.
- Die Warteschlange vor jeder Aktivität kann während der Simulation dargestellt werden.
- Die Verfügbarkeitszeiten und Sperrzeiten von Ressourcen können mit Hilfe eines Kalenders angegeben werden.



# Literaturverzeichnis

- [AAH98] N.R. Adam, V. Atluri und W. Huang. Modeling and Analysis of Workflows using Petri Nets. *Journal of Intelligent Information Systems*, 10:131–158, 1998.
- [Aal95] W.M.P. van der Aalst. Petri net based scheduling. Computing Science Reports 95/23, Eindhoven University of Technology, Eindhoven, 1995.
- [Aal97] W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azema und G. Balbo, Hrsg., *Application and Theory of Petri Nets 1997*, Vol. 1248 of *Lecture Notes in Computer Science*, Seiten 407–426. Springer, Berlin, Heidelberg, 1997.
- [Aal98a] W.M.P. van der Aalst. Chapter 10: Three Good reasons for Using a Petri-net-based Workflow Management System. In T. Wakayama et al., Hrsg., *Information and Process Integration in Enterprises: Rethinking documents*, The Kluwer International Series in Engineering and Computer Science, Seiten 161–182. Kluwer Academic Publishers, Norwell, 1998.
- [Aal98b] W.M.P. van der Aalst. Formalization and Verification of Event-driven Process Chains. Computing Science Reports 98/01, Eindhoven University of Technology, Eindhoven, 1998.
- [Aal98c] W.M.P. van der Aalst. Reengineering Knock-out Processes. Forschungsbericht Nr. 381, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1998.
- [Aal98d] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
- [Aal99] W.M.P. van der Aalst. Woflan: A Petri-net-based Workflow Analyzer. *Systems Analysis - Modelling - Simulation*, 35(3):345–357, 1999.
- [AB97] W.M.P. van der Aalst und T. Basten. Life-cycle Inheritance: A Petri-net-based approach. In P. Azema und G. Balbo, Hrsg., *Application and Theory of Petri Nets 1997*, Vol. 1248 of *Lecture Notes in Computer Science*, Seiten 62–81. Springer, Berlin, Heidelberg, 1997.
- [ABC<sup>+</sup>95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli und G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley series in parallel computing. Wiley, New York, 1995.
- [AE01] W.M.P. van der Aalst und T. Erwin. Resource Planning for Business Processes. Forschungsbericht Nr. 405, Universität Karlsruhe, Institut AIFB, Karlsruhe, 2001.
- [AH95] W.M.P. van der Aalst und K.M. van Hee. Framework for Business Process Redesign. In J.R. Callahan, Hrsg., *Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 95)*, Seiten 36–45, Berkeley Springs, April 1995. IEEE Computer Society Press.
- [AHH94] W.M.P. van der Aalst, K.M. van Hee und G.J. Houben. Modelling workflow management systems with high-level Petri nets. In G. De Michelis, C. Ellis und G. Memmi, Hrsg., *Proceedings of the second Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms*, Seiten 31–50, 1994.

- [AHKB00a] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski und A.P. Barros. Advanced Workflow Patterns. In O. Etzion und P. Scheuermann, Hrsg., *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, Vol. 1901 of *Lecture Notes in Computer Science*, Seiten 18–29. Springer, Berlin, Heidelberg, 2000.
- [AHKB00b] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski und A.P. Barros. Workflow Patterns. BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000.
- [AHV97] W.M.P. van der Aalst, D. Hauschildt und H.M.W. Verbeek. A Petri-net-based Tool to Analyze Workflows. In B. Farwer, D. Moldt und M.O. Stehr, Hrsg., *Proceedings of Petri Nets in System Engineering (PNSE'97)*, Seiten 78–90, Universität Hamburg, Hamburg (FBI-HH-B-205/97), 1997.
- [Ajm95] M. Ajmone Marsan. An introduction to generalized stochastic Petri nets. In *Second International Course on Petri Nets for Latin America*, 1995.
- [AKD98] W. M. P. van der Aalst, E. Kindler und J. Desel. Beyond Asymmetric Choice: A note on some extensions. *Petri Net Newsletter*, 55:3–13, 1998.
- [AM98] A. Agostini und G. De Michelis. Simple Workflow Models. In W.M.P. van der Aalst, G. De Michelis und C.A. Ellis, Hrsg., *Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98)*, Vol. 98/7 of *Computing Science Reports*, Seiten 146–164. Eindhoven University of Technology, Eindhoven, 1998.
- [AME98] W.M.P. van der Aalst, G. De Michelis und C.A. Ellis, Hrsg. *Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98)*, Vol. 98/7 of *Computing Science Reports*. Eindhoven University of Technology, Eindhoven, June 1998.
- [AW91] W.M.P. van der Aalst und A.W. Waltmans. Modelling logistic systems with EXPECT. In H.G. Sol und K.M. van Hee, Hrsg., *Dynamic Modelling of Information Systems*, Seiten 269–288. Elsevier Science Publishers, Amsterdam, 1991.
- [BA01] T. Basten und W.M.P. van der Aalst. Inheritance of Dynamic Behaviour: Development of a Groupware Editor. In G. Agha, F. De Cindo und G. Rozenberg, Hrsg., *Concurrent Object-Oriented Programming and Petri Nets*, Vol. 2001 of *Lecture Notes in Computer Science*, Seiten 391–405. Springer, Berlin, Heidelberg, 2001.
- [Bak98] Bakkenist Management Consultants. *ExSpect 6.2 User Manual*, 1998.
- [Bau96] B. Baumgarten. *Petri-Netze: Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1996.
- [Bau97] M. Bauhammer. Simulationsgestützte Optimierung von Geschäftsprozessen. Diplomarbeit, Universität Karlsruhe, 1997.
- [BCMP75] F. Baskett, K.M. Chandy, R.R. Muntz und F.G. Palacios. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. *Journal of the Association of Computing Machinery*, 22(2):248–260, April 1975.
- [Bec99] H. Becker. *Produktivitätssteigerungen durch Workflow-Management*. Reihe: Wirtschaftsinformatik. Eul, Bergisch Gladbach, Köln, 1999.
- [Bes86] U. Bestmann, Hrsg. *Kompendium der Betriebswirtschaftslehre*. Oldenbourg, München, Wien, 1986.
- [BH96] C. Berkau und P. Hirschmann, Hrsg. *Kostenorientiertes Geschäftsprozessmanagement: Methoden, Werkzeuge, Erfahrungen*. Vahlen, München, 1996.
- [Bre97] C. Brennecke. *Leu Smart Handbuch*, Release 2.0. Technischer Bericht, Vebacom Service GmbH, Bochum, 1997.

- [BRR87] W. Brauer, W. Reisig und G. Rozenberg, Hrsg. *Advances in Petri Nets 1986, Part II: Applications and Relationships to Other Models of Concurrency*, Vol. 254 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 1987.
- [BSG99] M. Brandt, R. Striemer und V. Gruhn. Ein Ansatz zur zielorientierten Simulation von Geschäftsprozeßmodellen. In A. Oberweis und H.M. Sneed, Hrsg., *Software Management 1999*, Reihe Wirtschaftsinformatik. Teubner, Stuttgart, Leipzig, 1999.
- [Buz96] J.A. Buzacott. Commonalities in Reengineered Business Processes: Models and Issues. *Management Science*, 42(5):768–782, 1996.
- [BY86] J.A. Buzacott und D.D. Yao. On Queueing Networks of Flexible Manufacturing Systems. *Queueing Systems*, 1:29–66, 1986.
- [Cha96] J. Charton-Brockmann. Prozeßkostensimulation mit dem objektorientierten Werkzeug SPARKS (TM). In C. Berkau und P. Hirschmann, Hrsg., *Kostenorientiertes Geschäftsprozessmanagement: Methoden, Werkzeuge, Erfahrungen*, Seiten 275–291, Vahlen, München, 1996.
- [Con01] The Python Consortium. *The Python Homepage*. WWW, <http://www.python.org>, 2001.
- [DA94] A. Desrochers und R. Al-Jaar. *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. IEEE Press, Piscataway, NJ, USA, 1994.
- [Dav93] T.H. Davenport. *Process innovation: reengineering work through information technology*. Harvard Business School Press, Boston, 1993.
- [DE95] J. Desel und J. Esparza. *Free Choice Petri Nets*, Vol. 40 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, Cambridge, 1995.
- [DE00] J. Desel und T. Erwin. Hybrid Specifications - Looking at Workflows from a Run-Time Perspective. *International Journal of Computer Systems Science & Engineering*, 15(5):291–302, 2000.
- [DE01] J. Desel und T. Erwin. Quantitative Engineering of Business Processes. Forschungsbericht Nr. 407, Universität Karlsruhe, Institut AIFB, Karlsruhe, 2001.
- [Dei00] W. Deiters. Information Gathering and Process Modeling in a Petri Net Based Approach. In W.M.P. van der Aalst, J. Desel und A. Oberweis, Hrsg., *Business Process Management: Models, Techniques, and Empirical Studies*, Vol. 1806 of *Lecture Notes in Computer Science*, Seiten 110–128. Springer, Berlin, Heidelberg, 2000.
- [Des92] J. Desel. *Struktur und Analyse von Free-Choice-Petrinetzen*. Deutscher Universitäts-Verlag, Wiesbaden, 1992.
- [Des96] J. Desel. Über den Beweis von Zielen mit linear-algebraischen Techniken. In J. Desel, A. Oberweis und E. Kindler, Hrsg., *3. Workshop Algorithmen und Werkzeuge für Petrinetze*, Seiten 8–13, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1996.
- [Des98] J. Desel. *Petrinetze, lineare Algebra und lineare Programmierung*. Teubner-Texte zur Informatik; Bd. 26. Teubner, Stuttgart, Leipzig, 1998.
- [Des00] J. Desel. Validation of Process Models by Construction of Process Nets. In W.M.P. van der Aalst, J. Desel und A. Oberweis, Hrsg., *Business Process Management: Models, Techniques, and Empirical Studies*, Vol. 1806 of *Lecture Notes in Computer Science*, Seiten 110–128. Springer, Berlin, Heidelberg, 2000.
- [DF96] W. Dangelmaier und W. Felser. Fertigungsprozesse und Fertigungssteuerung. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 225–242. International Thomson Publishing, Bonn, Albany, 1996.

- [DFO97] J. Desel, T. Freytag und A. Oberweis. Prozesse, Simulation und Eigenschaften netzmodellierter Systeme. In *Entwurf komplexer Automatisierungssysteme (EKA 97)*. Universität Braunschweig, 1997.
- [DG90] W. Deiters und V. Gruhn. Managing software processes in Melmac. Proceedings of the 4th ACM SIGSOFT Symposium on Software Development Environments, Irvine, USA. *ACM Sigsoft*, Seiten 193–205, 1990.
- [DGK95] W. Deiters, V. Gruhn und M. Kampmann. Der FUNSOFT-Ansatz zum integrierten Geschäftsprozeßmanagement. *Wirtschaftsinformatik*, 37:459–466, 1995.
- [DHP<sup>+</sup>93] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva und F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, London, 1993.
- [Dij93] N.M. van Dijk. *Queueing Networks and Product Forms: A Systems Approach*. John Wiley and Sons, Chichester, 1993.
- [DIN96] DIN, Hrsg. Geschäftsprozeßmodellierung und Workflow-Management. Forschungs- und Entwicklungsbedarf im Rahmen der Entwicklungsbegleitenden Normung (EBN). DIN Fachbericht 50, o.O., 1996.
- [DK96] J. Desel und E. Kindler. Der Traum von einem universellen Petrinetz-Werkzeug - der Petrinetz-Kern. In J. Desel, A. Oberweis und E. Kindler, Hrsg., *3. Workshop Algorithmen und Werkzeuge für Petrinetze*, Seiten 27–32, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1996.
- [DO95] J. Desel und A. Oberweis. Verifikation von Informationssystemen durch Auswertung halbgeordneter Petrinetz-Abläufe: Theoretische Untersuchungen, Methodik und Werkzeug - eine Projektübersicht. Forschungsbericht Nr. 375, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1995.
- [DO96] J. Desel und A. Oberweis. Petrinetze in der Angewandten Informatik - Einführung, Grundlagen und Perspektiven. *Wirtschaftsinformatik*, 38:359–368, Juli 1996.
- [DR98] J. Desel und W. Reisig. Place/Transition Petri Nets. In W. Reisig und G. Rozenberg, Hrsg., *Lectures on Petri Nets I: Basic Models*, Vol. 1491 of *Lecture Notes in Computer Science*, Seiten 122–173. Springer, Berlin, Heidelberg, 1998.
- [EH95] G. Erdl und H.G.Schönecker. *Workflowmanagement: Workflow-Produkte und Geschäftsprozeßoptimierung, B.BIT Consult*. Gabler, Wiesbaden, 1995.
- [EKO96] P. Elgass, H. Krcmar und A. Oberweis. Von der informalen zur formalen Geschäftsprozessmodellierung. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 141–162. International Thomson Publishing, Bonn, Albany, 1996.
- [EKR95] C. Ellis, K. Keddera und G. Rozenberg. Dynamic change within workflow systems. In N. Comstock und C. Ellis, Hrsg., *Conf. on Organizational Computing Systems*, Seiten 10 – 21. ACM SIGOIS, ACM. Milpitas, CA, 1995.
- [EN93] C.A. Ellis und G.J. Nutt. Modelling and Enactment of Workflow Systems. In M. Ajmone Marsan, Hrsg., *Application and Theory of Petri Nets 1993*, Vol. 691 of *Lecture Notes in Computer Science*, Seiten 1–16. Springer, Berlin, Heidelberg, 1993.
- [EN94] J. Esparza und M. Nielsen. Decidability Issues for Petri Nets - a Survey. *Journal of Information Processing and Cybernetics*, 30(3):210–242, 1994.
- [Erw98] T. Erwin. Anbindung von VIPsim an den Petrinetz-Kern. Technischer Bericht, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1998.
- [Esp94] J. Esparza. Model checking using net unfoldings. *Science of Computer Programming*, 23:151–195, 1994.



- [FHS83] K.W. Fuhrmann, H. Heisterkamp und K. Schröter. *Fertigungs- und Materialwirtschaft: Strukturen - Schwachstellen - Rationalisierungsansätze: Untersuchungsbericht d. REFA-Verb. für Arbeitsstudien u. Betriebsorganisation e.V., Darmstadt*. Betriebs-technische Reihe. Beuth, Berlin, Köln, 1983.
- [Fre86] C.-C. Freidank. *Kostenrechnung*. Oldenbourg, München, Wien, 1986.
- [Fre96] T. Freytag. Simulation halbgeordneter Petrinetz-Abläufe. In J. Desel, A. Oberweis und E. Kindler, Hrsg., *3. Workshop Algorithmen und Werkzeuge für Petrinetze*, Seiten 14–20, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1996.
- [Fre97a] T. Freytag. VIPsim-Erweiterungsmodul Referenzhandbuch. Technischer Bericht, Universität Karlsruhe, Institut AIFB, Karlsruhe, 1997.
- [Fre97b] T. Freytag. VIPtool - Ein halbordnungs-basiertes Simulations- und Validationswerkzeug für Petrinetze. In J. Desel, A. Oberweis und E. Kindler, Hrsg., *4. Workshop Algorithmen und Werkzeuge für Petrinetze*, Vol. 85, Seiten 7–12. Institut für Informatik, Humboldt-Universität zu Berlin, Oktober 1997.
- [Fre01] T. Freytag. *Softwarevalidierung durch Auswertung von Petrinetz-Abläufen*. Dissertation, Universität Karlsruhe, 2001.
- [FS93] O.K. Ferstl und E.J. Sinz. Geschäftsprozeßmodellierung. *Wirtschaftsinformatik*, 35(6):589–592, 1993.
- [FS95] O.K. Ferstl und E.J. Sinz. Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. *Wirtschaftsinformatik*, 37(3):209–220, 1995.
- [FS96] O.K. Ferstl und E.J. Sinz. Geschäftsprozeß-Modellierung im Rahmen des Semantischen Objektmodells. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 47–61. International Thomson Publishing, Bonn, Albany, 1996.
- [FS98] O.K. Ferstl und E.J. Sinz. *Grundlagen der Wirtschaftsinformatik - Band 1*. Oldenbourg, München, Wien, 1998.
- [Gab94] *Gabler-Wirtschafts-Lexikon*. Gabler, Wiesbaden, 1994.
- [Gai83] M. Gaitanides. *Prozeßorganisation*. Hanser, München, 1983.
- [Gal87] T. Gal, Hrsg. *Grundlagen des Operations Research*. Springer, Berlin, Heidelberg, 1987.
- [Gen87] H.J. Genrich. Predicate/Transition-Nets. In W. Brauer, W. Reisig und G. Rozenberg, Hrsg., *Advances in Petri Nets 1986 Part I: Petri Nets, central models and their properties*, Vol. 254 of *Lecture Notes in Computer Science*, Seiten 207–247. Springer, Berlin, Heidelberg, 1987.
- [GHS95] D. Georgakopoulos, M. Hornick und A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3:119–153, 1995.
- [GK96] V. Gruhn und M. Kampmann. Modellierung unternehmensübergreifender Geschäftsprozesse mit FUNSOFT-Netzen. *Wirtschaftsinformatik*, 38:369–381, 1996.
- [GL81] H.J. Genrich und K. Lautenbach. System modelling with high level Petri nets. *Theoretical Computer Science*, 13:109–136, 1981.
- [GP98] E. Gelenbe und G. Pujolle. *Introduction to Queueing Networks*. John Wiley and Sons, Chichester, 1998.
- [GR93] J. Gray und A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers, Inc., San Francisco, 1993.

- [Gra00] J.E. Grayson. *Python and Tkinter Programming*. Manning Publications, 2000.
- [Göt95] K.G. Götzer. *Workflow: Unternehmenserfolg durch effizientere Arbeitsabläufe. Technik, Einsatz, Fallstudien*. Computerwoche-Verlag, München, 1995.
- [Ham90] M. Hammer. Reengineering work: Don't automate, Obliterate. *Harvard Business Review*, Seiten 104–112, July/August 1990.
- [HB96] T. Hess und L. Brecht. *State of the Art des Business Process Redesign. Darstellung und Vergleich bestehender Methoden*. Gabler, Wiesbaden, 1996.
- [HBs95a] T. Hess, L. Brecht und H. Österle. Geschäftsstrategie-Prozeß-Informationssystem. In H. König, Hrsg., *Wirtschaftsinformatik '95: Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit*. Physica, Heidelberg, 1995.
- [HBs95b] T. Hess, L. Brecht und H. Österle. Stand und Defizite der Methoden des Business Process Redesign. *Wirtschaftsinformatik*, 37:480–486, 1995.
- [HC93] M. Hammer und J. Champy. *Reengineering the corporation*. Nicolas Brealey Publishing, London, 1993.
- [HC95] M. Hammer und J. Champy. *Business Reengineering: die Radikalkur für das Unternehmen*. Campus, Frankfurt am Main, New York, 1995.
- [Hei96] H. Heilmann. Die Integration der Aufbauorganisation in Workflow-Management-Systeme. In H. Heilmann, L.J. Heinrich und F. Rothmayr, Hrsg., *Information Engineering*. Oldenbourg, München, Wien, 1996.
- [Her96] H. Herbst. Business Rules in Systems Analysis: a Meta-Model and Repository System. *Information Systems*, 21:147–166, 1996.
- [HHJ+98] P. Heidl, S. Horn, S. Jablonski, J. Neeb, K. Stein und M. Teschke. A comprehensive approach to flexibility in workflow management systems. Technischer Bericht TR-16-1998-6, Universität Erlangen-Nürnberg, Erlangen, 1998.
- [HK96] H. Herbst und G. Knolmayer. Petrinets as derived process representations in the BROCOM approach. *Wirtschaftsinformatik*, 38:391–398, Juli 1996.
- [HR83] T. Härder und A. Reuter. Principles of Transaction-Oriented Database Recovery. *ACM Computing Survey*, 15:287–317, 1983.
- [HR98] K.M. van Hee und H.A. Reijers. An analytical method for assessing business processes. Computing Science Reports 98/09, Eindhoven University of Technology, Eindhoven, 1998.
- [HVA97] D. Hauschildt, H.M.W. Verbeek und W.M.P. van der Aalst. WOFLAN: a Petri-net-based Workflow Analyzer. Computing Science Reports 97/12, Eindhoven University of Technology, Eindhoven, 1997.
- [IDS00] IDS Scheer. Geschäftsprozeßmanagement: Prozessdesign, -implementierung, -analyse und -optimierung. ARIS Toolset Produkte, Version 4.1. White Paper, Februar 2000. Technischer Bericht, IDS Scheer AG, Saarbrücken, 2000.
- [Jae96] P. Jaeschke. Geschäftsprozessmodellierung mit INCOME. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 141–162. International Thomson Publishing, Bonn, Albany, 1996.
- [JB96] S. Jablonski und C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, 1996.
- [JBS97] S. Jablonski, M. Böhm und W. Schulze, Hrsg. *Workflow-Management; Entwicklung von Anwendungen und Systemen*. Dpunkt, Heidelberg, 1997.
- [Jen95] K. Jensen. *Coloured Petri Nets, Volume 2: Analysis Methods*. EATCS monographs on Theoretical Computer Science. Springer, Berlin, Heidelberg, 1995.

- [Jen96] K. Jensen. *Coloured Petri Nets. Volume 1: Basic concepts*. EATCS monographs on Theoretical Computer Science. Springer, Berlin, Heidelberg, 1996.
- [JMPW93] H.J. Johansson, P. McHugh, A.J. Pendlebury und W.A. Wheeler. *Business Process Reengineering: Breakpoint Strategies for Market Dominance*. Wiley & Sons, New York, Chichester, 1993.
- [JS98] S. Jablonski und K. Stein. Ein Vorgehensmodell für Workflow-Management-Anwendungen. In R. Kneuper, G. Müller-Luschnat und A. Oberweis, Hrsg., *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*, Reihe Wirtschaftsinformatik, Seiten 136–151. Teubner, Stuttgart, Leipzig, 1998.
- [Kel99] G. Keller & Partner. *SAP R/3 prozeßorientiert anwenden*. Edition SAP. Addison-Wesley, Reading, Massachusetts, 1999.
- [KK92] A. Kieser und H. Kubicek. *Organisation*. München, Wien, 1992.
- [KL96] K. Küting und P. Lorson. Benchmarking von Geschäftsprozessen als Instrument der Geschäftsprozeßanalyse. In C. Berkau und P. Hirschmann, Hrsg., *Kostenorientiertes Geschäftsprozessmanagement: Methoden, Werkzeuge, Erfahrungen*, Seiten 121–140. Vahlen, München, 1996.
- [Kle75] L. Kleinrock. *Queueing systems, Vol.1: Theory*. Wiley-Interscience, London, 1975.
- [KM95] S. Khoshanfian und M. Buckiewicz. *Introduction to Groupware, Workflow, and Workgroup Computing*. John Wiley and Sons, New York, 1995.
- [KNS92] G. Keller, M. Nüttgens und A.W. Scheer. Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Universität des Saarlands, Saarbrücken, 1992.
- [Kou95] T.M. Koulopoulos. *The Workflow Imperative*. Van Nostrand Reinhold, New York, 1995.
- [KS97] K.-P. Kistner und M. Seven. *Betriebswirtschaftslehre im Grundstudium 2 - Buchführung, Kostenrechnung, Bilanzen*. Physica, Heidelberg, 1997.
- [KSW97a] E. Kindler, I. Schwenzer und M. Weber. Der Petrinetz-Kern: Dokumentation der Anwendungs-Schnittstelle (V.0.8). Technischer Bericht, Humboldt Universität Berlin, Institut für Informatik, Berlin, 1997.
- [KSW97b] E. Kindler, I. Schwenzer und M. Weber. Dokumentation für den Petri Netz Kern - Editor-konzepte, 22. Oktober 1997. Technischer Bericht, Humboldt Universität Berlin, Institut für Informatik, Berlin, 1997.
- [Law97] P. Lawrence, Hrsg. *Workflow Handbook 1997, Workflow Management Coalition*. John Wiley and Sons, New York, 1997.
- [LK66] R. Levin und C. Kirkpatrick. *Planning and control with PERT/CPM*. McGraw-Hill, New York, 1966.
- [LPG98] V. Lullies, M. Pastowsky und S. Grandke. Geschäftsprozesse optimieren - ohne Diktat der Technik. *Harvard Business Manager*, (2):65–72, 1998.
- [LSKM00] Z. Luo, A. Sheth, K. Kochut und J. Miller. Exception Handling in Workflow Systems. *Applied Intelligence: the International Journal of AI, Neural Networks, and Complex Problem-Solving Technologies*, 13(2), 2000.
- [LSW97] P. Langner, C. Schneider und J. Wehler. Ereignisgesteuerter Prozessketten und Petri-Netze. Universität Hamburg, Hamburg, 1997.
- [Lut96] M. Lutz. *Programming Python*. O'Reilly & Associates, Sebastopol, 1996.
- [Mar89] J. Martin. *Information engineering: a trilogy*. Prentice-Hall, Englewood Cliffs, 1989.

- [Mat93] J. Matulka. Vernetzt-Verteilt-Parallel: Die Rolle moderner DV-Architekturen in Organisationen. In W. Frisch und A. Taudes, Hrsg., *Informationswirtschaft*, Seiten 263–272, Physica, Heidelberg, 1993.
- [MB93] D. Morris und J. Brandon. *Reengineering your business*. McGraw-Hill, New York, 1993.
- [McM92] K.L. McMillan. Using unfoldings to avoid the state space explosion problem in the verification of asynchronous circuits. In *Proceedings of the 4th Workshop on Computer Aided Verification, Montreal*, Seiten 164–174, 1992.
- [MK96] R.L. Manganelli und M.K. Klein. *The Reengineering Handbook: A Step-By-step Guide to Business Transformation*. Amacom, 1996.
- [Mül73] H. Müller-Merbach. *Operations Research*. Vahlers Handbücher der Wirtschafts- und Sozialwissenschaften. Vahlen, München, 1973.
- [Mül95] S. Müller. *Ablaufmodellierung als Analyse-, Entwurfs- und Realisierungsmethodik im Softwareentwicklungsprozeß*. Reihe: Wirtschaftsinformatik. Eul, Bergisch Gladbach, Köln, 1995.
- [MS99] B.F. Müller und P. Stolp. *Workflowmanagement in der industriellen Praxis*. Springer, Berlin, Heidelberg, 1999.
- [NM93] K. Neumann und M. Morlock. *Operations Research*. Hanser, München, Wien, 1993.
- [NS79] K. Neumann und U. Steinhardt. GERT Networks. Vol. 172 of *Lecture Notes in Economic and Mathematical Systems*. Springer, Berlin, Heidelberg, 1979.
- [Nüt97] M. Nüttgens. Ereignisgesteuerte Prozeßketten (EPK) -Forschungsansätze in der wissenschaftlichen Literatur und Praxis. In J. Desel und H. Reichel, Hrsg., *Grundlagen der Parallelität, Workshop der GI-Fachgruppen 0.0.1 und 0.0.7 im Rahmen der INFORMATIK '97*, Vol. TUD/FI 97/13. Technische Universität Dresden, 1997.
- [Obe90] A. Oberweis. *Zeitstrukturen für Informationssysteme*. Dissertation, Universität Mannheim, 1990.
- [Obe96] A. Oberweis. *Modellierung und Ausführung von Workflows mit Petri-Netzen*. Reihe Wirtschaftsinformatik. Teubner, Stuttgart, Leipzig, 1996.
- [Öst95a] H. Österle. *Business Engineering. Prozeß- und Systementwicklung. Bd. 1: Entwurfstechniken*. Springer, Berlin, Heidelberg, 1995.
- [Öst95b] H. Österle. *Business in the Information Age - Heading for New Processes*. Springer, Berlin, Heidelberg, 1995.
- [Öst96] H. Österle. *Business Engineering. Prozeß- und Systementwicklung. Bd. 2: Fallbeispiel*. Springer, Berlin, Heidelberg, 1996.
- [OSS94] A. Oberweis, G. Scherrer und W. Stucky. INCOME/STAR: Methodology and tools for the development of distributed information systems. *Information Systems*, 19(8):643–660, 1994.
- [OSS+97] A. Oberweis, R. Schätzle, W. Stucky, W. Weitz und G. Zimmermann. Geschäftsstrategie-Prozeß-Informationssystem. In H. Krallmann, Hrsg., *Wirtschaftsinformatik '97: Internationale Geschäftstätigkeit auf der Basis flexibler Organisationsstrukturen und leistungsfähiger Informationssysteme*. Physica, Heidelberg, 1997.
- [Ous94] J. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, Massachusetts, 1994.
- [Pal97] Pallas Athena. *Protos User Manual*. Pallas Athena BV, Plasmolen, The Netherlands, 1997.

- [PB95] A. Picot und M. Böhme. Prozeßorganisation. Eine Bewertung der Ansätze aus Sicht der Organisationslehre. In M. Nippa und A. Picot, Hrsg., *Prozeßmanagement und Reengineering*, Seiten 13–38. Frankfurt, 1995.
- [Pet62] C.A. Petri. *Kommunikation mit Automaten*. Dissertation, Institut für instrumentelle Mathematik, Bonn, 1962.
- [PH96] G. Poys sick und S. Hannaford. *Workflow Reengineering*. Adobe Press, Mountain View, CA, 1996.
- [PRO01] PROMATIS. INCOME Management von Geschäftsprozessen. Technischer Bericht, PROMATIS AG, Karlsbad, 2001.
- [PX96] J.M. Proth und X. Xie. *Petri Nets - a tool for design and management of manufacturing systems*. Wiley & Sons, New York, Chichester, 1996.
- [Rei86] W. Reisig. *Petrinetze: Eine Einführung*. Studienreihe Informatik. Springer, Berlin, Heidelberg, 1986.
- [Rei98] W. Reisig. *Elements of Distributed Algorithms*. Springer, Berlin, Heidelberg, 1998.
- [Reu95] B. Reuter. *Direkte und indirekte Wirkungen rechnergestützter Fertigungssysteme: formalisierte Netzstrukturen zur Darstellung und Analyse der Unternehmung*. Beiträge zur Wirtschaftsinformatik. Physica, Heidelberg, 1995.
- [Rol96] D. Roller. Verifikation von Workflows in IBM FlowMark. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 353–368. International Thomson Publishing, Bonn, Albany, 1996.
- [RR98a] W. Reisig und G. Rozenberg, Hrsg. *Lectures on Petri Nets I: Basic Models*, Vol. 1491 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 1998.
- [RR98b] W. Reisig und G. Rozenberg, Hrsg. *Lectures on Petri Nets II: Applications*, Vol. 1492 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 1998.
- [RS96] M. Rosemann und R. Schulte. Effiziente Prozeßgestaltung im Rechnungswesen. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 193–207. International Thomson Publishing, Bonn, Albany, 1996.
- [Rum97] F. Rump. Analyse ereignisgesteuerter Prozeßketten (EPK). In J. Desel und H. Reichel, Hrsg., *Grundlagen der Parallelität, Workshop der GI-Fachgruppen 0.0.1 und 0.0.7 im Rahmen der INFORMATIK '97*, Vol. TUD/FI 97/13. Technische Universität Dresden, 1997.
- [Rum99] F. Rump. *Geschäftsprozessmanagement auf der Basis ereignisgesteuerter Prozeßketten*. Reihe Wirtschaftsinformatik. Teubner, Stuttgart, Leipzig, 1999.
- [Sch91] G. Schmidt. *Methode und Techniken der Organisation*. Schriftenreihe Der Organisator. Verlag Dr. Götz Schmidt, Gießen, 1991.
- [Sch94] A.W. Scheer. *Business Process Engineering, ARIS-Navigator for Reference Models for Industrial Enterprises*. Springer, Berlin, Heidelberg, 1994.
- [Sch95] A.-W. Scheer. *Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse*. Springer, Berlin, Heidelberg, 1995.
- [Sch96] A.-W. Scheer. Modellunterstützung für das kostenorientierte Geschäftsprozessmanagement. In C. Berkau und P. Hirschmann, Hrsg., *Kostenorientiertes Geschäftsprozessmanagement: Methoden, Werkzeuge, Erfahrungen*, Seiten 3–25. Vahlen, München, 1996.
- [Sch97] G. Schmidt, Hrsg. *Prozeßmanagement: Modelle und Methoden*. Springer, Berlin, Heidelberg, 1997.

- [Sch99] P. Scharbert. Spezifikation semi- und hybrid strukturierter Geschäftsprozesse mit Petri-Netzen und Business Rules. Diplomarbeit, Institut AIFB, Universität Karlsruhe, 1999.
- [Sch00] R. Schätzle. *Workflow-Management - ein ereignisbasierter Ansatz*. Libri Books on Demand, 2000.
- [SGJ+96] A. Sheth, D. Georgakopoulos, S. Joosten, M. Rusinkiewicz, W. Scacchi, J. Wielen und A. Wolf. Report from the NSF workshop on workflow and process automation in information systems. *SIGMOD Record*, 25(4):55–67, 1996.
- [SJ96] A.W. Scheer und W. Jost. Geschäftsprozeßmodellierung innerhalb einer Unternehmensarchitektur. In G. Vossen und J. Becker, Hrsg., *Geschäftsprozessmodellierung und Workflow-Management*, Seiten 29–46. International Thomson Publishing, Bonn, Albany, 1996.
- [SL96] Software-Ley. *COSA User Manual*. Software-Ley GmbH, Pullheim, Germany, 1996.
- [SM96] A.R. Simon und W. Marion. *Workgroup computing: workflow, groupware and messaging*. McGraw-Hill, New York, 1996.
- [Smi98] E. Smith. Principles of High-Level Net Theory. In W. Reisig und G. Rozenberg, Hrsg., *Lectures on Petri Nets I: Basic Models*, Vol. 1491 of *Lecture Notes in Computer Science*, Seiten 174–210. Springer, Berlin, Heidelberg, 1998.
- [SO95] P.A. Steinbuch und K. Olfert. *Fertigungswirtschaft*. Kompendium der praktischen Betriebswirtschaft. Kiehl, Ludwigshafen, 1995.
- [SR93] G. Scheschonk und W. Reisig, Hrsg. *Petri-Netze im Einsatz für Entwurf und Entwicklung von Informationssystemen*. Informatik aktuell. Springer, Berlin, Heidelberg, 1993.
- [Sta97] Staffware. *Staffware 97 / GWD User Manual*. Staffware plc, Berkshire, United Kingdom, 1997.
- [Tij94] H.C. Tijms. *Stochastic Models: an Algorithmic Approach*. John Wiley and Sons, New York, 1994.
- [Uth01] C.v. Uthmann. *Geschäftsprozesssimulation von Supply Chains*. Advances in Simulation. SCS - Society for Computer Simulation International, San Diego, Erlangen, Ghent, Delft, 2001.
- [Val96] A. Valmari. The State Explosion Problem. In W. Reisig and G. Rosenberg, Hrsg., *Advanced Course on Petri Nets (ACPN)*, Seiten 2053–2070, 1996.
- [VB96] G. Vossen und J. Becker, Hrsg. *Geschäftsprozeßmodellierung und Workflow-Management*. International Thomson Publishing, Bonn, Albany, 1996.
- [VBA01] H.M.W. Verbeek, T. Basten und W.M.P. van der Aalst. Diagnosing workflow processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.
- [VG92] R. Jegenka und V. Gruhn. An evaluation of FUNSOFT nets. In J.-C. Derniame, Hrsg., *Software Process Technology - Proceedings of the 22nd European Software Process Modeling Workshop, Trondheim, Norwegen*, Lecture Notes in Computer Science, Seiten 194–214. Springer, Berlin, Heidelberg, 1992.
- [vHR00] K.M. van Hee und H.A. Reijers. Using Formal Analysis Techniques in Business Process Redesign. In W.M.P. van der Aalst, J. Desel und A. Oberweis, Hrsg., *Business Process Management: Models, Techniques, and Empirical Studies*, Vol. 1806 of *Lecture Notes in Computer Science*, Seiten 142–160. Springer, Berlin, Heidelberg, 2000.

- [Vos87] K. Voss. Nets in Data Bases. In W. Brauer, W. Reisig und G. Rozenberg, Hrsg., *Advances in Petri Nets 1986 Part II: Petri Nets, central models and their properties*, Vol. 254 of *Lecture Notes in Computer Science*, Seiten 234–257. Springer, Berlin, Heidelberg, 1987.
- [Wei98] W. Weitz. Combining structured documents with high-level Petri nets for workflow modeling in internet-based commerce. *International Journal of Cooperative Information Systems*, 7(4):275–296, 1998.
- [Wei00] W. Weitz. *Integrierte Dokumenten- und Ablaufmodellierung im Electronic Commerce*. Berichte aus der Informatik. Shaker, Aachen, 2000.
- [WFM94] WFMFC. Workflow reference model. Workflow Management Coalition, Brüssel, 1994.
- [WFM96] WFMFC. Workflow Management Coalition Terminology and Glossary (WFMFC-TC-1011). Technischer Bericht, Workflow Management Coalition, Brüssel, 1996.
- [WW67] G. Waschek und E. Weckerle. *Die Praxis der Netzplantechnik*. Verlag für Unternehmensführung Dr. Max Gehlen, Baden-Baden, Bad Homburg v.d.H., 1967.
- [Zel96] S. Zelewski. Eignung von Petrinetzen für die Modellierung komplexer Realsysteme - Beurteilungskriterien. *Wirtschaftsinformatik*, 38:369–381, 1996.
- [Zen69] Zentralverband der Elektrotechnischen Industrie e.V., Hrsg. ZVEI-Kennzahlensystem. Ein Instrument zur Unternehmenssteuerung. Betriebswirtschaftliche Schriftenreihe des ZVEI, Frankfurt am Main, 1969.
- [ZR96] O. Zukunft und F. Rump. From Business Process Modeling to Workflow Management: An Integrated Approach. In B. Scholz-Reiter und E. Stickel, Hrsg., *Business Process Modeling*, Seiten 3–22. Springer, Berlin, Heidelberg, 1996.