

Modellierungsmethode zur Auftragsabwicklung in der Automobilindustrie

Dissertationsschrift
zur Erlangung des ingenieurwissenschaftlichen Doktorgrades des
Fachbereichs Maschinenbau der Universität Dortmund

vorgelegt von

Dipl.-Inform. Axel Wagenitz
geboren am 22.07.1965

Düsseldorf, im Januar 2007

Erstgutachter: Prof. Dr.-Ing. A. Kuhn

Inhaltsverzeichnis

1. Einleitung	1
1.1. Ausgangssituation und Problemstellung	1
1.2. Inhalt der Arbeit	6
2. Auftragsabwicklung in der Automobilindustrie	7
2.1. Auftragsabwicklung in der Automobilindustrie	8
2.2. Elemente des Auftragsabwicklungsprozesses	11
2.2.1. Prozesskette »Prognosen«	12
Prognoseerstellung und Volumenvereinbarung Handel	12
Programmplanung und Kapazitätsmanagement	15
2.2.2. Prozesskette »Auftragsabwicklung«	15
Auftragsmanagement	16
Fertigungsprogrammplanung und Sequenzierung	19
Fahrzeugfertigung	21
Distribution	22
2.2.3. Prozesskette »Teile«	24
2.3. Kennzahlen und Ziele bei der Auftragsabwicklung	26
3. Aufgabenstellungen im Auftragsabwicklungsprozess	30
3.1. Einsatz im strategischen Bereich	31
3.1.1. Marktorientierte Produktion	31
Fragestellung	31
Lösungsstrategie	34
Lösungsvoraussetzungen	35
3.1.2. Belegung von Werken	36
Fragestellung	36
Lösungsstrategie	37
Lösungsvoraussetzungen	38
3.2. Einsatz im taktischen Bereich	39
3.2.1. Bedarfs- und Kapazitätsmanagement	39
Fragestellung	39
Lösungsstrategie	42
Lösungsvoraussetzungen	43
3.2.2. Werksprogrammplanung	43
Fragestellung	43
Lösungsstrategie	44
Lösungsvoraussetzungen	44
3.3. Einsatz im operativen Bereich	45
3.3.1. Distributionsplanung	45

Fragestellung	45
Lösungsstrategie	46
Lösungsvoraussetzungen	47
3.3.2. Störungsmanagement	47
Fragestellung	47
Lösungsstrategie	47
Lösungsvoraussetzungen	48
4. Anforderungen an einen Modellierungs- und Bewertungsansatz	50
4.1. Ausgangssituation	50
4.2. Allgemeine Anforderungen	53
4.3. Kommunikation und Dokumentation	56
4.4. Bewertung der Ergebnisse	57
4.5. Systemlast	59
4.5.1. Quellen für Aufträge	59
4.5.2. Baubarkeit von Aufträgen	60
4.5.3. Verteilung von Eigenschaften auf Aufträge	61
4.6. Abbildung von Werken	64
4.6.1. Ressourcen	65
4.6.2. Algorithmen	66
Einplanung von Aufträgen	66
Berechnung Auftragsstatus	67
Störungsmanagement	68
Kapazitätsverwaltung und -anpassung	68
4.7. Abbildung von Zulieferern	69
4.8. Abbildung der Distribution	70
4.8.1. Distribution gefertigter Fahrzeuge	71
4.8.2. Distributionsprozesse in der Teileversorgung	71
4.9. Modellgröße, Laufzeit	72
4.10. Automatisierbarkeit des Modellaufbaus	74
4.11. Zusammenfassung	74
5. Abgleich des Anforderungsprofils mit existierenden Ansätzen	76
5.1. Auswahl eines Modellierungsansatzes	76
5.2. Existierende Modellierungssprachen	80
5.3. Festlegung der Softwareplattform	85
5.4. Zusammenfassung	92
6. Modellierungs- und Bewertungsansatz	94
6.1. Grundlagen	94
6.2. Lastgenerator	100
6.2.1. Produktbeschreibung	100
6.2.2. Objektorientierte Struktur	104
6.2.3. Einbauraten	106
6.2.4. Wertverläufe	108
6.2.5. Erzeugung von Aufträgen	108
6.2.6. Konstruktionsregeln	110

6.2.7. Stücklisten	113
6.2.8. Zusammenfassung	116
6.3. Märkte und Händler	116
6.3.1. Einplanung von Aufträgen	117
6.3.2. Kundenverhalten	120
6.3.3. Zusammenfassung	122
6.4. Ressourcen	122
6.5. Werke	125
6.5.1. Kapazitäten	125
6.5.2. Einplanung von Aufträgen	126
6.5.3. Störungsmanagement	130
6.5.4. Fertigung	132
6.5.5. Zusammenfassung	132
6.6. Zulieferer	133
6.6.1. Verwaltung von Kapazitätsinformationen	134
6.6.2. Vorproduktion von Teilen	135
6.6.3. Begrenzung der Ausbringung	136
6.6.4. Flexibilitätsbandbreiten	137
6.6.5. Störungsmanagement	138
6.6.6. Zusammenfassung	139
6.7. Distribution	139
6.7.1. Transportrelationen	141
6.7.2. Transportdisposition	143
6.7.3. Progressive und retrograde Terminierung	144
6.7.4. Distributionsprozesse in der Beschaffung	145
6.7.5. Zusammenfassung	147
6.8. Modellelemente im beispielhaften Zusammenspiel	147
7. Systemarchitektur	150
7.1. Modell- und Ergebnisrepräsentation	152
7.1.1. Eingangsdaten	152
7.1.2. Ergebnisse	153
7.2. Das Bewertungswerkzeug	155
7.2.1. Struktur und Ablauf	155
7.2.2. Ablauf der Simulation	159
Einlesen des Modells	159
Initialisierung	159
Start der Ereignisbearbeitung	161
Ende der Ereignisbearbeitung	162
7.2.3. Kennzahlen	162
7.2.4. Zusammenfassung	164
7.3. Validierung	165
7.4. Abgleich mit den Anforderungen	168
7.5. Zusammenfassung	169
8. Beispielhafter Einsatz	171
8.1. Aufgabenstellung	171

8.2. Modellumfang	171
8.2.1. Produktstruktur	172
8.2.2. Märkte	173
8.2.3. Prozesssteuerung	174
8.2.4. Werke und Zulieferer	174
8.2.5. Distribution	175
8.3. Ergebnisse	176
8.4. Weitere Anwendungsbeispiele	178
8.5. Laufzeitverhalten	180
8.6. Zusammenfassung	181
9. Zusammenfassung und Ausblick	183
Abbildungsverzeichnis	187
Literatur	190
A. Anhang – Entwicklungssystem	216
B. Anhang – Teile des Beispielmodells	218
C. Anhang – Beispiel Produktbeschreibung	226
D. Anhang – Ausgewählte UML Diagramme	227
D.1. Klassenstruktur CSimObject	228
D.2. Klassenstruktur CExperiment	229
D.3. Klassenstruktur CSimParameters	230
D.4. Zentrale Klassen der Produktbeschreibung	231
D.5. Klassenstruktur CBOStructure und CBOClass	232
D.6. Klassenstruktur CPropertyCluster	233
D.7. Klassenstruktur CPropertySpec	234
D.8. Klassenstruktur CBOM	235
D.9. Klassenstruktur COrderModifier	236
D.10. Klassenstruktur CDistributionSystem	237
D.11. Klassenstruktur CDistributionChannel	238
D.12. Klassenstruktur CTransportRelation	239
D.13. Klassenstruktur CRoutingTable	240
D.14. Klassenstruktur CLogisticsServiceProvider	241
D.15. Klassenstruktur CTransport	242
D.16. Klassenstruktur CContainer	243
D.17. Klassenstruktur CRouteInformation	244
D.18. Klassenstruktur CTransportOrder	245
D.19. Klassenstruktur CMarket	246
D.20. Klassenstruktur CDealer	247
D.21. Klassenstruktur CDealer und CDealerOrderPool	248
D.22. Klassenstruktur CCustomerType	249
D.23. Klassenstruktur CPlanningDealer	250
D.24. Klassenstruktur CResource	251

D.25. Klassenstruktur CResourceLeaf	252
D.26. Klassenstruktur CResourceAssocData	253
D.27. Klassenstruktur CResourceWeekTime	254
D.28. Klassenstruktur CProcess	255
D.29. Klassenstruktur CPlant	256
D.30. Klassenstruktur CFailure	257
D.31. Klassenstruktur CProductPattern	258
D.32. Klassenstruktur CProbabilityDistribution	259
D.33. Klassenstruktur COrderBank	260
D.34. Klassenstruktur CPlanning	261
D.35. Klassenstruktur CSupplier	262
D.36. Klassenstruktur CSupplierBTO	263
D.37. Klassenstruktur CSupplierBTS	264
D.38. Klassenstruktur CSupplierCustomer	265
D.39. Klassenstruktur CAssocData	266
D.40. Von CAssocData abgeleitete Klassen	267
D.41. Klassenstruktur CSchedule	268
E. Anhang – Ausgewählte OTD-NET-Klassen	269
E.1. Klassenstruktur AGGREGATINGRESOURCE	269
E.2. Klassenstruktur ASS_DTA	269
E.3. Klassenstruktur ASSOCDAT	269
E.4. Klassenstruktur ASSOCDAT_TIMESPAN_FLOAT	270
E.5. Klassenstruktur ASSOCDATCAPACITY	270
E.6. Klassenstruktur ASSOCDATRESOURCE	270
E.7. Klassenstruktur BASEDEALER	271
E.8. Klassenstruktur BASETYPETRANSFORMATION	272
E.9. Klassenstruktur BILLOFMATERIALRECORD	272
E.10. Klassenstruktur BINOMIALDISTRIBUTION	272
E.11. Klassenstruktur BOOLTRANSFORM	272
E.12. Klassenstruktur BUFFER	273
E.13. Klassenstruktur CONTAINER	273
E.14. Klassenstruktur COSTSCOMMON	274
E.15. Klassenstruktur COSTSOBJECT	274
E.16. Klassenstruktur CUSTOMERTYPE	274
E.17. Klassenstruktur DAILYRCT	275
E.18. Klassenstruktur DEALER	275
E.19. Klassenstruktur DEALERPLANNING	275
E.20. Klassenstruktur DISTRIBUTION	277
E.21. Klassenstruktur DISTRIBUTIONCHANNEL	277
E.22. Klassenstruktur DT_ASSOCDAT	279
E.23. Klassenstruktur EXPERIMENT	279
E.24. Klassenstruktur EXPONENTIALDISTRIBUTION	280
E.25. Klassenstruktur FAILURERULE	280
E.26. Klassenstruktur FAULT	280
E.27. Klassenstruktur HISTOGRAMDISTRIBUTION	281
E.28. Klassenstruktur HOURLYRCT	281

E.29. Klassenstruktur IMPORTER	281
E.30. Klassenstruktur LOCATION	281
E.31. Klassenstruktur LOGISTICSSERVICEPROVIDER	282
E.32. Klassenstruktur LSPROUTE	282
E.33. Klassenstruktur MARKET	282
E.34. Klassenstruktur MONTHLYRCT	283
E.35. Klassenstruktur MONTHLYTIMESPAN	283
E.36. Klassenstruktur NORMALDISTRIBUTION	283
E.37. Klassenstruktur OBJECTQUOTA	283
E.38. Klassenstruktur ORDERMODIFIER	284
E.39. Klassenstruktur PART	284
E.40. Klassenstruktur PARTREFERENCE	284
E.41. Klassenstruktur PLANNING	285
E.42. Klassenstruktur PLANT	285
E.43. Klassenstruktur PLANTPROPERTYSPECINFORMATIONIN	286
E.44. Klassenstruktur POINTSINTIME	287
E.45. Klassenstruktur POISSONDISTRIBUTION	287
E.46. Klassenstruktur PROBABILITYDISTRIBUTION	287
E.47. Klassenstruktur PROCESS	288
E.48. Klassenstruktur PRODUCTCLASS	288
E.49. Klassenstruktur PRODUCTIONUNITRESOURCE	289
E.50. Klassenstruktur PRODUCTPATTERN	290
E.51. Klassenstruktur PRODUCTQUOTA	290
E.52. Klassenstruktur PRODUCTSHARE	291
E.53. Klassenstruktur PRODUCTSTRUCTURE	291
E.54. Klassenstruktur PROPERTY	291
E.55. Klassenstruktur PROPERTYCONSTRAINT	292
E.56. Klassenstruktur PROPERTYCONSTRAINTBASE	292
E.57. Klassenstruktur PROPERTYCONSTRAINTRULE	292
E.58. Klassenstruktur PROPERTYDEV	293
E.59. Klassenstruktur PROPERTYGROUP	293
E.60. Klassenstruktur PROPERTYSPEC	293
E.61. Klassenstruktur RECURRENTTASK	294
E.62. Klassenstruktur RESOURCE	294
E.63. Klassenstruktur ROUTEINFORMATION	295
E.64. Klassenstruktur ROUTING	295
E.65. Klassenstruktur ROUTINGTABLE	295
E.66. Klassenstruktur SALES	296
E.67. Klassenstruktur SCENARIO	296
E.68. Klassenstruktur SIMOBJECT	296
E.69. Klassenstruktur SIMPARAMETERS	297
E.70. Klassenstruktur SUMRESOURCE	298
E.71. Klassenstruktur SUPPLIER	298
E.72. Klassenstruktur SUPPLIERBTO	299
E.73. Klassenstruktur SUPPLIERBTS	302
E.74. Klassenstruktur SUPPLIERPARTINFORMATION	302
E.75. Klassenstruktur SUPPLIERPARTINFORMATIONIN	302

E.76. Klassenstruktur SUPPLIERPARTINFORMATIONOUT	303
E.77. Klassenstruktur SUPPLIERPRIORITISATION	303
E.78. Klassenstruktur TIMESPAN	304
E.79. Klassenstruktur TRANSPORTDEFINITION	304
E.80. Klassenstruktur TRANSPORTER	304
E.81. Klassenstruktur TRANSPORTRELATION	304
E.82. Klassenstruktur UNIFORMDISTRIBUTION	305
E.83. Klassenstruktur VISUALOBJECT	305
E.84. Klassenstruktur WEEKLYRCT	305
E.85. Klassenstruktur WEEKLYTIMESPAN	305
E.86. Klassenstruktur WEEKTIMERESSOURCE	306
E.87. Klassenstruktur WEIGHTEDREFERENCE	306
E.88. Klassenstruktur YEARLYTIMESPAN	306
F. Anhang – Tabellen Simulationsergebnisse	307
F.1. Aufträge	307
F.1.1. Tabelle OUT_BO	307
F.1.2. Tabelle OUT_CustomerOrder	310
F.1.3. Tabelle OUT_DealerOrder	311
F.1.4. Tabelle OUT_Properties	312
F.2. Märkte und Händler	312
F.2.1. Tabelle OUT_Stat_Dealer	312
F.2.2. Tabelle OUT_Stat_All	313
F.2.3. Tabelle OUT_PropertyQuota	315
F.3. Ressourcen	315
F.3.1. Tabelle OUT_Resource	315
F.4. Prozesse	315
F.4.1. Tabelle OUT_BO_Process	315
F.5. Werke	316
F.5.1. Tabelle OUT_Plant	316
F.6. Zulieferer	317
F.6.1. Tabelle OUT_Supplier	317
F.6.2. Tabelle OUT_Parts	318
F.7. Distribution	319
F.7.1. Tabelle OUT_Transports	319
F.7.2. Tabelle OUT_BO_Transport	319
F.7.3. Tabelle OUT_Stat_Distribution	320
G. Anhang – Auftragsortierung	321
G.1. Vergleichsfunktion	321
G.2. Grammatik	321
G.3. Softwareauswahl nach VDI 3633 Blatt vier	325

Abkürzungsverzeichnis

ARIS	Architektur Integrierter Informationssysteme
ATP	Available-to-Promise
IT	Information Technology
B2B	Business-to-Business
B2C	Business-to-Customer
BOM	Bill-of-Materials
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Modeling
BPMI	Business Process Management Initiative
BPMN	Business Process Modeling Notation
BPR	Business Process Reengineering
BTO	Build-to-Order
CASE	Computer Aided Software Engineering
CPI	Continuous Process Improvement
CTP	Capable-to-Promise
DBMS	Database Management System
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
GME	Graphical Modeling Environment
GOM	Grundsätze ordnungsmäßiger Modellierung
GP	Geschäftsprozess
HTTP	Hyper Text Transfer Protocol
ICDP	International Car Distribution Programme
IDEF	Integrated Definition Methods
ILIPT	Intelligent Logistics for Innovative Product Technologies
KPI	Key Performance Indicators
JIS	Just in Sequence
JIT	Just in Time
NAFTA	North American Free Trade Agreement
ODBC	Open Database Connectivity
ODETTE	Organisation for Data Exchange by Tele Transmission in Europe
OEM	Original Equipment Manufacturer
OLAP	Online Analytical Processing
OTD	Order to delivery
MDAC	Microsoft Data Access Components
MFC	Microsoft Foundation Classes
PLM	Product Life-Cycle Management
PSL	Process Specification Language
PVS	Produkt-Varianten-Struktur

RDBS	Relational Database System
ROCE	Return on Capital Employed
ROI	Return of Investment
SIMD	Single Instruction Multiple Data
SCM	Supply Chain Management
SSE	Streaming SIMD Extensions
UEML	Unified Enterprise Modelling Language
UML	Unified Modeling Language
VDA	Verband der Automobilindustrie

1. Einleitung

1.1. Ausgangssituation und Problemstellung

Die Gestaltung der vielfältigen Prozesse in einem Unternehmen ist eine herausfordernde Aufgabe, insbesondere wenn es sich um komplexe Prozesse handelt, die viele Teilprozesse umfassen und deren Gestaltung zahlreichen Randbedingungen unterliegt. Der Auftragsabwicklungsprozess umfasst die Teilprozesse zwischen Auftragserteilung und Auslieferung eines Produktes und damit z.B. die Zuordnung eines Fertigungsortes, die Sicherstellung von Kapazitäten an diesem Fertigungsort und die Nennung eines Auslieferungstermins an den Auftraggeber.

Sind die Produkte eines Unternehmens aus hunderten oder sogar tausenden Komponenten zusammengesetzt, die von einer großen Anzahl Lieferanten bereitgestellt werden müssen, wie z.B. in der Luftfahrtindustrie oder im Automobilbau, sind in der Prozessgestaltung und -bewertung eine so große Zahl von Abhängigkeiten zu berücksichtigen, dass diese Gestaltung ohne Werkzeugunterstützung nur auf einer sehr abstrakten Ebene möglich ist. Diese werkzeuggestaltete Gestaltung kann z.B. anhand eines *Modells* durchgeführt werden, wobei durch Untersuchungen am Modell Rückschlüsse auf den abgebildeten Prozess gewonnen werden können.

Bislang verfügbare Methoden bzw. Werkzeuge unterstützen zwar die Prozessgestaltung und -bewertung, weisen aber einige Defizite in den Modellierungsmöglichkeiten auf. Diese Möglichkeiten sind insofern beschränkt, dass der maximal mögliche Detaillierungsgrad, z.B. bei der Größe der untersuchbaren Modelle oder bei der Abbildung der Eigenschaften eines Produktes (Produktstruktur), hinter dem zurückbleibt, was für die Prozessgestaltung wünschenswert sein kann.

Insbesondere ist daher bislang eine *ganzheitliche* Modellierung und Bewertung von Auftragsabwicklungsprozessen nicht möglich. Diese Ganzheitlichkeit ist ein Schlüssel für das Verständnis solcher Prozesse.

Können die Elemente, aus denen die Prozessmodelle aufgebaut werden, bei einer sehr abstrakten Abbildung noch so allgemein sein, dass die Elemente für sehr viele Industriezweige eingesetzt werden können, sollten für eine detailliertere Abbildung spezifische Strukturen und Abläufe dieser Industriebereiche in die Elemente aufgenommen werden, die zur Erzeugung von Modellen von Auftragsabwicklungsprozessen dieses Industriebereiches vorgesehen werden.

In dieser Arbeit soll ein Ansatz zur Gestaltung und Bewertung von Auftragsabwicklungsprozessen der Automobilindustrie entwickelt werden, ein Industriebereich, in dem sehr komplexe Produkte den Auftragsabwicklungsprozess durchlaufen und in dem weit verzweigte Produktionsnetzwerke gesteuert werden müssen. Dieser Ansatz soll es ermöglichen, die modellgestützte Prozessgestaltung sowohl detaillierter als auch in einem größeren Rahmen, bezogen auf das Datenvolumen, als bislang möglich durchzuführen.

Dazu sind folgende Aufgaben zu lösen:

1. Erfassung der für die Modellierung relevanten Aspekte des Auftragsabwicklungsprozesses.
2. Ableitung von Anforderungen an einen Gestaltungs- und Bewertungsansatz.
3. Auswahl einer geeigneten Basis für die Entwicklung eines Bewertungswerkzeugs.
4. Entwicklung von Modellelementen für die Abbildung der Auftragsabwicklungsprozesse der Automobilindustrie.
5. Umsetzung des Bewertungswerkzeugs.
6. Prüfung des Ansatzes auf seine Tauglichkeit.

Zahlreiche Veränderungen im Umfeld der Automobilhersteller können dazu führen, dass der Auftragsabwicklungsprozess an neue Anforderungen angepasst werden muss. Die folgenden Stichpunkte (vgl. [MMC04], [PH04, S. 400], [Hol01, S. 17]) werden im Folgenden aufgegriffen und auf ihren (potenziellen) Einfluss auf die Auftragsabwicklung diskutiert:

1. Stark gestiegene Anzahl der Fahrzeugmodelle.
2. Gestiegene Anzahl von Produktionswerken.
3. Erhöhte Produktkomplexität bzw. -variabilität¹.
4. Erhöhte Kundenanforderungen.
5. Anhaltender Preisdruck.
6. Die Erhöhung der Rentabilität wird durch Anteilseigner (Aktionäre) eingefordert.

Gestiegene Anzahl der Fahrzeugmodelle. Alle großen Automobilhersteller haben in den letzten Jahren ihre Produktpaletten erheblich vergrößert und sind immer noch darin begriffen (vgl. [Eco, S. 1], [Hol01, S. 18], [MMC04, S. 27]). Die Karosserieformen Limousine, Kombi und Schrägheck als Variationen eines Baumusters sind nach wie vor am Markt vertreten. Bei vielen Herstellern sind »straßentaugliche Geländewagen«, Coupés und Cabrios sowie familienfreundliche Vans hinzugekommen. So ist z.B. die Anzahl unterschiedlicher Baureihen bei der Marke Mercedes-Benz von 3 im Jahr 1970 auf 15 im Jahr 2004 gestiegen.

Es ist nahe liegend, dass mit der Anzahl neuer Modelle auch der logistische Aufwand, der für eine Marke, bzw. ein Unternehmen zu erbringen ist, stark ansteigt. Neue Modelle vergrößern stets die Anzahl der Teile, die vom Automobilhersteller zu hand-

¹Die *Produktkomplexität* kann in *strukturelle Komplexität* und *funktionale Komplexität* differenziert werden (vgl. [CW01, S. 10]). Die strukturelle Komplexität wird durch die Anzahl der Teile in einem Produkt bestimmt; die funktionale Komplexität durch die Anzahl der funktionalen Interaktionen zwischen diesen Teilen. Bei der Produktvariabilität kann unterschieden werden zwischen *interner*, *externer* und *dynamischer* Produktvariabilität (vgl. [Hol04, S. 15-23]). Die externe Produktvariabilität wird definiert durch die dem Kunden gebotene Auswahl (Karosserieformen, Farben, Ausstattungen, etc.); die interne Variabilität durch die Auswirkungen der externen Variabilität im Herstellungsprozess. Die dynamische Variabilität wird durch die Zunahme der Dimension Zeit zur externen Variabilität gebildet, reflektiert also die Veränderung der externen Variabilität durch z.B. Modellwechsel.

haben sind; aber auch Teile, die für eine zusätzliche Baureihe nicht neu entwickelt werden, können den logistischen Aufwand erhöhen: Die Verwendung von z.B. einem Motor in verschiedenen Baureihen führt dazu, dass diese Baureihen nicht unabhängig voneinander geplant werden können. Veränderungen der Planung einer Baureihe beeinflussen die Kapazitätssituation aller Baureihen, die mit dieser Baureihe »Gleichteile²« aufweisen.

Für jede neue Baureihe ist festzulegen, wie genau der Auftragsabwicklungsprozess hierfür aussehen soll, bzw. zu ermitteln, welche »Leistungen« (vgl. Kapitel 2) den Kunden geboten werden können. Einige der Fragestellungen, die im Zusammenhang mit der Einführung einer neuen Baureihe zu beantworten sind:

- ▷ In welchem (n) Werk (en) werden die Fahrzeuge gebaut?
- ▷ Sind die Lieferanten in der Lage, die benötigten Teile zur Verfügung zu stellen?
- ▷ Hat der zusätzliche Kapazitätsbedarf Auswirkungen auf die Versorgungssituation anderer Baureihen und wie sehen diese aus?
- ▷ Welche Auswirkungen haben unterschiedliche Nachfrageszenarien (optimistisch, pessimistisch) auf das betrachtete Netzwerk?
- ▷ Welche Lieferzeit³ ist in den jeweiligen Märkten zu erwarten?
- ▷ Welche Liefertreue⁴ ist hier zu erwarten?
- ▷ Welche Optionen (Motoren, sonstige Ausstattungen) können den Kunden in den Märkten unter den gegebenen Kapazitätsbeschränkungen zur Auswahl gegeben werden?
- ▷ Welche Kosteneffekte sind zu erwarten?

Gestiegene Anzahl von Produktionswerken. Der Bau neuer Werke durch die Automobilhersteller wird u.a. durch neue Baureihen getrieben (z.B. das Porsche Werk in Leipzig). Weitere Motivationsgründe sind z.B. Zollregelungen, die eine Einfuhr bereits montierter Fahrzeuge verteuern. Treffen die Errichtung eines neuen Werkes und die Einführung einer neuen Baureihe zusammen, sind die oben genannten Fragestellungen weiterhin zu bearbeiten. Die Analyse von unterschiedlichen Szenarien in Bezug auf die Kapazitätssituation gewinnt aber an Bedeutung, weil die Kombination »neue Baureihe, neues Werk« besonders hohe Unsicherheiten in diesem Bereich erwarten lässt.

Erhöhte Produktkomplexität bzw. -variabilität. Viele Ausstattungen von Automobilen können von Kunden zunächst als kostenpflichtige Optionen geordert werden, werden dann aber zu einer »Grundausstattung« und sind also bereits im Grundpreis enthalten, z.B. Klimaanlage, Airbags und viele elektronische Systeme (z.B. ABS – Antiblockiersystem). Jedes der genannten Beispiele umfasst sowohl mechanische als auch elektronische Komponenten, die Verfügbarkeit all dieser Teile muss sichergestellt sein, wenn ein Auftrag in ein Werk eingeplant werden soll. Elektronische Kom-

²Teile, die identisch in mehreren Baureihen verwendet werden.

³Zeit zwischen Auftragserteilung und Auslieferung des Fahrzeugs.

⁴Abweichung zwischen dem bei Auftragserteilung prognostizierten Liefertermin und dem tatsächlichen Liefertermin.

ponenten erzeugen insofern weiterhin zusätzliche Komplexität, als das sichergestellt sein muss, dass die Software der jeweiligen Steuergeräte zusammenpassen muss. Das heißt, es ist nicht ausreichend, dass ein Steuergerät vorhanden ist, sondern es muss in der richtigen Softwareversion vorhanden sein.

Ein wichtiges Instrument der Automobilhersteller zur Erlössteigerung ist es, den Kunden eine lange Aufpreisliste zur Verfügung zu stellen. Durch Ausstattung eines Fahrzeugs mit allen zur Wahl stehenden Optionen lässt sich der Fahrzeugpreis in vielen Fällen verdoppeln, wenn man das günstigste Fahrzeug einer Baureihe mit dem teuersten Möglichen vergleicht. Folgen dieser vielen Optionen aus Sicht des Auftragsabwicklungsprozesses sind (exemplarisch) zum einen wiederum viele Kapazitätsinformationen, die im Blick zu halten sind, aber insbesondere eine hohe Volatilität der Teilebedarfe, da hier die Schätzung der Fahrzeugvolumina erst zusammen mit einer zweiten Schätzung – nämlich des Anteils der Kunden, die eine bestimmte Option wünschen – den tatsächlichen Teilebedarf zu berechnen erlauben.

Besonders problematisch sind Teile, deren Bedarf sich aus der Kombination zweier optionaler Ausstattungen ergibt, z.B. der Lederschaltknauf (Option: Lederausstattung) mit einem speziellen Schriftzug (Option: Sportpaket). Zusammengefasst hat also die Anzahl der zu überwachenden Kapazitäten stark zugenommen und die Variabilität der Fahrzeugausstattungen kann zu einer hohen Varianz der Teilebedarfe führen. Die aus der zunehmenden Produktvariabilität abzuleitenden Fragestellungen an den Auftragsabwicklungsprozess sind z.B. die Frage nach dem Einfluss einer etwaigen Kapazitätsbeschränkung, Liefertreue und Lieferzeit der hiervon betroffenen Fahrzeuge.

Erhöhte Kundenanforderungen. Verbesserungen im Hinblick auf das Fahrverhalten und den Ausstattungsumfang, die von einzelnen Automobilherstellern umgesetzt werden, erhöhen die Erwartungshaltung der Kunden an die gesamte Branche. Eine der Auswirkungen dieser Erwartungshaltung ist die bereits angesprochene, ständige Erhöhung der Produktkomplexität. Aber nicht nur die Automobilindustrie beeinflusst die Erwartungshaltung der Kunden beim Kauf eines Fahrzeugs. Auch Erfahrungen mit anderen Industriezweigen nehmen hier Einfluss. So kann z.B. ein Computer im Internet konfiguriert werden und wird wenige Tage später in genau der gewünschten Ausstattung geliefert. Dass ein auf den ersten Blick ähnlicher Vorgang bei der Automobilindustrie mehrere Monate dauert, ist nicht unmittelbar nachvollziehbar.

Neben höheren Ansprüchen an eine schnelle Lieferung von Produkten steht die Erwartung an eine fristgerechte Lieferung. Viele Unternehmen (z.B. UPS) erlauben eine detaillierte Auftragsverfolgung und erlauben den Kunden eine stetige Überwachung des voraussichtlichen Liefertermins. Der Wunsch der Kunden, eine stetige Auftragsverfolgung durchführen zu können, stellt völlig neue Anforderungen an den Auftragsabwicklungsprozess. Jeder dem Kunden sichtbar gemachte Prozessschritt muss »präsentierbar« sein. Das heißt, ist ein Fahrzeug zum zugesagten Termin noch nicht geliefert und der Kunde sieht über eine Auftragsverfolgung das Fahrzeug tagelang in einem Puffer stehen, kommt zur Verärgerung über die Verspätung, die Verärgerung über die vermeintlich unprofessionelle Ablauforganisation des Automobilherstellers.

Preisdruck. Aufstrebende Nationen, wie Süd-Korea und zukünftig insbesondere China, sorgen mit ihren Produkten für einen starken Preisdruck im Automobilgeschäft. Die Hersteller haben daher einerseits ein Interesse die Kosten gering zu halten, um ihre Produkte zu einem absolut günstigen Preis anbieten zu können und andererseits ein Interesse, technisch höher positionierte Produkte zu einem geringen Mehrpreis anbieten

zu können. Der Auftragsabwicklungsprozess umfasst sehr viele Bereiche eines Unternehmens und ist daher ein wichtiger Ansatzpunkt von Strategien zur Kostensenkung. Die Bewertung der Effekte auf die Kosten ist sicherlich eine der wichtigsten Fragen, die im Zusammenhang mit einer Veränderung eines Auftragsabwicklungsprozesses zu beantworten sind.

Rentabilitätsanforderungen. Eine günstige (hohe) Bewertung eines Unternehmens durch den Kapitalmarkt hat zahlreiche Auswirkungen für ein Unternehmen. Zum Beispiel wird die Wahrscheinlichkeit einer feindlichen Übernahme geringer und die Kosten für Kapitalbeschaffungsmaßnahmen sinken. Eine günstige Kostenstruktur ist nicht nur erforderlich, um dem oben geschilderten Preisdruck begegnen zu können, sondern auch, um dem Anteilseigner eine hohe Rendite bieten zu können.

Sollen Veränderungen des Auftragsabwicklungsprozess an ein sich änderndes Wettbewerbsumfeld zunächst am Modell untersucht werden, ist es offenbar erforderlich, die »Essenz« der Veränderung im Modell abbilden zu können. Fragestellungen, wie z.B. die Frage nach den Auswirkungen der Einführung einer neuen Baureihe auf die Kapazitätssituation in einem Produktionsnetzwerk, lassen erahnen, dass hier eine sehr große Anzahl Parameter zu erfassen und zu bewerten sind.

Der im Folgenden entwickelte Ansatz ermöglicht eine Bewertung alternativer Konzeptionen bzgl. ausgewählter Kenngrößen (siehe Abbildung 1.1) und kann einen Beitrag leisten, die Effizienz der Auftragsabwicklungsprozesse von Automobilunternehmen zu steigern.

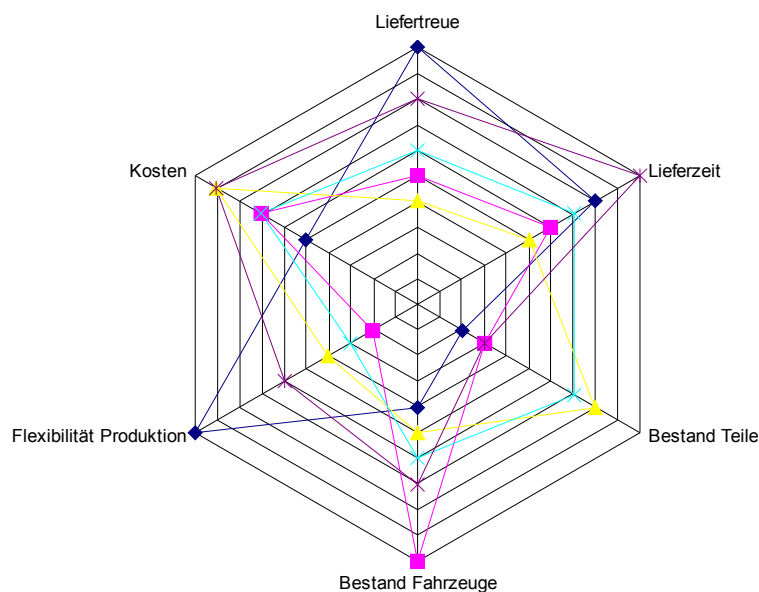


Abbildung 1.1.: Vergleich alternativer Konzepte im Auftragsabwicklungsprozess

1.2. Inhalt der Arbeit

Im Anschluss an die Einleitung werden in Kapitel 2 die Bestandteile der Auftragsabwicklungsprozesse in der Automobilindustrie aufgeführt. Ziel dieser Darstellung ist es aufzuzeigen, welche Informationen zu welchen Zeitpunkten den Akteuren des Prozesses zur Verfügung stehen und welche Materialflüsse auf dieser Basis »entstehen«.

Die hohe Komplexität des Auftragsabwicklungsprozesses bringt es mit sich, dass von der strategischen Gestaltung bis hin zur operativen Steuerung ein Bedarf an Werkzeugen entsteht, die die Transparenz verbessern und somit die Komplexität beherrschbar machen. Kapitel 3 zeigt exemplarisch jeweils zwei Fragestellungen im Kontext strategischer, taktischer und operativer Planung auf, die die Grundlage bilden, um Anforderungen an einen Ansatz zur Gestaltung und Steuerung von Auftragsabwicklungsprozessen in der Automobilindustrie zu definieren.

Zur Strukturierung dieser Anforderungen werden in Kapitel 4 die »Grundsätze ordnungsmäßiger Modellierung« (GoM) (vgl. [BRS95, S. 438], [BSGI00, S. 13-17]) genutzt. Die eher abstrakten Forderungen aus den GoM, wie z.B. die nach »Klarheit« oder nach »Wirtschaftlichkeit«, werden hier durch Nutzung der Ergebnisse des Kapitels 3 zu konkreten Anforderungen transformiert, die klare Vorgaben sowohl für die Auswahl eines Modellierungsansatzes als auch für die Auswahl eines Bewertungswerkzeugs setzen.

Sowohl im Bereich der Modellierungssprachen, die den Anspruch erheben, für die Abbildung von Unternehmensprozessen geeignet zu sein, als auch im Bereich der Bewertungswerkzeuge, die Unternehmensprozesse bewertbar zu machen versprechen, ist eine große Vielfalt anzutreffen. In Kapitel 5 wird nach einer geeigneten Kombination aus Modellierungssprache und Bewertungswerkzeug gesucht, um die Anforderungen aus Kapitel 4 erfüllen zu können. Das Fehlen einer solchen Kombination, die die benötigten Modellierungsbausteine bereit hält, führt zum Ergebnis, dass diese Bausteine neu geschaffen werden müssen.

Für das Erzeugen von Modellierungsbausteinen im beabsichtigten Umfang ist es zweckmäßig, zunächst einen Rahmen und eine Infrastruktur zu schaffen, die grundlegende »Dienste« für die Bausteine zur Verfügung stellt. In den Kapiteln 6 und 7 wird diese Infrastruktur zusammen mit den Bausteinen entwickelt, die es ermöglichen, Auftragsabwicklungsprozesse in der gewünschten Weise abzubilden.

Der praktische Einsatz des Ansatzes wird in Kapitel 8 anhand von Beispielen gezeigt, wobei ein Anwendungsfall detailliert geschildert wird. Kapitel 9 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf weitere Einsatzmöglichkeiten des entwickelten Instrumentariums.

2. Auftragsabwicklung in der Automobilindustrie

Die vier wesentlichen Prozesse in der Automobilindustrie sind die Prozesse der Entwicklung, Versorgung, Auftragsabwicklung sowie Entsorgung (siehe Abbildung 2.1, vgl. [BW00, S. 7]). Die Auftragsabwicklung stellt aus Sicht des Kunden die eigentliche Leistungserstellung dar (vgl. [HC94, S. 52]) und ist auch aus Sicht des Unternehmens eine der Kernprozessketten (vgl. [Eve95, S. 37], [WS01, S. 17]), da sie eine Vielzahl von Schnittstellen mit dem Kunden aufweist. Verbesserungen in der Auftragsabwicklung eignen sich aus diesem Grund besonders zur Differenzierung vom Wettbewerb und dienen darüber hinaus der direkten Wertschöpfung [Wol95, S. 62]. Dies ist der Grund für die große Bedeutung der Gestaltung dieses Prozesses, da Verbesserungen erhebliche und direkt spürbare Wettbewerbsvorteile mit sich bringen können.

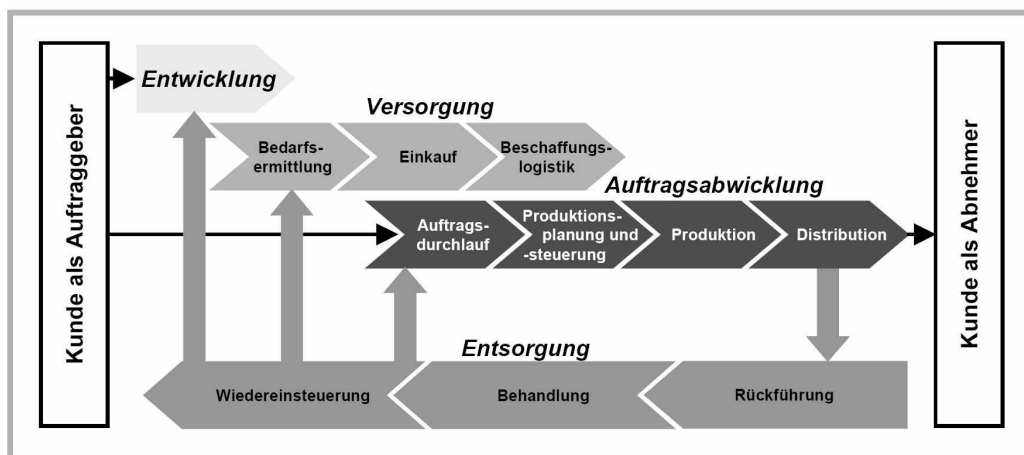


Abbildung 2.1.: Zentrale Unternehmensprozesse in der Automobilindustrie (Quelle: [BW00, S. 7])

Der Auftragsabwicklungsprozess beeinflusst in vielen Punkten den *Lieferservice*, der den Kunden des Unternehmens geboten werden kann. Der Lieferservice wird von PFOHL (vgl. [Pfo04, S. 36-38]) durch fünf Kriterien definiert (vgl. [Fle02, S. A 1-8]):

1. Lieferzeit als Zeit zwischen Auftragserteilung und Bereitstellung des Fahrzeugs.
2. Liefertreue als Maß für die Einhaltung des zugesagten Liefertermins.
3. Lieferbereitschaft als Maß für die Fähigkeit, einen Kundenauftrag sofort aus dem Lager zu erfüllen.

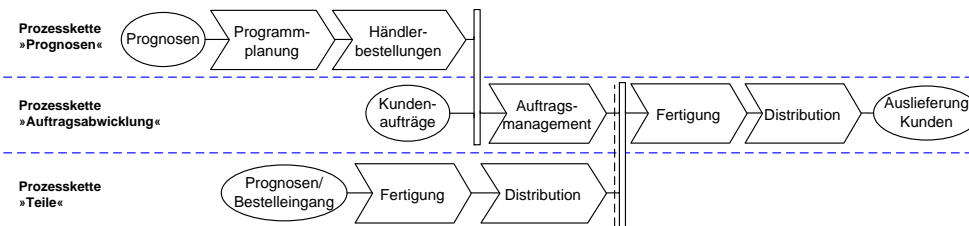


Abbildung 2.2.: Prozessketten im Bereich Auftragsabwicklungsprozess (vgl. z.B. [Gra01, S. 75], [Sta01b, S. 48])

4. Lieferqualität als Maß für die Übereinstimmung eines Fahrzeugs mit dem Kundenauftrag.
5. Lieferflexibilität als Maß für die Fähigkeit, auf Änderungswünsche des Kunden bzgl. Modell, Ausstattung oder Lieferort einzugehen.

Auf die Einflüsse des Auftragsabwicklungsprozesses auf die oben genannten Punkte wird im weiteren Verlauf des Kapitels eingegangen.

2.1. Auftragsabwicklung in der Automobilindustrie

Die Fertigung von Automobilen kann für viele Unternehmen als »Mass-Customization« – die Fertigung von individuell konfigurierten Produkten in großer Stückzahl – charakterisiert werden (vgl. z.B. [SSE02], [Kno02], [PS02, S. 773]). Insbesondere bei deutschen Herstellern des »Premiumsegments«¹ kann ein Kunde die Eigenschaften eines Fahrzeugs durch eine Auswahl aus einer sehr großen Anzahl von Optionen festlegen, aber auch für Fahrzeuge des »Massenmarktes« (z.B. Volkswagen Golf) sind die Auswahlmöglichkeiten immer noch sehr groß.

Durch diese Gestaltungsmöglichkeiten kann ein Kunde ein in vielen Bereichen² individuell gestaltetes Produkt (Fahrzeug) erwerben, wenn die Bereitschaft besteht, auf dieses Fahrzeug zu warten³. HOLWEG und PIL (vgl. [HP04, S. 44-46]) ermitteln in einer Studie über fünf Automobilhersteller eine mittlere Lieferzeit von 40 Tagen (vgl. auch [Bru04, S. 30]) bei einer Bandbreite von etwas über 20 Tagen im besten und nahezu 100 Tagen in schlechtesten Fall⁴.

¹Porsche, DaimlerChrysler, BMW, Audi (vgl. cite [S. 38] Fast04).

²Die Möglichkeiten des Kunden werden durch technische Restriktionen und Vertriebsentscheidungen begrenzt. So lassen sich, zumindest derzeit, mit keinem der Internet-Konfigurationswerkzeuge der deutschen Hersteller Fahrzeuge mit einem Lenkrad auf der rechten Seite konfigurieren, obwohl solche Fahrzeuge im Ausland sehr wohl konfiguriert werden können.

³Die von den Kunden gewünschte Lieferzeit für Automobile beträgt gemäß einer Studie der Universität Cardiff [Eli02, S. 14] für 59 % der Kunden bei bis zu zwei Wochen und für 81 % der Kunden bis zu drei Wochen, wobei 67 % der Kunden die Lieferzeit »wichtig« oder »sehr wichtig« finden. Einer Studie des Unternehmens CapGemini (vgl. [Gem05, S. 12]) zufolge liegt dieser Wert sogar bei 74 %, wobei allerdings der Punkt »Vehicle availability« sales/delivery date« erst auf Position sechs der wichtigsten Aspekte zu finden ist. Diese gewünschte Lieferzeit wird von einer großen Anzahl der auch auftragsbezogen fertigen Automobilhersteller nicht erreicht (vgl. [HP04, S. 44]).

⁴Mittelwerte für einen individuellen Hersteller, für einen kundenindividuellen Auftrag. Feuerer gibt für

Die Aufgabe des Auftragsabwicklungsprozesses aus Sicht des Kunden kann darin gesehen werden, innerhalb einer möglichst kurzen Zeitspanne⁵ (*Lieferzeit*) mit größtmöglicher Termingenaugkeit (*Liefertreue*) ein Fahrzeug zu liefern, welches den Wünschen bzgl. Modell und Konfiguration weitestgehend entspricht (*Lieferqualität*). Für Kunden, die Fahrzeuge sofort wünschen, ist es von Bedeutung, dass Fahrzeuge ab Lager verfügbar sind (*Lieferbereitschaft*). Kunden, die auch nach Bestellung eines Fahrzeugs Änderungen der Konfiguration wünschen, ist eine Möglichkeit hierfür bereitzustellen (*Lieferflexibilität*). Zusammengefasst hat der Auftragsabwicklungsprozess Einfluss auf *alle* Komponenten des Lieferservices.

Neben den oben formulierten Zielen für den Prozess aus Sicht der Kunden, können weitere Ziele aus Sicht der Automobilhersteller formuliert werden, wie z.B. die Sicherstellung einer gleichmäßigen Auslastung der Produktionswerke⁶ sowohl hinsichtlich der Anzahl als auch hinsichtlich der Eigenschaften der Produkte⁷ und die Vermeidung hoher Lagerbestände von Fahrzeugen, für die Abnehmer erst noch gefunden werden müssen.

Da die Durchlaufzeiten durch die Produktion zusammen mit den Wiederbeschaffungszeiten für viele Materialien im Automobilbereich länger sind, als die vom Kunden akzeptierte Lieferzeit, ist es i.d.R. erforderlich, einen sog. *Kundenentkopplungspunkt* zu definieren (vgl. [WLB⁺02, S. B 3-30], [Sch98a, S. 344]). In Abbildung 2.2 ist die Zusammenführung der kundenanonym – entkoppelt vom konkreten Bedarf der Märkte – von den Händlern bzw. dem Vertrieb bestellten Fahrzeuge und der Kundenaufträge zu sehen. In Abbildung 2.2 liegt der Kundenentkopplungspunkt *vor* der Produktion, dies ist jedoch nicht immer der Fall.

Die vielen Millionen möglicher Produktvarianten, die sich aufgrund der zahlreichen vom Kunden wählbaren Optionen ergeben⁸, legen es nahe, ein Fahrzeug erst dann zu fertigen, wenn der Kundenauftrag und somit die gewünschte Produktspezifikation bekannt ist. Nachteil eines solchen »Pull-Systems« ist offenbar, dass Marktschwankungen zu einer ungleichmäßigen Auslastung der Fertigungseinrichtungen des Herstellers führen und hohe Anforderungen im Hinblick auf Flexibilität an die Versorgung mit den benötigten Vorprodukten gerichtet werden. Der Kundenentkopplungspunkt in einem solchen System liegt vor der Bestellung von Fahrzeugen durch Händler bzw. Kunden. Das heißt, es werden lediglich Vormaterialien kundenanonym gefertigt, aber keine Fahrzeuge.

Der hohe Anteil von Fahrzeugen, die ab Händler-Lager verkauft werden⁹, legt nahe,

BMW eine minimale Durchlaufzeit (Bestellung bis Auslieferung Handel) für einen Auftrag von zehn Tagen an, für ein deutsches Werk, bei Lieferung innerhalb Deutschlands (vgl. [Feu02, S. 6]).

⁵ELIAS gibt an, dass z.B. 81 % der britischen Kunden auf ein Fahrzeug max. drei Wochen warten wollen (vgl. [Eli02, S. 3]), eine Zeitspanne, die von den Herstellern für kundenindividuell gefertigte Fahrzeuge häufig nicht erreicht wird (vgl. [HP04, S. 13]).

⁶HOLWEG und PIL geben an, dass ein zu 50 % ausgelastetes Werk in der Automobilindustrie 76 % der Kosten eines voll ausgelasteten Werkes verursacht (Personal, Maschinen, etc.) [HP04, S. 192].

⁷Fahrzeuge mit sehr hohem Fertigungsaufwand (z.B. Sonderfahrzeuge wie Krankenwagen) können nur mit einem begrenzten Anteil des möglichen Produktionsvolumens gefertigt werden.

⁸MEYR spricht im Fall von BMW von 10^{32} theoretisch baubaren Varianten (vgl. [Mey04b, S. 1], Feuerer nennt 10^{17} für die BMW 7er-Serie (vgl. [Feu02, S. 5]), HOLWEG gibt 3.993×10^9 Varianten für die Mercedes-Benz E-Klasse im Modelljahr 1999 an (vgl. [Hol00, S. 33]). Die Angaben zur Berechnung dieser Zahlen sind sehr vage, aber es kann vermutet werden, dass die Anzahl der möglichen Produktvarianten die oben angegebenen »vielen Millionen« nicht unterschreitet.

⁹HOLWEG und PIL geben allein für den US Markt eine Quote von 89 % an und für Deutschland 30 %

dass die meisten Automobilhersteller Fahrzeuge nach Prognosen fertigen und diese in den Markt »drücken« (Push-System) und damit die Freiheit gewinnen, die Volumina so zu steuern, dass die Fertigungseinrichtungen gleichmäßig ausgelastet werden können (vgl. [HP04, S. 12], [ADFS02, S. 3]). Der Kundenentkopplungspunkt liegt bei einem Push-System teilweise vor der Produktion (Abbildung 2.3 unten) oder aber erst danach (Abbildung 2.3 oben).

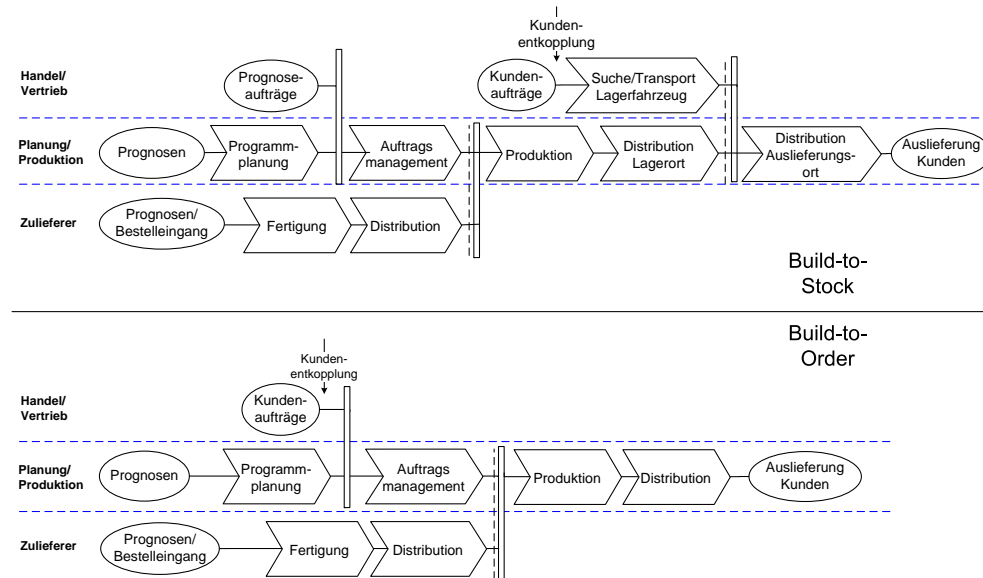


Abbildung 2.3.: Lagerorientierte (Build-To-Stock) vs. auftragsorientierte (Build-To-Order) Bestellabwicklung

Die Auswahl, ob ein Push- oder ein Pullorientiertes System vom Automobilhersteller umgesetzt wird, hat nachhaltigen Einfluss auf die Gestaltung des Auftragsabwicklungsprozesses. HOLWEG und JONES unterscheiden fünf Formen der Auftragsabwicklung. In ähnlicher Form wird diese Kategorisierung von Alkire et al unterstützt (vgl. [HJ00, S. 5], sowie [ADFS02, S. 4]¹⁰).

Loop	Order-to-Delivery-Approach	Description	Order-to-Delivery Time (UK data)
Loop 1	Dealer Stock	The car is bought from the stock at the visited dealer.	Instantly available
Loop 2	Dealer	The car is located at another dealer in the country, and transported to the dealer. The additional cost occurring is > £100 for dealer »swap« within the UK.	3 Days
Loop 3	Distribution Centre	The vehicle is sourced from a central stock location, controlled by the manufacturer. Generally the dealer does not hold any new cars in his own stock, so most sales would be made from the DC itself.	4 Days

(alle Angaben für den Zeitraum 1999-2000) [HP04, S. 12].

¹⁰ ALKIRE et al unterscheiden BTO, DTO (Deliver-To-Order), LTO (Locate-To-Order) und BTM (Build-To-Market-Forecast).

Loop4	Order Amendment	Orders are laid out as forecast in the first place, and once the customer specifies his order, these unsold »pipeline« orders are amended to customer requirements.	Variable, 11 days on average
Loop 5	Build-To-Order	This implies that the order is entered as a new order into the system. This happens only in 32 % of the new vehicle purchases in the UK at the moment, which an average order-to-delivery lead time of 48 days	40 - 60 days

Tabelle 2.1.: Order Fulfilment Loops (Quelle: [HJ00, S. 5])

HOLWEG und JONES bemerken mit Bezug auf Tabelle 2.1 (vgl. [HJ00, S. 6]):

... it should be noted that the term »build-to-order« is sometimes incorrectly used by manufactures to describe the order amendment function (loop 4), whereby forecast orders in the pipeline are amended to customer requirements.

Das angesprochene »order amendment«, die Anpassung eines bereits eingeplanten Händlernauftrags an einen Kundenwunsch, ist charakteristisch für die Festlegung von Fertigungsvolumina anhand von Prognosen und die anschließende Erzeugung von Aufträgen durch Händler in dem festgelegten Umfang. Das heißt, die Händler erzeugen die Aufträge nicht aufgrund konkreter Marktnachfrage, sondern aufgrund von »einvernehmlichen« Vereinbarungen mit dem Automobilhersteller und »versuchen dabei den noch nicht bekannten Geschmack des Kunden möglichst gut zu antizipieren« [Mey04b, S. 4]. Eine beim Händler eintreffende Kundenbestellung führt, wenn nicht aus dem Bestand bereits gefertigter Fahrzeuge befriedigt, zu einer Anpassung (»order amendment«) des bereits eingeplanten Auftrags an den Kundenwunsch (soweit noch möglich). Findet der Händler keinen Kunden für den Auftrag, wird das Fahrzeug in der vom Händler spezifizierten Form an diesen geliefert und geht, abhängig von der Vertriebsphilosophie, in den Besitz des Händlers über (vgl. [Mey04b, S. 4]).

2.2. Elemente des Auftragsabwicklungsprozesses

Der Auftragsabwicklungsprozess in der Automobilindustrie soll im Folgenden nicht beschränkt auf die Bearbeitung eines Kundenauftrags dargestellt werden, sondern in Zusammenhang mit den wichtigsten benachbarten Prozessen¹¹.

- ▷ Prozesskette »Prognosen«
- ▷ **Prozesskette »Auftragsabwicklung«**
- ▷ Prozesskette »Teile«

Die Prozesskette »Prognosen« setzt zum einen den Rahmen im Hinblick auf Kapazitäten, innerhalb dessen die Auftragsabwicklung arbeiten kann und ist zum anderen eine Quelle für Aufträge (Händlerbestellungen). Die Aufträge, die in der Prozesskette »Auftragsabwicklung« bearbeitet werden, stammen in Teilen von Händlern bzw. Vertriebseinheiten, die diese Aufträge auf der Basis von Prognosen erzeugen sowie von den Kunden, an deren Wünsche entweder ein Auftrag angepasst wird (Abbildung 2.2),

¹¹vgl. [Gra01, S. 91], [Sta01b, S. 47], [Mey04b, S. 2-8], [HP04, S. 23-47].

oder aber die ein möglicherweise abweichend spezifiziertes Lagerfahrzeug akzeptieren (Abbildung 2.3).

Die Flexibilität¹², mit der auf Veränderungen der Marktsituation reagiert werden kann, wird durch die Kapazitäten der Werke der Hersteller ebenso bestimmt, wie durch die Flexibilität der Zulieferer, die durch die Prozesskette »Teile« abgebildet werden (vgl. [BWL⁺03, S. 76-77]). In Bezug auf Abbildung 2.1 können die beiden benachbarten Prozessketten als »Versorgungsprozesse« aufgefasst werden.

Die unten dargestellten Abläufe in den Teilprozessen des Auftragsabwicklungsprozesses beschreiben diese im »gegenwärtigen Zustand« mit einem Fokus auf der europäischen Automobilindustrie, soweit sich dieser der Literatur entnehmen lässt und beschreiben den Zusammenhang zwischen den oben genannten drei Prozessketten.

Durch die hohe Bedeutung des Auftragsabwicklungsprozesses im Hinblick auf den Erfolg eines Unternehmens am Markt und die folglich hohe Motivation diesen wichtigen Prozess optimal zu gestalten, ist es kaum möglich, eine umfassende, konkrete und langfristig gültige Beschreibung von Auftragsabwicklungsprozessen in der Automobilindustrie zu geben. Das in Kapitel 6.1 präsentierte Modellierungskonzept abstrahiert viele der in diesem Kapitel dargestellten Details und erlaubt eine Beschreibung der übergreifend gültigen Aspekte von Auftragsabwicklungsprozessen.

Um die einzelnen Prozesselemente in der hier erforderlichen Weise beschreiben zu können, ist die abstrakte Prozesskette in Abbildung 2.2 nicht mehr ausreichend. Abbildung 2.4 zeigt eine detaillierte Darstellung der Prozesse der Auftragsabwicklung und der oben genannten benachbarten Prozesse. Die Struktur des Kapitels ist an diese Darstellung angelehnt und beschreibt die Prozesse im Fokus von »Prognosen« bis »Distribution« (siehe Abbildung 2.4).

2.2.1. Prozesskette »Prognosen«

Prognosen über das zukünftige Marktverhalten spielen eine wichtige Rolle für den Auftragsabwicklungsprozess. Zum einen dienen die Prognosen als Richtschnur für die Vorhaltung von Kapazitäten und zum anderen kann auf der Basis von Prognosen die Produktionsmenge festgelegt werden, zu einem Zeitpunkt, an dem nur wenige Kundenaufträge vorliegen¹³.

Prognoseerstellung und Volumenvereinbarung Handel

Um für diese vielfältigen Zwecke adäquate Informationen liefern zu können, werden sehr langfristige Prognosen mit einem zeitlichen Vorlauf von mehreren Jahren abgegeben, um z.B. den Zulieferern einer neuen Baureihe Informationen über die geplante Absatzmenge geben zu können und die Kapazitäten der Werke entsprechend auszurichten (vgl. [Mey04b, S. 12]). Diese langfristigen Prognosen werden im weiteren

¹²Voss zitiert den Flexibilitätsbegriff des ForLog-Forschungsverbundes als (vgl. [Voß06, S. 12]):

die Fähigkeit eines Systems (Unternehmen oder Netzwerk), selbstständig proaktiv oder reaktiv mit externen oder intern induzierten Veränderungen umzugehen, um die bisherigen Ziele zu erreichen.

¹³Aktueller Forschungsgegenstand sind Systeme, die eine Festlegung der Produktionsmenge ausschließlich auf Basis von Kundenaufträgen vornehmen, z.B. im EU-Projekt »ILIPT« (siehe Kapitel 3.1.1).

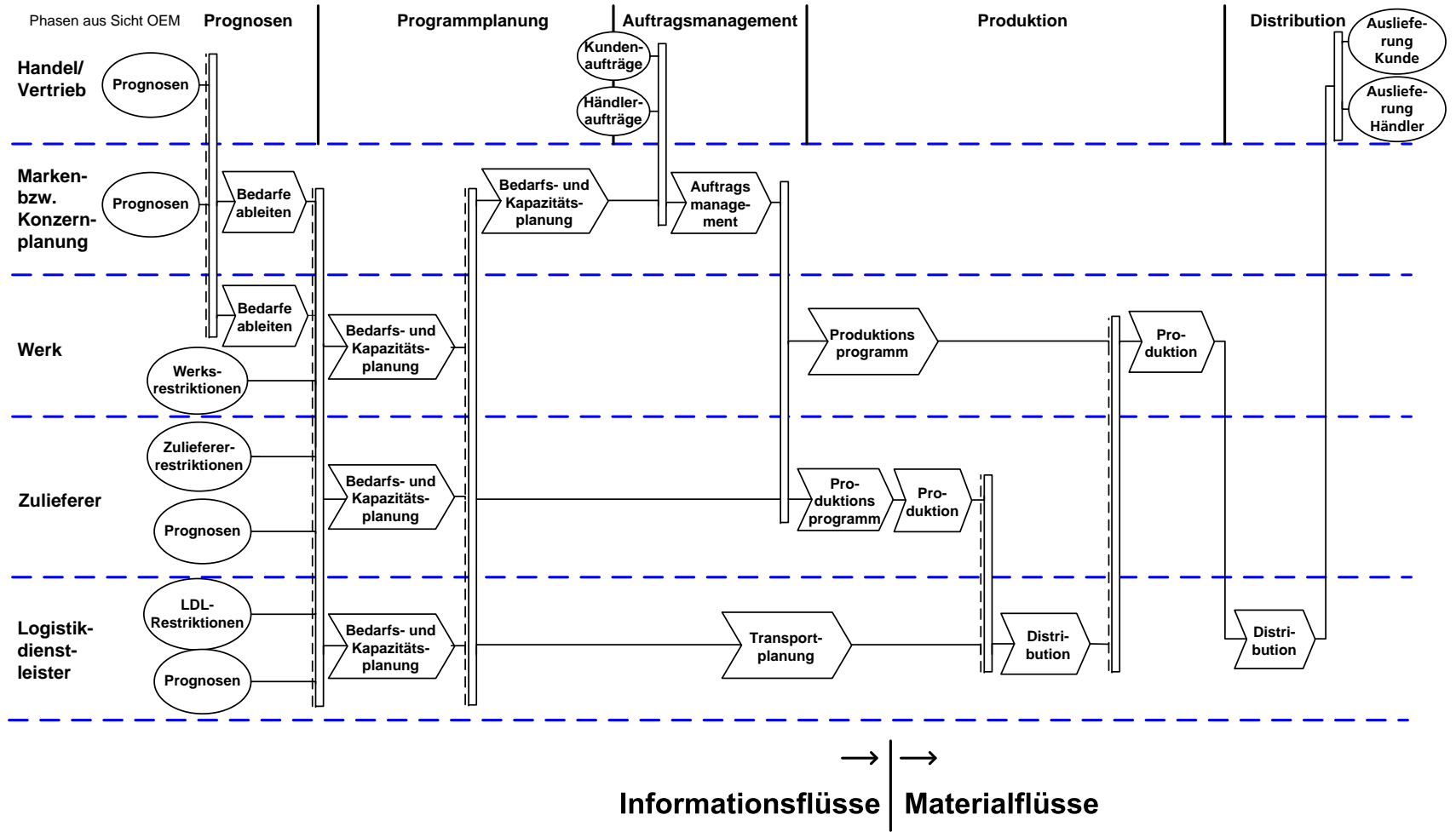


Abbildung 2.4.: Übersicht: Prozesse der Auftragsabwicklung und Umfeld

zeitlichen Verlauf immer weiter verfeinert, um zusätzliche bzw. neue Informationen über das Marktgeschehen reflektieren zu können (Jahresplanung, Programmplanung).

Die Prognosen werden in Zusammenarbeit von Vertrieb und Händlerschaft regelmäßig¹⁴ verfeinert und bilden zumeist die Basis für die Vereinbarung der Abnahmeverpflichtung¹⁵ der Händler (vgl. [Gra01, S. 144], [Mey04b, S. 5], [HP04, S. 23]). Die Abnahmeverpflichtung bezieht sich dabei auf Anzahlen von »Modellen« (siehe Abbildung 2.5), bzw. bei flexibleren Herstellern auf abstrakteren Ebenen der Produktbeschreibungshierarchie (z.B. »Karosserieformen« in dem Schema in Abbildung 2.5) (vgl. [Sta01b, S. 52]). Bei der Festlegung der Abnahmeverpflichtungen werden die zur Verfügung stehenden Kapazitäten in Bezug auf die Anzahl der Fahrzeuge und Eigenschaften, bei denen Engpässe befürchtet werden, bereits berücksichtigt (vgl. [Sta01b, S. 110], [Mey04b, S. 13]).

Die Abnahmeverpflichtungen werden von den Automobilherstellern verlangt, weil die jeweiligen Fertigungseinrichtungen gegenwärtig nur mit einer gewissen Mindestauslastung bei einem bestimmten Modell-Mix rentabel betrieben werden können¹⁶ (vgl. [Mey04b, S. 10]).

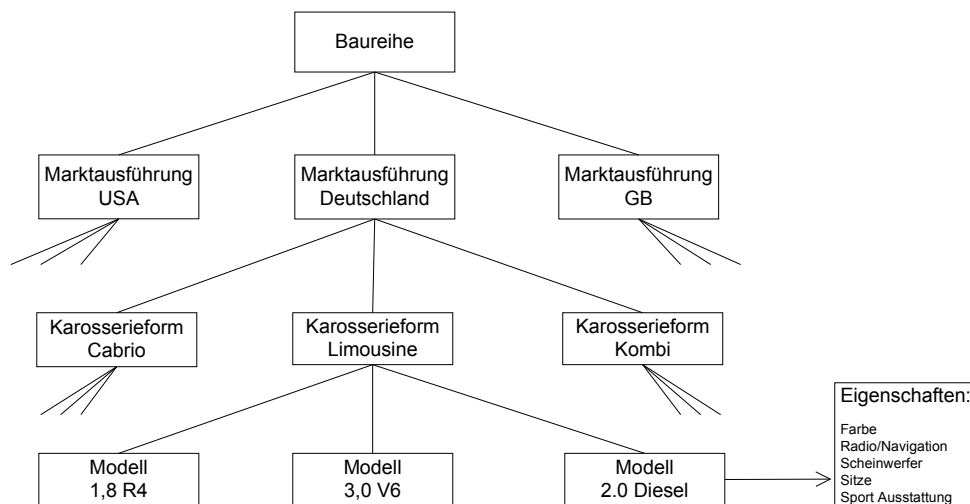


Abbildung 2.5.: Terminologie Produktaufbau (beispielhafte Zuordnung zu Ebenen): Baureihe, Marktausführung, Karosserieform, Modell, Eigenschaft

¹⁴HOLWEG spricht von einem oder zwei Monaten [HP04, S. 23].

¹⁵Bei stark nachgefragten Modellen hat diese sog. »Quote« eher den Charakter einer Beschränkung, da nicht nur die minimale, sondern auch die max. Abnahmemenge festgelegt wird (vgl. [Sta01b, 53]).

¹⁶Nach Angaben des Werksleiters, der Volkswagen AG in Wolfsburg, Dietmar Korzekwa, »haben wir ein Problem, die festen Kosten wirklich zu decken« bei einer Auslastung von zum Zeitpunkt des Interviews (10/2004) 66 % (vgl. http://autogramm.volkswagen.de/10_04/wolfsburg/1004_wolfsburg_titel01.htm).

Programmplanung und Kapazitätsmanagement

Die von den Vertriebseinheiten und Händlern erzeugten Prognosen bzw. Volumenvereinbarungen dienen in der Phase der *Programmplanung*¹⁷ als Eingangsdaten. Neben den Prognosen gehen weitere Informationen bzw. Zielgrößen in die Programmplanung ein (vgl. [HP04, S. 24]):

- ▷ **Kapazitäten:** Kapazitäten der Werke und Lieferanten für Fahrzeuge je Modell und wichtige Eigenschaften¹⁸.
- ▷ **Lagerbestände:** Die Bestände bereits gefertigter Fahrzeuge in den jeweiligen Märkten werden mit den Plan-Werten hierfür verglichen, um im Anschluss die Anzahl benötigter Einheiten zu berechnen, die erforderlich ist, um den prognostizierten Marktbedarf zu decken und den gewünschten Lagerbestand sicherzustellen.
- ▷ **Aufträge:** Bereits eingegangene Aufträge für die Periode, für die die Planung durchgeführt wird, werden berücksichtigt.
- ▷ **Deckungsbeiträge:** Die unterschiedliche Wettbewerbssituation in den einzelnen Märkten führt teilweise zu einer Anpassung der Preise für die Produkte in diesen Märkten und somit zu einer jeweils unterschiedlichen Profitabilität. Durch bevorzugte Versorgung von Märkten mit hoher Profitabilität kann eine Ertragsoptimierung vorgenommen werden.

Die Programmplanung erfolgt i.d.R. im monatlichen Rhythmus und wird typischerweise drei Monate im Voraus mit einem Planungshorizont von drei Monaten bis zu einem Jahr durchgeführt (vgl. [Mey04b, S. 13], [HP04, S. 24]). Ein wesentliches Ziel der Programmplanung ist es, für eine gleichmäßige Auslastung der Fertigungseinrichtungen zu sorgen und im Zuge des Kapazitätsmanagements die Verfügbarkeit benötigter Ressourcen¹⁹ sicherzustellen (vgl. [Gra01, S. 91]). Es werden also grundlegende Entscheidungen in Bezug auf zu fertigende Fahrzeugvolumina und bereitzustellende Kapazitäten bereits mehrere Monate vor der tatsächlichen Fertigung getroffen (vgl. [HP04, S. 25]).

2.2.2. Prozesskette »Auftragsabwicklung«

Parallel bzw. nachgelagert zur Programmplanung bzw. zum Kapazitätsmanagement findet das Auftragsmanagement statt (siehe Abbildung 2.2). Aufträge von Kunden bzw. von Händlern werden zunächst von der nationalen Vertriebsorganisation auf die Einhaltung der Händler-Quoten geprüft, im Anschluss vom Hersteller entgegengenommen und auf Vollständigkeit und Baubarkeit kontrolliert (vgl. [Wol95, S. 91], [HP04, S. 25]).

¹⁷Die *Programmplanung* beantwortet die Fragen (vgl. [Ohl00, S. 64]): Was soll produziert werden, wie viel soll produziert werden und wann soll produziert werden?

¹⁸Zum Beispiel Motoren, Getriebe und andere sog. »Heavy-Items«, die entweder starken Einfluss auf die technische Ausführung des Fahrzeugs haben (z.B. Allrad-Antrieb), oder aber in der Vergangenheit als Engpässe aufgefallen sind.

¹⁹Maschinen- und Personalkapazitäten, Fahrzeugkomponenten, Finanzmittel, etc.

Auftragsmanagement

Ein Auftrag muss den für den jeweiligen Markt geltenden Regeln entsprechen und darf somit nur ein Fahrzeug-Modell betreffen, welches auch in diesem Markt verfügbar ist und nur die hier freigegebenen Eigenschaften (Ausstattungen) beinhaltet²⁰. Neben der Übereinstimmung mit den Vertriebsregularien für den Markt und den technischen Restriktionen, muss der Auftrag zudem innerhalb des für diesen Markt und den jeweiligen Händler allokierten Volumens liegen (siehe Abbildung 2.6).

Das heißt, sollte ein Händler Fahrzeuge bestellen, die zu einer Überschreitung der maximalen Abnahmemenge (sofern für diesen Markt definiert) führen, werden Aufträge zu einem späteren Zeitpunkt eingeplant, für den der Händler eine noch nicht »verbrauchte« Mengenzuteilung besitzt (vgl. [Sta01a, S. 8], [Mey04a, S. 450], [HP04, S. 26]).

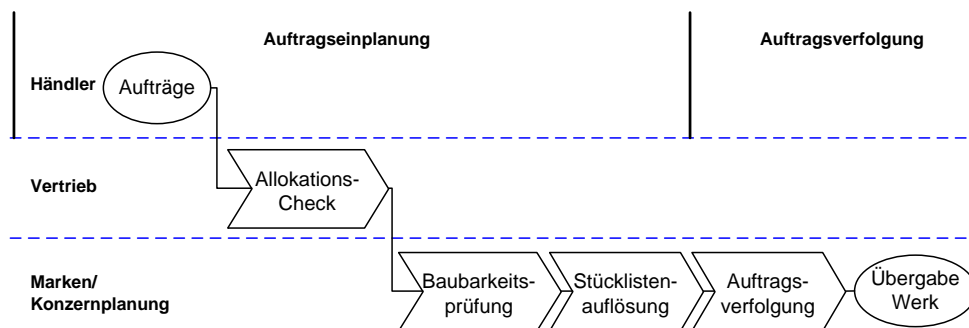


Abbildung 2.6.: Auftragsmanagement (vgl. [HP04, S. 26])

Der vom Kunden bzw. Händler spezifizierte Auftrag beschreibt die gewünschten Ausstattungsmerkmale in einer kundenorientierten Sicht. Wird z.B. eine verstärkte Lichtmaschine vom Kunden gewünscht, steht damit nicht notwendigerweise fest, welche Teile hierfür benötigt werden – es können zum einen mehrere Teile-Hersteller für eine Belieferung in Frage kommen und zum anderen kann z.B. die Halterung der Lichtmaschine davon abhängen, ob das Fahrzeug mit einer Klimaanlage ausgestattet ist oder nicht²¹.

Identisch spezifizierte Aufträge²² können auch abhängig vom Markt zu einer anderen Menge von benötigten Teilen führen, z.B. aufgrund spezieller Vorschriften für die Ausführung von Beleuchtungseinrichtungen oder spezifischer Abgasvorschriften. Der Prozess der Stücklistenauflösung erzeugt aus einer Auftragspezifikation eines Kunden bzw. eines Händlers eine Liste von Teilen, die der Automobilhersteller für die

²⁰Nicht alle für ein Modell vom Hersteller vorgesehenen Eigenschaften werden auf allen Märkten vertrieben, z.B. könnte eine Standheizung in sehr heißen Regionen nicht angeboten werden.

²¹Gebhardt beschreibt die Bestimmung der benötigten Teile für einen Auftrag (vgl. [GDM03, S. 2]):

... that a vehicle can be interpreted as a large set of installation points, each of which is characterized by a set of alternative build-in parts for the corresponding location. Which one of the alternative parts has to be chosen at an installation point, depends on its installation condition that can be specified by an item combination.

²²Identisch in Bezug auf die gewählten Zusatzausstattungen.

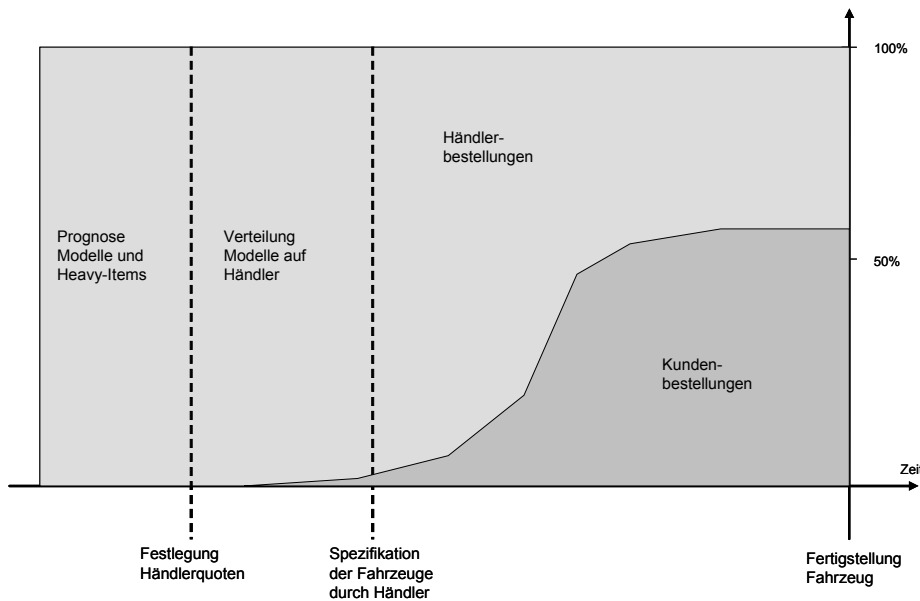


Abbildung 2.7.: Anteil Kundenaufträge (vgl. [Mey04b, S. 9])

Produktion benötigt. Weiterhin hängt die Stücklistenauflösung vom effektiven Produktionsdatum²³ ab, so dass im Falle einer Verschiebung des Produktionsdatums die Stücklistenauflösung jedenfalls neu durchzuführen ist.

Der verbuchte Auftrag kann innerhalb bestimmter Zeitfenster vom Kunden bzw. vom Händler in der Spezifikation angepasst werden (»order amendment«, siehe oben). Die Möglichkeiten der Anpassung sind allerdings abhängig von Änderungswunsch und Hersteller stark unterschiedlich. HOLWEG gibt die Änderungsmöglichkeiten für fünf (anonyme) Hersteller an (siehe Tabelle 2.2) und schließt die Änderung des Modells bzw. der Karosserieform durch Auftragsanpassung über alle Hersteller hinweg aus²⁴.

Eine Änderung des Modells aus Sicht des Kunden kann jedenfalls von einem Händler dadurch realisiert werden, dass der Händler einen anderen Auftrag aus seinem Auftragsbestand für das vom Kunden gewünschte Modell an die Kundenwunschspezifikation anpasst, selbst wenn eine solche Änderung vom Hersteller bzw. der nationalen Vertriebsorganisation bezogen auf einen konkreten Auftrag ausgeschlossen wird.

Manu- facturer	Engine	Options²⁵	Color	Wheels	Stereo
---------------------------	---------------	-----------------------------	--------------	---------------	---------------

²³ Änderungen der technischen Ausführung oder auch des Lieferanten können auch innerhalb des Produktionszeitraumes eines Fahrzeugs stattfinden und werden zu festgelegten Daten durchgeführt. Eine Verschiebung des Produktionsdatums kann dazu führen, dass ein Auftrag von einem Zeitpunkt »vor der Änderung« auf einen Zeitpunkt »nach der Änderung« verschoben wird.

²⁴ Stautner gibt für BMW eine Änderungsmöglichkeit auch des Modells an (vgl. [Sta01b, S. 111]), Feurer spricht von »Various levels of change flexibility for engines, national variants and steering within model groups and bandwidths« (vgl. [Feu02, S. 10]).

²⁵ air conditioning, sunroof, etc.

A	No amendment permitted	No amendment permitted	No amendment permitted	No amendment permitted	Amendment after assembly at dealership
B	60 days	15 days	6 days	6 days	6 days
C	26 days	26 days	19 days	19 days	Information not available
D	26 days body, 42 days engine	26 days	19 days	19 days	19 days
E	6 weeks	3 – 6 weeks	3 – 5 weeks	Amendment after assembly at distribution center	Amendment after assembly at distribution center

Tabelle 2.2.: Lead times for order amendment (days before start of production) for five manufacturers (Quelle: [HP04, S. 28])

FEURER gibt für BMW nach Einführung des »Kundenorientierten Vertriebs- und Produktionsprozesses« (KOV²⁶) an, dass Änderungen bzgl. Farbe, Eigenschaften und Sitzen und unter Einschränkungen des Motors, der Marktvariante und der Lenkanordnung²⁷, bis zu sechs Tage vor dem geplanten Produktionsende eines Auftrages möglich seien (vgl. [Feu02, S. 10]).

Die oben genannten Regelungen zur Auftragsanpassung haben einerseits erheblichen Einfluss auf die Materialdisposition²⁸ eines Herstellers, da hierdurch die Materialbedarfe kurzfristigen Schwankungen unterworfen werden, und bestimmen andererseits wesentlich die Flexibilität des Herstellers auf Änderungswünsche von Kunden und Händlern, im Hinblick auf die Eigenschaften der Fahrzeuge, zu reagieren²⁹.

Die Einplanung eines Auftrags erfolgt i.d.R. auf der Ebene von Wochen (vgl. [HP04, S. 29]). Eine Festlegung des Tages, an dem die Fertigung beginnen soll, erfolgt im nächsten Teilprozess der Auftragsabwicklung. Können mehrere Werke einen Auftrag fertigen, kann die Einplanung auf Wochen in einen Auftragspool zunächst »werkstanononym« erfolgen. In Folge dessen werden Kapazitäten nicht gegen die Ressourcen in den Werken geprüft, sondern gegen einen übergreifenden Ressourcenpool, der die Summe der verfügbaren Kapazitäten verwaltet.

DaimlerChrysler und BMW dagegen³⁰ ermitteln für einen Kundenauftrag bei der Einbuchung sofort ein Produktionswerk und einen Liefertermin (vgl. [Dai04, S. 2]),

²⁶Übergeordnete Ziele bei der Einführung des KOVP waren eine Senkung der Lieferzeit, so dass ein kundenspezifiziertes Fahrzeug mit einer Lieferzeit von zehn Tagen in bestimmten Regionen angeboten werden kann sowie eine möglichst große Anpassbarkeit eines Auftrags an Kundenwünsche (vgl. [Feu02, S. 9-10]).

²⁷Rechts- oder Linkslenker.

²⁸WITTHAUT und HELLINGRATH definieren *Disposition* als [WH03, S. 5]:

Aufgabe der Disposition ist die operative Planung und Steuerung von logistischen Systemen. Dies erfolgt durch eine Zuordnung von konkurrierenden Aktivitäten (Aufträgen) zu limitierten Ressourcen (Faktoren) unter Berücksichtigung von Randbedingungen und ausgerichtet auf Zielgrößen. Wichtiger Bestandteil der Zuordnung ist die Festlegung der Durchführungszeit (Start und Ende) der Aktivitäten.

²⁹BMW beziffert die Zahl der monatlichen Änderungswünsche mit 120.000 (vgl. [Bru04, S. 1], [Ott02, S. 1]).

³⁰Die Abweichungen der Auftragsabwicklung von DaimlerChrysler (Global Ordering) und BMW (KOV²⁶) gegenüber den von HOLWEG und PIL (vgl. [HP04], [Ren00, S. 23-33]) gegebenen Informationen legen den Schluss nahe, dass DaimlerChrysler und BMW sich nicht unter den untersuchten Unternehmen befanden).

[Sta01a, S. 6]) und sind somit in der Lage, einerseits eine Kapazitätsprüfung gegen die Kapazitätsrestriktionen dieses einen Werks durchzuführen und andererseits eine präzisere Lieferterminvorhersage zu machen, da mehr Informationen über die weitere Auftragsabwicklung zur Verfügung stehen.

MEYR unterscheidet bei der Ressourcenprüfung während der Auftragseinplanung zwischen »quota-available-to-promise« (QATP) und »capable-to-promise« (CTP³¹), wobei erstere Form der Prüfung gegen Kapazitätsinformationen erfolgt, die auf Basis von mittelfristig verfügbaren Prognosen abgegeben werden und z.B. werksübergreifend auf Wochenbasis vorliegen (vgl. [KS00, S. 179]). Es wird weder ein Werk bestimmt, in dem der Auftrag gefertigt wird (werksanonyme Einplanung), noch werden Restriktionen berücksichtigt, die in den jeweiligen Werken für die Sequenzbildung zu berücksichtigen sind (vgl. [Mey04a, S. 463]).

Der CTP-Ansatz hingegen führt zu einer Bestimmung sowohl eines Werkes als auch eines Produktionstages und sorgt für eine Einbuchung eines Auftrages unter Berücksichtigung der Restriktionen des jeweiligen Werkes. Um Kriterien im Hinblick auf die Gleichverteilung von Ausstattungsmerkmalen der Aufträge über die einzelnen Tage hinweg einhalten zu können, kann der CTP-Ansatz eine Umplanung von bereits gebuchten Aufträgen erfordern, wobei die Umplanung unter Beachtung etwaiger getroffener Lieferterminzusagen erfolgen sollte.

Fertigungsprogrammplanung und Sequenzierung

Um aus den auf eine Fertigungswoche festgelegten Aufträgen ein konkretes Fertigungsprogramm für die Werke abzuleiten, sind eine Reihe von Schritten notwendig (siehe Abbildung 2.8).

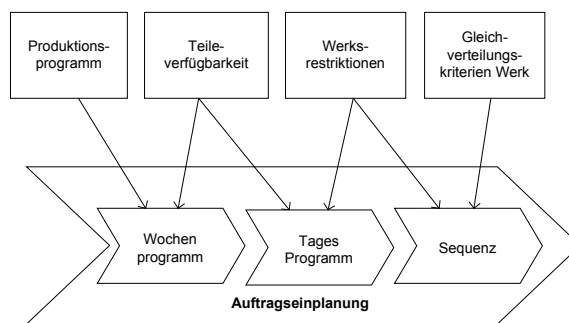


Abbildung 2.8.: Teilprozesse der Auftragseinplanung (vgl. [HP04, S. 29])

Liegen die Aufträge zunächst werksübergreifend auf der Ebene von »Wochenpaketen« vor³², muss zur Erstellung eines Fertigungsprogramms zunächst ein Werk ausge-

³¹KUHN und HELINGRATH definieren (vgl. [KH02, S. 148]):

- ▷ **Available-to-Promise (ATP)** Prüfung der Verfügbarkeit des gewünschten Produktes im Lagerbestand oder im Produktionsplan.
- ▷ **Capable-to-Promise (CTP)** Prüfung der Kapazitäten und Materialien zur Produktion des gewünschten Produktes.

Siehe dazu auch [Zei02, S. 20].

³²Auch wenn die Einplanung eines Auftrags direkt in ein Werk unter Bestimmung eines Fertigungstages

wählt werden und des Weiteren hier ein Tag innerhalb der Woche, an dem die jeweiligen Aufträge gefertigt werden sollen.

Für die Auswahl eines Werkes können zahlreiche Kriterien in Betracht gezogen werden. Kommen für die Fertigung der Aufträge nicht mehrere Werke in Betracht, beschränkt sich die Verteilungsaufgabe auf die Auswahl eines Tages aus der fraglichen Woche (vgl. [Gra01, S. 94], [HP04, S. 29]):

- ▷ **Grundsätzliche Eignung.** Nicht in allen Werken können stets alle Modelle einer Baureihe gefertigt werden.
- ▷ **Auslastung.** Die Einplanung eines zusätzlichen Auftrags sollte im Interesse einer gleichmäßigen Werksauslastung erfolgen insofern, dass sowohl die Anzahl der Aufträge an den jeweiligen Tagen möglichst wenig schwankt und dass die Teilebedarfe möglichst gleichmäßig verteilt sind (vgl. [Lem00, S. 43]).
- ▷ **Kapazitäten Engpassteile und Fertigungsrestriktionen.** Im Fall von Kapazitätsengpässen und in Folge nicht ausreichender Teileversorgung (Engpassteile), sollten die Aufträge so auf die Werke verteilt werden, dass die Engpasskapazität möglichst weitgehend ausgeschöpft wird (vgl. [HP04, S. 32]). Analog ist für die Fertigungsrestriktionen³³ der Werke zu verfahren.
- ▷ **Distributionskriterien.** Die Fertigung eines Fahrzeugs in einem Werk in der Nähe des Abnehmers kann die Distributionskosten³⁴ senken.
- ▷ **Sonstige Kriterien.** Neben den oben genannten Kriterien, die Auswirkungen auf die Kosten für die Fertigung eines Fahrzeugs haben, können z.B. niedrigere Lohnkosten den Ausschlag für eine Werksauswahl geben, auch wenn dadurch ein Standort eine deutlich höhere Werksauslastung erhält, als ein anderer. Weitere Faktoren können z.B. Betriebsvereinbarungen zwischen Arbeitgebern und -nehmern sein, dass ein Werk trotz höherer Kosten einen gewissen Anteil der Aufträge zugewiesen bekommt.

Zusätzliche Beschränkungen (sog. »Constraints«) bzgl. der Auftragsreihenfolge können Kostenvorteile mit sich bringen (z.B. durch die Bildung von »Farbblöcken«), sie führen auf der anderen Seite aber dazu, dass nicht notwendig aus den Aufträgen eines Tages³⁵ eine regelkonforme Sequenz gebildet werden kann, die gleichzeitig bezüglich der Kosten günstig ist, z.B. durch Ausschöpfung von Engpassressourcen. HOLLOWEG gibt an, dass ein Auftragsbestand von ca. fünf Tagen erforderlich ist, um aus den vorliegenden Aufträgen eine Sequenz für einen Produktionstag bilden zu können bei einer typischen Anzahl von Constraints (vgl. [HP04, S. 32]) und impliziert, dass durch die Sequenzbildung die Auftragsreihenfolge stark verändert wird mit der Folge, dass die auf Basis der jeweiligen Auftragsreihenfolge berechneten Teilebedarfe ebenfalls

erfolgt (siehe oben), muss ein Werk ausgewählt werden, es entfällt aber die Verteilung der Aufträge von Wochenpaketen auf Tage.

³³Beschränkungen des prozentualen Anteils bestimmter Karosserieformen, Fahrzeugmodelle oder Eigenschaften, wie z.B. »max. jedes dritte Fahrzeug mit Schiebedach«, »max. 30 % mit automatischem Getriebe«, »nicht mehr als 100 Fahrzeuge des Modells XY« usw..

³⁴Ein Schiffstransport kann z.B. kostengünstiger sein als ein LKW-Transport, auch wenn die Entfernung zum Zielort größer ist.

³⁵Unter der Annahme, dass vor der Sequenzbildung den Aufträgen ein Produktionstag zugeordnet wurde.

schwanken.

Durch das Konzept der »Perlenkette« wird die Veränderung der Auftragsreihenfolge kurz vor Produktionsbeginn vermieden. Die Auftragsreihenfolge wird als *Montagereihenfolge* geplant und z.B. 6 Tage (vgl. [WS01, S. 18]) vor Produktionsbeginn fixiert. Der Karosserierohbau beginnt mit der Fertigung der Karossen so, dass zu Beginn der Montage mit größtmöglicher Wahrscheinlichkeit die geplante Reihenfolge der Aufträge erreicht wird (vgl. [Mey04a, S. 460]).

Auch im Fall von Störungen der Teileversorgung kann die geplante Reihenfolge erhalten bleiben, wenn nur die durch die Störung nicht mehr baubaren Fahrzeuge aus der Sequenz entfernt werden³⁶. Das Produktionsvolumen kann z.B. durch »Vorziehen« von Aufträgen gehalten werden, wobei aber durch Beibehaltung der Auftragsreihenfolge die Effekte auf die Logistik berechenbar bleiben.

Die dadurch erreichte lange Phase der Stabilität der Auftragsreihenfolge wirkt sich positiv auf die Versorgungssicherheit aus, zusätzliche Maßnahmen, wie z.B. Sortierpuffer mit wahlfreiem Zugriff, und die späte Auftragszuordnung³⁷ erlauben es, die gewünschte Auftragsreihenfolge auch nach der Lackierung (weitgehend) wieder herzustellen (vgl. [How00, S. 19], [Mey04a, S. 460]).

Fahrzeugfertigung

In der sich anschließenden Fahrzeugfertigung wird die zuvor festgelegte Fertigungssequenz umgesetzt. Dabei lässt sich die Produktion im Automobilbau grob in drei aufeinander folgende Abschnitte teilen: den Rohbau, die Lackierung und die Montage (vgl. [Mey04b, S. 5]). Zwischen den Prozessabschnitten Rohbau und Lack, sowie Lack und Montage werden Sortierspeicher angeordnet, die mehrere hundert Karossen aufnehmen können (vgl. [How00, S. 9]), um eine für den jeweils folgenden Abschnitt optimierte Sequenz herstellen zu können (vgl. [HP04, S. 39]). Für die Lackiererei ist die Reduktion der Farbwechsel, also die Bildung von Sequenzen von Karossen, die in derselben Farbe lackiert werden sollen, ein nahe liegendes Optimierungsziel (vgl. [How00, S. 9]).

Die Optimierung der Sequenz für den Abschnitt Montage hat zum einen das Ziel, Taktverluste zu reduzieren, indem alle Montagestationen gleichmäßig ausgelastet werden³⁸, zum anderen ist z.B. im Fall der Steuerung nach dem »Perlenkettenprinzip« die

³⁶Eine Alternative kann darin bestehen, die Reihenfolge der Aufträge neu zu berechnen, um eine bezüglich der oben genannten Kriterien optimale Reihenfolge zu erhalten. Allerdings kann eine Veränderung der Reihenfolge dazu führen, dass Aufträge soweit »vorrücken«, dass die Teileversorgung aufgrund der verkürzten Vorlaufzeiten schwierig wird.

³⁷Am Ende der Lackierung wird der Kundenauftrag mit der nächsten Karosse verbunden, bei der Farbe und Karosserievariante mit dem Kundenwunsch übereinstimmen. Bei einer geringen Anzahl Rohbauvarianten kann so die »Verwirbelung« der Aufträge im Rohbau, aber insbesondere der Lackiererei, aufgefangen werden.

³⁸Nicht für alle montierten Teile muss Kapazität vorgehalten werden, um dieses Teil bzw. Modul in jedes Fahrzeug verbauen zu können. Wird z.B. im Schnitt jedes 20. Fahrzeug mit einer Standheizung ausgerüstet, ist eine weitaus geringere Kapazität erforderlich. Allerdings erzeugt die Reduktion der Montagekapazität für Standheizungen eine zusätzliche Restriktion: Fahrzeuge mit Standheizung dürfen nicht zu dicht aufeinander folgen. Ist es nicht möglich, eine Sequenz zu erzeugen, die keine solcher Restriktionen verletzt, kann z.B. die Ausbringung reduziert werden. Durch Freilassen von »Plätzen« in der Sequenz wird so eine Überschreitung der Restriktionen vermieden, auf der anderen Seite wird Montagekapazität »verschenkt«.

Reihenfolge herzustellen, die unter Umständen bereits vor mehreren Tagen an den Hersteller von JIS-Modulen genannt wurde.

Im Anschluss an die Montage wird die Einhaltung der Qualitätsmaßstäbe der Hersteller geprüft und soweit notwendig eine Nachbearbeitung durchgeführt. FRITS und HOLWEG geben eine empirisch ermittelte Rate von 59 % bis 71 % aller Aufträge an, die ohne Nacharbeit fertig gestellt werden können. Die Reihenfolge der fertig gestellten Fahrzeuge unterscheidet sich also substantziell von der Reihenfolge zu Beginn der Montage.

Distribution

Im Anschluss an die Fertigung gelangen die Fahrzeuge über unterschiedliche Distributionswege zu den Abnehmern. Die Transportmittel Schiff, Bahn und LKW werden abhängig von einer Reihe von (nicht voneinander unabhängigen) Gesichtspunkten ausgewählt:

- ▷ **Entfernung und geographische Lage.** Große Entfernung oder eine mangelnde Erreichbarkeit auf der Straße oder der Schiene sprechen für einen Transport per Schiff (z.B. Überseemärkte). Bei Erreichbarkeit auf dem Landwege stehen Bahn und LKW in Konkurrenz.
- ▷ **Zu transportierende Volumina.** Ein hohes Transportvolumen ist die Voraussetzung für die Nutzung von dedizierten Fahrzeugtransportern für den Schiffsverkehr³⁹, wenn ein Hersteller das Schiff allein nutzt. Die Anforderungen an die Volumina sinken, wenn ein solches Schiff von mehreren Herstellern genutzt wird. Für die Nutzung von Bahntransporten sind zwei Varianten zu unterscheiden: Die Einzelwaggonverladung sowie Ganzzüge. Einzelwaggons können auch bei geringen Volumina auf einer Transportrelation eingesetzt werden. Ganzzüge kommen nur in Frage, wenn ein solcher Zug in »angemessener Zeit« zusammengestellt werden kann. LKW-Transporte können ebenso wie Einzelwaggons auch bei sehr geringen Volumina eingesetzt werden.
- ▷ **Zeitliche Vorgaben.** Die tatsächliche Fahrzeit der Transportmittel Schiff, Bahn und LKW kann zueinander sehr unterschiedlich ausfallen. Ein Schiff, welches mit z.B. 16 Knoten kontinuierlich über einen längeren Zeitraum fährt, kann selbst LKW-Transporte überholen, die durch Ruhezeiten der Fahrer, Staus und bürokratischen Aktivitäten im grenzüberschreitenden Verkehr gebremst werden. Auch Bahntransporte können mit LKW-Transporten konkurrieren, wenn die Züge ohne Verzögerungen beladen werden und nicht auf dem Weg neu zusammengestellt werden. In Folge dessen kann die Auswahl eines Transportmittels unter Vorgaben für die maximale Fahrzeit nur unter Beachtung der Gegebenheiten der Transportrelation vorgenommen werden. Insbesondere im Fall von Schiffstransporten spielt die Wartezeit bis zum eigentlichen Transport eine wichtige Rolle.
- ▷ **Ökologische Erwägungen.** Insbesondere Automobilhersteller stehen oft als »Umweltsünder« im Licht der Öffentlichkeit. Um möglicherweise verkaufsschädigende

³⁹Fahrzeuge können auch in Containern verladen auf einem Containerschiff zusammen mit anderen Gütern transportiert werden. Schiffe, die ausschließlich Fahrzeuge transportieren, können nur wirtschaftlich eingesetzt werden, wenn eine gewisse Mindestauslastung vorliegt.

Imageschäden zu verhindern, kann die Auswahl eines Transportmittels unter ökologischen Gesichtspunkten vorgenommen werden (z.B. Bahn oder Binnenschiff anstatt LKW).

- ▷ **Kosten.** Alle oben genannten Kriterien für die Auswahl eines Transportmittels können letztlich unter dem Kostenaspekt subsummiert werden. Verärgerte Kunden können ebenso unter Kostenaspekten betrachtet werden, wie ein Imageschaden, der durch die Nutzung von LKW anstatt der Bahn als Transportmittel entsteht. Leichter zu messen sind hingegen die Kosten, die für den eigentlichen Fahrzeugtransport zu entrichten sind. Auch im Fall der Kostenbetrachtung ist keines der erwähnten Transportmittel per se überlegen. Ein Transport auf einem voll ausgelasteten Schiff ist günstiger als auf einem LKW, ein nur gering befülltes Schiff hingegen teurer. Auch hier kann die Auswahl eines Transportmittels nur unter Beachtung der Relationsspezifika erfolgen.

Nach der Qualitätskontrolle am Ende der Fahrzeugfertigung werden die Fahrzeuge auf die für die Zielrelation festgelegten Transportmittel verladen. Das Ziel dieses ersten Transportes hängt von vielen Faktoren ab. Fahrzeuge, die bereits an einen Endkunden verkauft wurden, können in Richtung des Abnahmeortes bewegt werden. Fahrzeuge, die vom Kunden aus einem Distributionszentrum in Werksnähe abgeholt werden, müssen bis zum Abholzeitpunkt eingelagert werden. Fahrzeuge, für die noch kein Kunde gefunden werden konnte, können zum bestellenden Händler bewegt werden, aber auch zunächst in einem zentral gelegenen Lager abgestellt werden.

Während die Ziele der Automobilunternehmen im Fall von Fahrzeugen, die bereits einen Kunden gefunden haben, leicht zu beschreiben sind, ist die Distribution von »Lagerfahrzeugen« komplexer. Fahrzeuge mit einem Endabnehmer müssen unter Beachtung oben genannter Kriterien (Termine, Kosten, Zeiten) an das jeweilige Ziel transportiert werden. Fahrzeuge ohne bekannten Endabnehmer wurden zwar i.d.R. von einem Händler bestellt⁴⁰, aber ein Transport zum bestellenden Händler ist nicht unbedingt kostenoptimal.

Die Attraktivität bereits gefertigter Fahrzeuge für den Kunden ist geringer, als die individuell bestellter Fahrzeuge, vor allem weil die Fahrzeugkonfiguration bereits festliegt. Der Kunde kann nur noch geringen Einfluss auf Aussehen und Ausstattung der Fahrzeuge nehmen⁴¹. Trotz der hohen Anzahl von Konfigurationsmöglichkeiten für einen Auftrag, kann für eine gewisse Anzahl von Kunden gewünschter Fahrzeugspezifikationen in der Gesamtzahl der Lagerfahrzeuge eines gefunden werden, welches dem Kundenwunsch so nahe kommt, dass dieser bereit wäre, das Fahrzeug unter Umständen mit einem Preisabschlag zu akzeptieren. Das zugrunde liegende Szenario: Händler A besitze ein Fahrzeug, welches von einem Kunden bei Händler B akzeptiert würde, führte nicht notwendig dazu, dass das Fahrzeug von Händler A an den fraglichen Kunden transferiert würde.

Es liegt im Interesse von Händler A ein Lagerfahrzeug zu verkaufen, aber nur wenn Händler B seinerseits einen Vorteil an einem solchen Vorgang hat, wird es zu einem

⁴⁰Zum Beispiel um die Abnahmeverpflichtung zu erfüllen (siehe oben). Eine Alternative hierzu ist, dass die Fahrzeuge von einer zentralen Vertriebsseinheit bestellt wurden und von den Händlern im Bedarfsfall abgenommen werden.

⁴¹Möglich sind z.B. Veränderungen im Bereich der Felgen oder der Einbau eines Radios. Also Veränderungen, die ein Händler auch nach der Fertigung des Fahrzeuges selbsttätig vornehmen kann.

Abschluss kommen. Vorbedingung für die Transaktion ist, dass der Händler B überhaupt Kenntnis von dem »passenden« Fahrzeug bei Händler A erlangt. Das große Interesse der Automobilhersteller, Lagerfahrzeugbestände niedrig zu halten (siehe Kapitel 3.1.1), führt zur Festlegung von Prozessen, die einen Lagerfahrzeugtransfer zu einem Händler mit einem Kunden für dieses Fahrzeug für alle Beteiligten attraktiv macht (»Locating«). Enabler für diese Prozesse sind IT-Systeme, die es den Händlern erlauben, aus dem Pool von Lagerfahrzeugen (»Locating-Pool«) ein möglichst »passendes« herauszufiltern, welches zu möglichst geringen Kosten zum abnehmenden Händler transportiert werden kann.

Im Fall des US Marktes ist für nahezu 90 % der Fahrzeuge der Endabnehmer noch nicht bekannt, während es im deutschen Markt immerhin noch ca. 40 % des Gesamtvolumens sind (vgl. [HP04, S. 12]). Die Extrempunkte von Distributionsstrategien für Lagerfahrzeuge in einem Markt⁴² sind die Lagerung an einem zentralen Ort (minimal verteilt) und die Lagerung der Fahrzeuge bei den Händlern (maximal verteilt). In der Praxis werden Mischformen dieser beiden Konzepte genutzt.

Auf der einen Seite ist ein gewisser Bestand von Fahrzeugen in den Ausstellungsräumen der Händler (»Showroom-Fahrzeuge«) offenbar notwendig, auf der anderen Seite sinken leicht erkennbar die Distributionskosten, wenn die Fahrzeuge erst dann in die Fläche verteilt werden, wenn der Endabnehmer bereits bekannt ist⁴³. Fahrzeuge ohne Endabnehmer werden also entweder direkt zum bestellenden Händler transportiert oder in einem Fahrzeugpool gelagert, bis ein Endabnehmer vorhanden ist.

2.2.3. Prozesskette »Teile«

Den Lieferanten der Automobilhersteller werden mit langem zeitlichem Vorlauf bereits Informationen über die voraussichtlichen Abnahmemengen mitgeteilt. Diese Prognosen werden basierend auf der Jahresplanung und im Folgenden auf dem Produktionsprogramm erzeugt (vgl. [S. 14], [HP04, S. 34]). HOLWEG und PIL geben einen Horizont von 12 Monaten für diese Prognosen an. Weiterhin stellen die Automobilhersteller auf Wochenbasis Informationen für die nächsten 6-10 Wochen zur Verfügung⁴⁴.

Diese letzteren Informationen sind genauer⁴⁵ als die langfristigen Prognosen, enthalten aber immer noch einige Unsicherheit, da zum einen noch nicht notwendigerweise alle Aufträge für diesen Zeitraum vorliegen und zum anderen die Zuordnung der Aufträge zu Tagen noch nicht stattgefunden hat (vgl. [HP04, S. 34]). Ausnahmen sind Hersteller, die den Aufträgen sofort einen Produktionstag zuweisen. Bei diesen verbleibt aber ebenfalls das Moment der Unsicherheit, welches aus den noch nicht eingegangenen Aufträgen für diesen Zeitraum herrührt.

⁴²Lagerfahrzeuge können auch marktübergreifend verteilt werden, wenn Landesspezifika (z.B. die Bedienungsanleitung, Sprache der Meldungen des Navigationssystems) erst nach bekannt werden des Endabnehmers festgelegt werden.

⁴³1999 werden in Großbritannien 15 % aller Fahrzeuge von einem lagernden Händler zu dem Händler gefahren, bei dem das Fahrzeug verkauft wurde (vgl. [Hol01, S. 6] »Sales via dealer transfer«). 1992 waren es noch 45 %.

⁴⁴Volkswagen z.B. prognostiziert in einem *Lieferabruf* (s.u.) den Bedarf für sechs Monate und übermittelt für 11 Wochen Einzelabrufe unter Angabe des Belieferungstages (Eintrefftermin im Werk) (vgl. [VA02, S. 15]). Nach Einführung eines neuen Systems zur Prognose von Teilebedarfen (EPL) kann allerdings eine Vorschau von 24 Monaten erzeugt werden (vgl. [GDM03, S. 2]).

⁴⁵Genauer in dem Sinn, dass die tatsächlich benötigte Teilemenge besser prognostiziert wird.

Für die Übermittlung sowohl der Prognosen als auch der tatsächlichen Bedarfsmengen stehen genormte Austauschformate zur Verfügung, wie z.B. die Normen 4905 (Lieferabruf), 4915 (Feinabruf) und 4916 (produktionssynchroner Abruf) des Verbandes der Automobilindustrie VDA oder ODETTE/EDIFACT Standards, die den Transfer der benötigten Daten erlauben (für die Hersteller Volkswagen bzw. MAN vgl. [VA02, S. 8], [Gru05, S. 8]). Welcher dieser Standards zum Einsatz kommt, wird abhängig von den Teilen bzw. Lieferanten ausgewählt. Volkswagen z.B. versendet Lieferabrufe im Minimum alle vier Wochen. Im dazwischen liegenden Zeitraum nur dann, wenn sich relevante Änderungen in Bezug auf Bedarfsmengen oder z.B. Abladestellen ergeben haben (vgl. [VA02, S. 17]). Lieferabrufe als Kommunikationsmittel eignen sich z.B. für Teile, die losgrößenorientiert gefertigt und über ein Lager versorgt werden (vgl. [Gra00, S. 7]).

Teile mit geringer Varianz und hohem Volumen, die zeitnah produziert und mit geringem oder ohne Lager versorgt werden, können über Feinabrufe gesteuert werden, die z.B. (Volkswagen) täglich versandt werden (vgl. [VA02, S. 18], [Gra00, S. 8]). Für Teile mit hoher Varianz, die erst wenige Stunden vor dem Bedarfszeitpunkt sequenz- und zeitpunktgenau bestellt werden, kann der produktionssynchrone Abruf genutzt werden.

Im engen Zusammenhang mit der Klassifizierung der Teile für die Auswahl eines geeigneten Abrufes ist die *Belieferungsform* zu sehen. Graf (vgl. [Gra00, S. 7]) unterscheidet (für DaimlerChrysler) drei Belieferungsformen bzw. -ketten (ähnlich bei Volkswagen vgl.[VA02, S. 18-19]):

1. **Just-in-Sequence-Kette.** Komplexe kundenindividuelle Module und Teileumfänge, sequenz- und zeitpunktgenaue Anlieferung. Die für die jeweiligen Fahrzeuge auftragsbezogen gefertigten Teile bzw. Module werden vom Lieferanten bzw. Logistikdienstleister ohne Zwischenlagerung direkt an den Verbrauchsort geliefert.
2. **Lagerkette.** Teile, die nur in Losgrößen gefertigt und deren Bedarf nur schwer prognostiziert werden kann. Diese Teile werden in geeigneten Losen gefertigt und in solchen Beständen gelagert, so dass die gewünschte Versorgungssicherheit gewährleistet werden kann. Die Anlieferung kann z.B. durch *Gebietsspediteure* erfolgen, die Lieferanten regional bündeln (vgl. [Sch05, S. 10]).
3. **Direktkette.** Teile mit geringer Varianz und hohem Volumen, die zeitnah produziert und lagerlos versorgt werden.

Abhängig von der Informationsanbindung und Belieferungsform kann der Zulieferer seine Produktionsplanung so ausrichten, dass einerseits die von den Automobilherstellern gewünschten Liefertermine erreicht und andererseits eine optimale Nutzung der internen Kapazitäten unter den gegebenen Randbedingungen erreicht werden kann.

2.3. Kennzahlen und Ziele bei der Auftragsabwicklung

Um einen komplexen Prozess, wie die Auftragsabwicklung effizient steuern zu können, ist die Aggregation von Zustandsinformationen zu Kennzahlen⁴⁶ ein effizientes Werkzeug. So liefert z.B. die Angabe der Entwicklung der durchschnittlichen *Lieferzeit* in Kürze entscheidende Hinweise, ob die Unternehmensprozesse in der gewünschten Weise »funktionieren«, da viele Probleme in den Teilprozessen zu Verzögerungen in der Produktauslieferung führen können. Häufig werden Kennzahlen genutzt, um strategische Ziele der Unternehmensentwicklung vorzugeben. Beispiel hierfür sind publizierte Ziele für Lieferzeiten für verschiedene Automobilhersteller:

- ▷ Volkswagen 14 Tage [Har96, S. 33]
- ▷ Toyota 14 Tage [HP04, S. 13]
- ▷ Nissan 15 Tage [HP04, S. 13]
- ▷ BMW 10 Tage⁴⁷ [Ren04, S. 4]

Selten wird jedoch hinzugefügt, wie diese Lieferzeiten zu interpretieren sind, bzw. wie diese gemessen werden sollen. Beispiele für den Interpretationsspielraum in Bezug auf die Lieferzeit sind:

- ▷ **Markt?** Auf welchen Markt bezieht sich die Angabe? Eine mögliche Einschränkung: Auf einen Kunden, der bei einem deutschen Händler ein Fahrzeug bestellt, bei Auslieferung in Deutschland.
- ▷ **Mittelwert, Minimalwert?** Ist die Lieferzeit ein Mittelwert über alle Aufträge oder ein Minimalwert, der nur für einen kleinen Teil der Aufträge realisierbar ist?
- ▷ **Lagerfahrzeug, Auftragsfertigung?** Gehen in die Ermittlung einer mittleren Lieferzeit auch Lagerverkäufe ein (mit einer Lieferzeit von Null)?
- ▷ **Auslieferung Händler oder Kunde?** Gilt ein Fahrzeug als ausgeliefert, wenn es beim Händler eintrifft und potenziell vom Kunden übernommen werden kann, oder wird die tatsächliche Abholung durch den Kunden gewertet?

Es ist nahe liegend, dass aus identischen Grunddaten eine große Menge mehr oder weniger plausible Angaben über Lieferzeiten abgeleitet werden können und dass dieser Freiheitsgrad »kreativ« genutzt werden kann, um eine bestimmte, gewünschte Aussage über die Lieferzeit zu machen oder um unangenehme Sachverhalte zu verschleiern. Eine sinnvolle Interpretation von Kennzahlen setzt also stets voraus, dass Angaben über die Berechnung dieser Zahlen vorliegen.

⁴⁶Das »Gabler-Wirtschaftsinformatik-Lexikon« definiert:

Kennzahlen *informieren rückwirkend über quantifizierbare, betriebswirtschaftlich interessierende Sachverhalte oder legen diese vorausschauend fest* (vgl. [SGR97, S. 365]).

⁴⁷Renner gibt 10 Tage minimale »Prozessdurchlaufzeit« (eine Def. wird nicht gegeben) für Auslieferung innerhalb Deutschlands aus einem deutschen Werk als Ziel der Durchlaufzeitverkürzung bei Einführung von KOVP an.

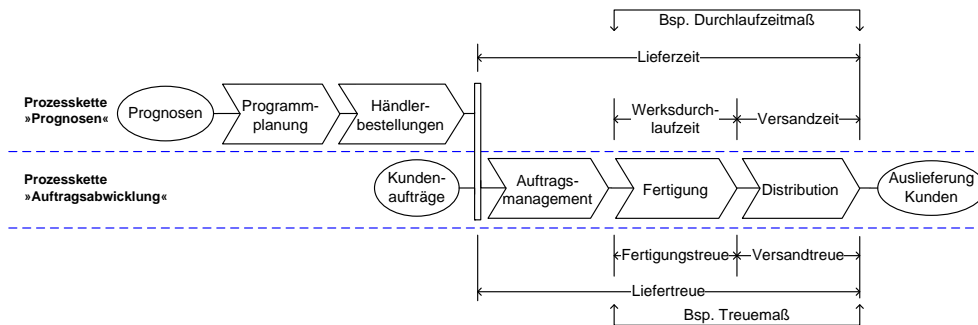


Abbildung 2.9.: Kennzahlen im Auftragsabwicklungsprozess

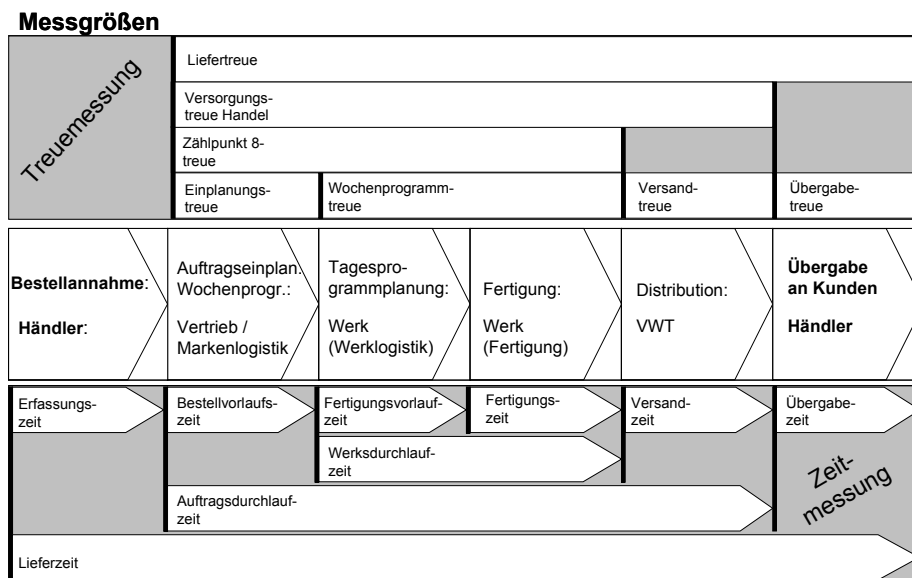


Abbildung 2.10.: Kennzahlen im Auftragsabwicklungsprozess am Beispiel Volkswagen (Quelle: [Hoo04, S. 16])

Ein konkretes Beispiel für die Definition von Kennzahlen bei einem Automobilhersteller (Volkswagen) wird in Abbildung 2.10 (vgl. [Hoo04, S. 16]) gegeben. In den oben diskutierten *Lieferservice* gehen unmittelbar die Kennzahlen *Lieferzeit* und *Liefertreue* ein. Abbildung 2.10 zeigt die Berechnungsvorschrift für verschiedene Zeit- bzw. Treuemessgrößen. Für eine effektive Berechnung der Kennzahlen zur Analyse des Auftragsabwicklungsprozesses sind diese Definitionen aber nicht hinreichend. Werden z.B. Verkäufe von Lagerfahrzeugen in die Berechnung aufgenommen, führen die hieraus resultierenden sehr kurzen Lieferzeiten – insbesondere wenn das Fahrzeug beim Händler bereits vor Ort ist – und die sehr hohe Liefertreue zu einem sehr positiven Bild, abhängig vom Anteil der Lagerfahrzeugverkäufe am Gesamtabsatz.

Die Kennzahlen aus Abbildung 2.10 können in die vier Kategorien des im ILIPT-Projekt entwickelten »KPI Framework Model« (siehe Abbildung 2.11) eingebettet

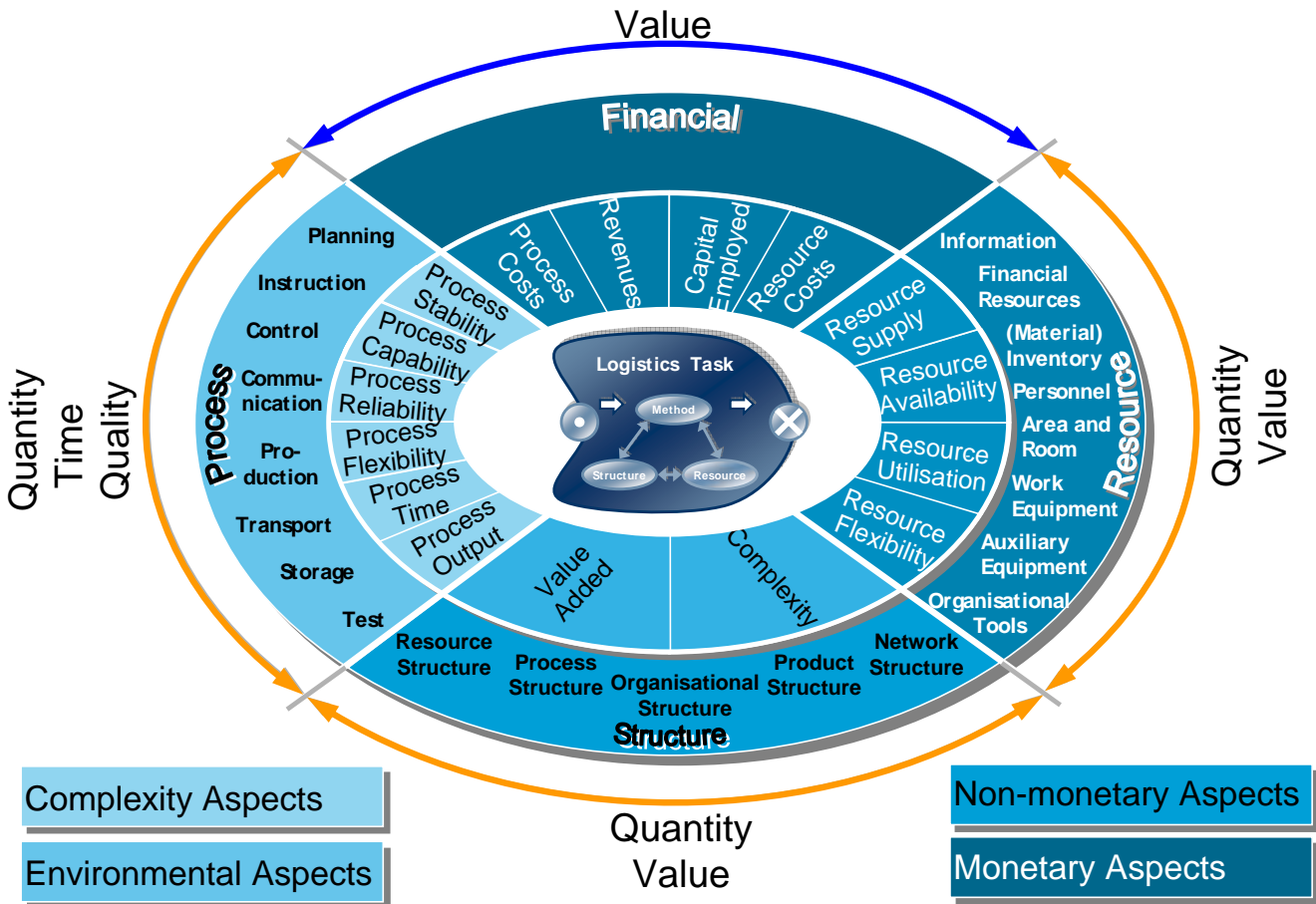


Abbildung 2.11.: Kennzahlenkategorien (Quelle: [Kel06, S. 16])

werden:

- ▷ **Prozessorientierte Bewertung.** Die Zeitmessgrößen aus Abbildung 2.10 fallen in die Kategorie »Process Time«, die Treuemessgrößen in die Kategorie »Process Reliability« (siehe [Kel06, S. 102-103]).

- ▷ **Finanzorientierte Bewertung.** Obwohl die Abbildung 2.10 Kennzahlen dieser Kategorie nicht repräsentiert, sind Kosten bei der Bewertung von Prozessalternativen eine Zielgröße, deren Wichtigkeit nicht hervorgehoben werden muss.
- ▷ **Ressourcenorientierte Bewertung.** Kennzahlen dieser Kategorie weisen z.B. auf Kapazitätsengpässe hin, die wiederum direkte Auswirkungen auf Kennzahlen wie Lieferzeit, Liefertreue und auch Kosten haben.
- ▷ **Strukturorientierte Bewertung.** Die Komplexität als beispielhafte Kategorie im Bereich der strukturorientierten Kennzahlen kann z.B. als Produktvariabilität (siehe Kapitel 1.1) aus dem Produktaufbau abgeleitet werden.

Selbst wenn Ziele anderer Kategorien im Vordergrund zu stehen scheinen, z.B. beim Ziel der Verkürzung der Lieferzeit, so wird auch dieses Ziel wahrscheinlich vor dem Hintergrund formuliert, dass über eine solche Verbesserung des Lieferservices neue Kunden gewonnen werden, bzw. eine Abwanderung von Kunden verhindert werden soll.

Neben einem möglichst guten Lieferservice den Kunden des Unternehmens gegenüber, ist also die Realisierung einer vorteilhaften Kostenstruktur ein wichtiges Ziel bei der Gestaltung von Auftragsabwicklungsprozessen.

3. Aufgabenstellungen im Auftragsabwicklungsprozess

In den folgenden Kapiteln 3.1, 3.2 und 3.3 werden Beispiele für mögliche Anwendungen eines Ansatzes zur Gestaltung und Bewertung von Auftragsabwicklungsprozessen in der Automobilindustrie gegeben. Diese Anwendungsbeispiele sind in drei Kategorien gegliedert: Der strategische, taktische und operative Einsatz bildet eine Basis für die Ableitung von Anforderungen an einen Modellierungs- und Bewertungsansatz. Zu jeder dieser drei Kategorien werden zwei Beispiele angeführt.

An die Darstellung der Fragestellung, die den Beispielen jeweils zugrundeliegt, schließt sich die Skizze einer Lösungsstrategie, die zu einer Beantwortung der Fragestellung führen kann, an. Auf das Unterkapitel »Lösungsstrategie« folgen wiederum jeweils Anforderungen an einen Modellierungs- und Bewertungsansatz, die sich ergeben, wenn die Lösungsstrategie umgesetzt werden soll. Die Grundidee dieser Lösungsstrategien ist stets, dass ein *Modell* von einem Untersuchungsgegenstand erstellt wird, an welchem dann Fragen beantwortet werden können.

Der Begriff »Modell« wird von STACHOWIAK durch drei Merkmale charakterisiert (vgl. [Sta73, S. 131-133]):

Abbildungsmerkmal Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.

Verkürzungsmerkmal Modelle erfassen im Allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant scheinen.

Pragmatisches Merkmal Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für bestimmte – erkennende und/oder handelnde, modellbenutzende – Subjekte, b) innerhalb bestimmter Zeitintervalle und c) unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen.

Um Fragen im Kontext einer Aufgabenstellung anhand eines Modells beantworten zu können, müssen die hierfür relevanten Attribute (siehe obige Definition) im Modell abgebildet sein. Die Unterkapitel mit dem Titel »Lösungsvoraussetzungen« skizzieren (unter anderem), welche Attribute zur Beantwortung der jeweiligen Fragen in den Modellen repräsentiert sein sollten. Anstatt des Begriffes »Original« aus obiger Definition, werden hier auch die Begriffe »System«¹ und »Real-System« verwendet.

¹System ist ein zentraler Begriff der Kybernetik und wird dort als »ein aus mehreren Teilen zusammengesetztes, gegliedertes Ganzes verstanden, dessen Teile strukturell oder funktional miteinander in Beziehung stehen« (vgl. [Bro02]).

3.1. Einsatz im strategischen Bereich

Kennzeichnend für Aufgaben im strategischen Bereich sind zum einen die langfristige Orientierung der Fragestellungen und zum anderen die abstrakte Betrachtungsweise. Das heißt, kurzfristig zu bearbeitende Problemstellungen aus dem operativen Betrieb eines Unternehmens werden nicht im Rahmen strategischer Erwägungen gelöst. Ziel ist es vielmehr, die langfristige Wettbewerbsfähigkeit des Unternehmens sicherzustellen [KH02, S. 143].

3.1.1. Marktorientierte Produktion

Fragestellung

Während die Bestände der Automobilhersteller im Bereich der Teileversorgung ein Niveau erreicht haben, welches für substanzielle Verbesserungen kaum noch Spielraum lässt, werden auf der anderen Seite – also bei den bereits gefertigten Fahrzeugen – immense Bestände gelagert (siehe Abbildung 3.1). Eine Lagerreichweite von über 60 Tagen² bedeutet bei »Volumenherstellern«, wie z.B. Volkswagen oder Toyota, gebundenes Kapital in Höhe von mehreren Milliarden Euro. Zu den hohen Kapitalkosten addieren sich Kosten für die Wartung der Lagerfahrzeuge sowie die Kosten für Rabatte, die eingeräumt werden müssen, weil vielen Kunden Fahrzeuge nur in einer Konfiguration angeboten werden können, die sich von der jeweiligen Wunschkonfiguration unterscheiden (vgl. [HP04, S. 16]).

Eine wesentliche Ursache für die hohen Bestände bereits gefertigter Fahrzeuge ist der in Kapitel 2.1 beschriebene Auftragsabwicklungsprozess, insbesondere die prognosegetriebene Festlegung von Produktionsvolumina (vgl. [HP04, S. 17]). Die Anzahl produzierter Fahrzeuge wird auf Basis von Prognosen und Effizienzbetrachtungen im Hinblick auf die Fertigung festgelegt und nicht auf Basis der effektiven Marktnachfrage. Während ein vom Kunden bestelltes Fahrzeug in dessen Wunschkonfiguration produziert werden kann, müssen die Fahrzeuge, für die gegenwärtig kein Kundenwunsch vorliegt, in einer vom Händler oder vom Vertrieb gewählten Farbe und Ausstattung gebaut werden.

Die Nachteile dieses »Push-Systems« sind in der Automobilindustrie lange bekannt. Ein seinerzeitiges Vorstandsmitglied der Volkswagen AG schrieb bereits 1996 (vgl. [Har96, S. 38]):

Statt Kapazitäten und Produktionsprogramme zu optimieren, »Stückzahlen zu kloppen«, fehlende Kundenaufträge durch unspezifizierte Mengen zu füllen und anschließend in die Händlerlager und Märkte zu drücken, soll der Fertigungsstrom ausschließlich durch Kunden- bzw. Händleraufträge aus den Fabriken gezogen werden. ... Die Chancen sind enorm:

1. *Höhere Kundenzufriedenheit;*
2. *höhere Kundenloyalität;*
3. *höheres Absatzpotenzial;*
4. *weniger Lagerbindung;*

²Die größte Handelskette für Fahrzeuge in den USA »AutoNation« gibt an, dass die Lagerreichweite in den USA für den Hersteller GM 94 Tage, bei Ford 105 Tage und bei Chrysler 126 Tage betrage (vgl. [Web06]).

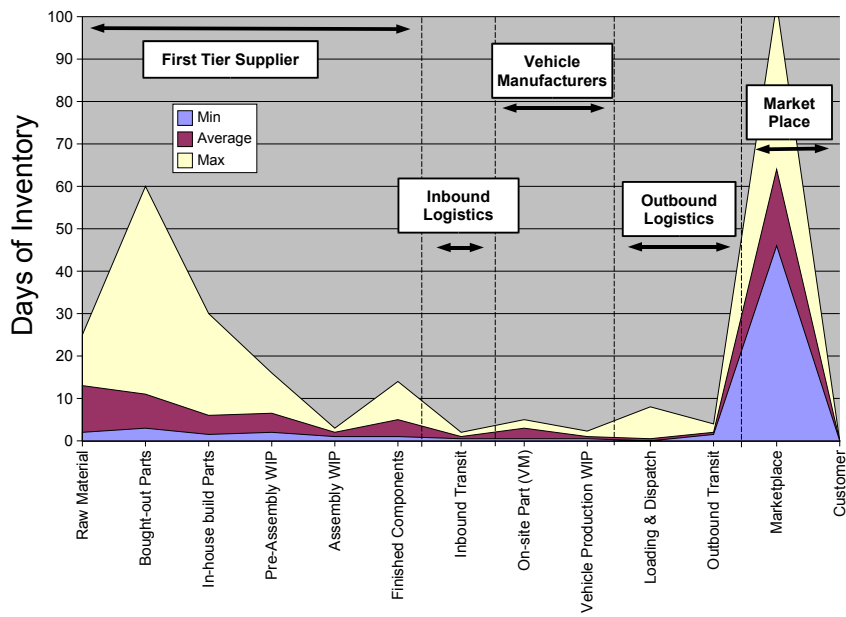


Abbildung 3.1.: Bestandsprofil über die Supply Chain der Automobilhersteller (vgl. [HP04, S. 61])

5. weniger Abverkaufshilfen;
6. weniger Standzeiten und Gewährleistungen;
7. geringere Prozesskosten;
8. und allgemein mehr Marktchancen.

In dem von 1999 bis 2001 durchgeführten »3Day-Car-Programme«³ wurden praktikable Ansätze für eine Umstellung auf eine reine BTO-Strategie gesucht und in zahlreichen Artikeln veröffentlicht (vgl. z.B. [Tea03], [WB02], [EWW00], [How02], [HJ00]). Das »3Day-Car-Programme« ist eines der Fundamente des Forschungsprojektes ILIPT (Intelligent Logistics for Innovative Product Technologies) im Rahmen der »EU 5-Day Car Initiative«, welches den Gedanken der Abkehr von einem push-orientierten System weiter voranbringt (vgl. Projektbeschreibung unter <http://www.ilipt.org>):

The project represents a radical leap for the European automotive industry from the »stock push« and »mass production« thinking of the last century, to a stockless »build-to-order« (BTO) production strategy. This will require the re-invention of the complete automotive value stream from material producers to end consumers of cars, through a cost optimised system delivering what the customer really wants without delay.

Obwohl ein Systemwandel offenbar vielfach als sinnvoll erachtet wird, konstatieren

³Das »Research Team« bestand aus: dem »Lean Enterprise Research Center«, der »University of Bologna«, dem »International Car Distribution Programme (ICDP)« und der »University of Bath« (vgl. [Tea03, S. 33]).

Atmendes Unternehmen

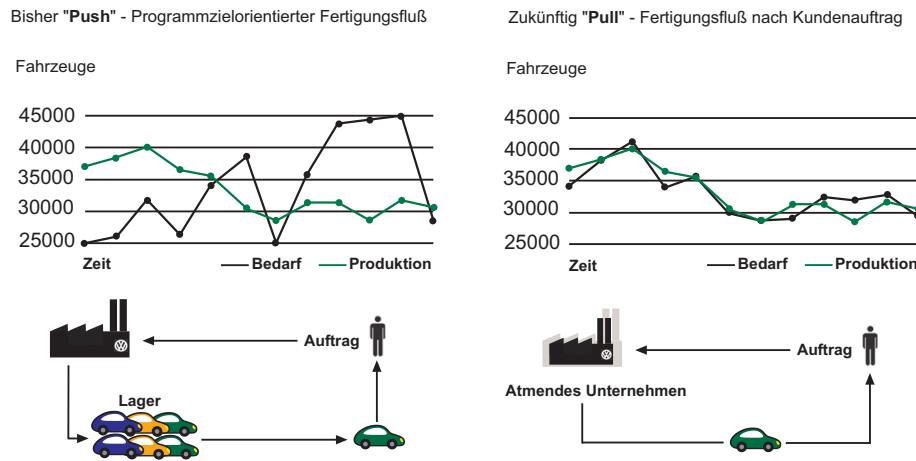


Abbildung 3.2.: Das »atmende Unternehmen« (vgl. [Har96, S. 38])

HOLWEG und PIL [HP04, S. 67]:

Still, the industry stubbornly insists that scale is the only way to survive, and its resistance to moving beyond scale is creating a variety of problems.

Und führen als mögliche Begründung an [HP04, S. 17]:

... that a transition to build-to-order is far more challenging than many companies had envisioned, perhaps in part because they lacked a deep understanding of how each stage of the order-to-delivery process affects build-to-order.

Als erschwerender Umstand wird eingeschätzt, dass die positiven Effekte einer Prozessveränderung, wie z.B. »mehr Flexibilität«, nur schwer monetär zu bewerten sind (vgl. [HP04, S. 173]). Eine kritische Einschätzung zur ausschließlichen Fertigung von Fahrzeugen im Kundenauftrag wird in einem Artikel der Unternehmensberatung Accenture formuliert [Acc02, S. 2]:

BTO, on the other hand, will deliver products that are completely tailor-made but considering existing supply chain constraints – given that infrastructure and processes are predicated on mass production – and high implementation efforts, a pure BTO approach isn't feasible for most OEMs. Furthermore, BTO systems tend to be more vulnerable to demand fluctuations.

Zusammenfassend stehen auf der Seite der Vorteile einer erfolgreichen Umsetzung einer reinen Build-to-Order Fertigung mit minimalen Fahrzeugbeständen unter anderem eine sehr große Freisetzung von gebundenem Kapital, reduzierte Bestandskosten und eine geringere Wahrscheinlichkeit durch große Bestände zur Einräumung von Rabatten gezwungen zu werden (vgl. [WB02, S. 31]). Auf der anderen Seite stehen sowohl für die Automobilhersteller als auch für die Zulieferer hohe Investitionen, um die Anforderungen an Lieferzeit und Flexibilität zu erreichen (vgl. [WB02, S. 17]).

Lösungsstrategie

Um eine Veränderung in Richtung einer reinen Build-to-Order Fertigung zu erreichen, sind zahlreiche konkrete Maßnahmen erforderlich. Beispiele für solche Maßnahmen sind:

1. Eine Senkung der Anzahl der Produktvariabilität (vgl. [HP04, S. 161]), um die Wahrscheinlichkeit zu erhöhen, dass ein Kunde das Wunschfahrzeug im Bestand bereits gefertigter Fahrzeuge findet. Weiterhin erleichtert eine Reduktion der Variabilität die Steigerung der Flexibilität der Werke (siehe unten).

Diese Steigerung der Flexibilität kann z.B. dadurch erreicht werden, dass

- ▷ die Anzahl der vom Kunden wählbaren Optionen eingeschränkt wird. Ein Vergleich der Auswahlmöglichkeiten zwischen z.B. Toyota und BMW⁴ zeigt eine im Vergleich sehr eingeschränkte Wählmöglichkeit bei Toyota. HOLWEG und GREENWOOD konstatieren [HG00, S. 2]:

The main impact of variety is that the more variety is offered, the less likely the customer will find the right vehicle amongst finished car stock.

- ▷ die Anzahl der Varianten eingeschränkt wird, die beim Hersteller intern aus der Kombination der vom Kunden gewünschten Optionen gebildet wird (z.B. der Kunde wünscht die Außenfarbe rot und Ledersitze, der Hersteller färbt die Nähte der Ledersitze in der Außenfarbe⁵ und erhöht damit durch die Farbgebung der Nähte nochmals die Anzahl der Varianten).
- ▷ die Anpassung der Fahrzeuge an den Kundenwunsch in Teilen beim Händler vorgenommen wird⁶ (»late customization«).

Eine weitere Auswirkung hoher Produktvariabilität ist die Beschränkung der Flexibilität der Werke. So wird z.B. durch eine hohe Anzahl von Karosserievarianten die Möglichkeit reduziert, zwischen mehreren Karossen nach dem Rohbau zu wählen, um diesen einem Kundenauftrag zuzuordnen (späte Auftragszuordnung, siehe oben).

2. Veränderung des Auftragsabwicklungsprozesses im Bereich der Einplanung. Das größte Potenzial für eine Verkürzung der Lieferzeit liegt in der Verringerung der Zeit, die zwischen Auftragseingang und Fertigungsbeginn liegt (im Mittel 34 Tage [HP04, S 44]). Hierzu sind umfangreiche Maßnahmen in der Logistik er-

⁴Siehe Produktkonfiguratoren auf den jeweiligen Internetpräsenzen <http://www.toyota.de> und <http://www.bmw.de>.

⁵Rose gibt an, dass es über 18.000 verschiedene Varianten der Türverkleidung für den Audi A6 gibt, 152 verschiedene Ausführung des Handschuhkastens und beim Audi A4 24 verschiedene Ausführungen des Scheibenwaschbehälters (vgl. [Ros05, S. 17]). Baumgarten spricht von 16.000 Varianten einer Türinnenverkleidung bei einem »Fahrzeug der Oberklasse« und immer noch 1.600 Varianten, die der 1st-Tier Lieferant vor der Montage der Verkleidung liefern muss (vgl. [Bau03, S. 5]).

⁶Für den MCC Smart wurde eine Kombination dieser Wege gewählt. Siehe Produktkonfigurator <http://www.smart.com>. Es gibt (Stand: 05.2005, Markt Deutschland) lediglich zwei Farben für die »Tridion Sicherheitszelle« (Karosserie), die Außenfarbe wird über nachträgliche veränderbare Kunststoff-Panels aufgebracht. Eine Anpassung an den Kundenwunsch kann also auch nach der eigentlichen Fertigung erfolgen.

forderlich, aber auch im Bereich der IT-Systeme.

3. Erhöhte Flexibilität zur Anpassung der Anzahl produzierter Fahrzeuge an die tatsächliche Nachfrage durch den Markt (vgl. [HP04, S. 191]). Werden Fahrzeuge ausschließlich dem tatsächlichen Bedarf entsprechend gefertigt, können Bestände auf ein Minimum gesenkt werden. Zur Erhöhung der Flexibilität der Werke der Automobilhersteller sind Veränderungen der Arbeitszeitmodelle ein mögliches Mittel (vgl. [HP04, S. 128]), weiterhin die Auslösung von Restriktionen, die in Bezug auf die Zusammenstellung von Aufträgen in den Werken bestehen, wie z.B. dass keine zwei Fahrzeuge mit Schiebedächern in direkter Folge gefertigt werden können.

Eine erhöhte Flexibilität der Werke der Automobilhersteller kann nur dann genutzt werden, wenn das Lieferantennetzwerk in der Lage ist, den unter Umständen stärkeren Schwankungen in der Nachfrage zu begegnen (vgl. [Hol02, S. 5], [HM03, S. 64]).

4. Nachfragemanagement (vgl. [HP04, S. 200]). Ein Anreizsystem, welches Kunden motiviert, Fahrzeuge zu solchen Zeitpunkten zu bestellen, an denen die Marktnachfrage insgesamt niedrig ist, kann helfen, kurzfristige Nachfrageschwankungen zu nivellieren⁷.

Viele dieser Maßnahmen stehen in Wechselwirkungen miteinander. Eine geringere Produktvariabilität erleichtert die Schaffung eines flexiblen Fertigungssystems, weil weniger Beschränkungen bezüglich der Fertigungsreihenfolge bestehen und die Teileversorgung weniger komplex ist (vgl. [Ada98, S. 36]). Durch Nachfragemanagement können die Flexibilitätsanforderungen für eine Produktion gemäß der tatsächlichen Marktnachfrage gesenkt werden. Eine Verkürzung der Verweildauer der Aufträge in der Zeit zwischen Auftragseingang und tatsächlicher Fertigung erschwert die Umsetzung eines flexiblen Fertigungssystems, da die Zeit, in der Informationen über die Aufträge vorliegen, kürzer wird.

Die Umstellung eines push-orientierten Fertigungssystems auf eine ausschließliche Fertigung im Kundenauftrag erfordert ein ganzes Paket von Maßnahmen, wobei die oben erwähnten nur einen Teil ausmachen. Einige Maßnahmen ermöglichen bereits bei Umsetzung als Einzelmaßnahme eine Kosteneinsparung⁸, andere entfalten den vollen Nutzen erst im Zusammenspiel mit allen Maßnahmen des Paketes⁹.

Lösungsvoraussetzungen

Da die Auswirkungen von Einzelmaßnahmen auf den Gesamtprozess bislang kaum eingeschätzt werden können (vgl. [HP04, S. 17]), bleiben zahlreiche Fragen auf dem Weg einer Implementierung eines BTO-Ansatzes unbeantwortet:

⁷Ziel dieser Anreizsysteme ist es nicht, durch Rabatte das Gesamtvolumen zu erhöhen, sondern lediglich die Kunden zu motivieren, z.B. ein um wenige Tage verzögertes Lieferdatum zu akzeptieren.

⁸Durch die Integration der Fahrzeuge, die sich im Transport zwischen Werk und Händler befinden, in den Pool der für die Kunden zugreifbaren Fahrzeuge, sollte bei Ford eine Senkung des gebundenen Kapitals um eine Milliarde Dollar ermöglicht werden [OLCB01, S. 9].

⁹Ein hohes Maß an Flexibilität ist Voraussetzung für eine reine BTO-Fertigung, bringt aber nur geringe Vorteile, wenn nicht sogar Kostennachteile bei der Beibehaltung der auslastungsorientierten Steuerung.

1. Welche Einzelmaßnahmen führen im Gesamtpaket zum gewünschten Ergebnis?
2. Wie kann eine schrittweise Umsetzung erfolgen und in welcher Reihenfolge?
3. Welche Kosten bzw. welcher Nutzen entsteht bei einer Umsetzung in Schritten?
4. Welche Anforderungen sind an die Teilprozesse (z.B. Teileversorgung) zu stellen?

Der im Rahmen dieser Arbeit vorgestellte Ansatz soll sich für die Beantwortung dieser Fragen eignen und die Umsetzung einer BTO-Strategie unterstützen, indem

- ▷ die Bewertung der Auswirkungen einer Prozessänderung sowohl in Bezug auf die Leistung des Gesamtprozesses als auch auf die Wechselwirkungen mit anderen Teilprozessen möglich sein soll,
- ▷ ein Kommunikationsmittel zur Verfügung gestellt wird, welches in möglichst eindeutiger Weise die Maßnahmen beschreibt und in vielfältiger Weise »befragt« werden kann¹⁰,
- ▷ eine kostenseitige Bewertung der getroffenen Maßnahmen möglich wird,
- ▷ Anforderungen an die Prozesse der Unternehmen der Automobilhersteller, wie auch der Zulieferer, quantifizierbar gemacht werden können (z.B. welche Anforderungen werden an die Flexibilität der einzelnen Zulieferer gestellt).

Es sollen sowohl bereits vorhandene Strukturen abgebildet werden können als auch Konzepte, die noch nicht in allen Details ausformuliert sind. Eine hohe Variabilität des Abstraktionsgrades soll die Begleitung der Prozessgestaltung von der Bewertung eines Grobkonzeptes bis hin zur Realisierung erlauben.

3.1.2. Belegung von Werken

Im Rahmen der Planung der Belegung von Werken versuchen die Automobilhersteller auf Basis sehr langfristiger Prognosen und Planungen zur Modellpalette eine möglichst effiziente Verteilung von zu bauenden Fahrzeugen auf Werksstandorte durchzuführen (vgl. [Bie03, S. 57], [Aut05, S. 1]). Dieser Prozess ist zum einen aufgrund der langfristigen, strategischen Ausrichtung durch ein hohes Maß an Unwägbarkeit gekennzeichnet und zum anderen durch hohe Komplexität, die durch die hohe Anzahl von Produkten und Ressourcen im beplanten Netzwerk verursacht wird.

Fragestellung

Für die Produktion einer Baureihe eines Automobils haben große Automobilhersteller häufig mehrere Optionen, wenn ein Werk oder mehrere Werke zu bestimmen sind, in dem oder denen dieses Automobil gefertigt werden soll. Basis einer Standortentscheidung kann die Bestimmung der Kosten- bzw. Ertragssituation sein, die sich durch eine Auswahl ergibt. Die Bestimmung der Kostenstruktur für eine Werksbelegung hängt von vielen Faktoren ab. Einige dieser Faktoren sind:

¹⁰Es steht dann nicht nur eine Sicht auf den Zielprozess zur Verfügung, sondern es kann mit dem zukünftigen Auftragsabwicklungssystem experimentiert werden.

Fertigungseinrichtungen vor Ort: Ist ein Karosserierohbau vorhanden oder müssen die Rohkarossen angeliefert werden? Von welchen Zuliefererstandorten werden Teile geliefert¹¹? Entsprechen die Lackieranlagen dem gewünschten Standard oder sind Investitionen nötig?

Kapazitäten: In welchem Teil des Modell-Lebenszyklusses¹² befinden sich die anderen Modelle, die in dem Werk gefertigt werden? Soll z.B. ein anderes Modell zum gleichen Zeitpunkt anlaufen, können Ressourcenengpässe entstehen? Ergibt sich durch die geplanten Volumina eine ausreichende Auslastung des Werkes?

Distributionswege: Die in den Werken gefertigten Fahrzeuge müssen zu den Abnehmern transportiert werden. Große Entfernungen zwischen produzierendem Werk und den Abnehmern bringen sowohl relativ hohe Kosten als auch Verzögerungen mit sich, die in langen Lieferzeiten für den Kunden resultieren. Die Entfernung zu den Märkten ist allerdings allein noch kein ausreichendes Kriterium, es müssen stets die konkreten Transportmittel mit Zeitbedarf und Kosten in Betracht gezogen werden. Ein Transport mit dem Schiff von Portugal nach Emden kann u.U. schneller sein als ein Bahntransport von Bayern nach Bukarest. Die Volumina, die über eine Relation fließen, sind in Betracht zu ziehen, da die Volumina entscheidenden Einfluss auf die Wahl des Transportmittels haben und weiterhin die Frequenz der Transporte mitbestimmen.

Lösungsstrategie

Große Konzerne verfügen über Werksanzahlen im zweistelligen Bereich und produzieren in diesen Werken eine nochmals höhere Anzahl von Fahrzeugtypen. Eine bezüglich der Kosten optimale Belegung dieser Werke vorzunehmen, ist daher eine höchst komplexe Aufgabe.

Die Abbildung der einzelnen Baureihen in einem Modell zur Bewertung eines Gestaltungsvorschlages kann recht abstrakt erfolgen, da die genaue Ausgestaltung des Produktes in vielen Fällen noch nicht vorliegt. Jedoch ist sowohl die Anzahl der im Modell abzubildenden Baureihen als auch die Anzahl der Werke verglichen mit anderen Aufgabenstellungen sehr hoch¹³. Innerhalb der Werke sind kritische Ressourcen abzubilden, deren Belastung im zeitlichen Verlauf Rückschlüsse über die Notwendigkeit von Investitionen zulässt. Die Abbildung des Distributionsnetzwerkes muss detailliert erfolgen, um Rückschlüsse auf Kosten und Laufzeiten zu ermöglichen. Im Hinblick auf die Erkennung von Ressourcenengpässen ist eine Berücksichtigung des zeitlichen Verlaufes der Belastung in ihren dynamischen Abhängigkeiten wünschenswert.

¹¹Diese Frage ist im Zuge der Belegungsplanung unter Umständen noch nicht vollständig zu beantworten, da die Auswahl der Zulieferer auch noch zu einem späteren Zeitpunkt stattfinden kann. Lieferanten von »C-Teilen« können auch kurzfristig gewechselt werden.

¹²Mit dem Modell-Zyklus ist typischerweise ein bestimmter Verlauf der Produktionsvolumina verbunden: In der Anlaufphase werden (aus technischen Gründen) zunächst wenige Fahrzeuge gefertigt, diese Anzahl steigt sich bis zum Erreichen der max. Marktnachfrage, nach einigen Jahren der Laufzeit sinken die Volumina bis zur Ablösung des Modells durch den Nachfolger.

¹³Um die Kapazitätssituation der in Betracht kommenden Werke realistisch abbilden zu können, müssen die im fraglichen Intervall gefertigten Baureihen umfassend abgebildet werden.

Die Abbildung des Gesamtnetzes in sehr detaillierter Weise, wie oben angedeutet, eröffnet die Möglichkeit, Alternativen bzgl. ihrer Kosteneffekte ebenso zu bewerten wie bzgl. der Auswirkungen auf kundenrelevante Kenngrößen wie Lieferzeit und Liefertreue.

Eine weitere Strategie zur Lösung der beschriebenen Aufgabe ist die Verwendung von Optimierungsansätzen. BIERWIRTH z.B. (vgl. [Bie03, S. 57-69]) schlägt ein heuristisches Optimierungsverfahren vor, um eine Belegung der Werke zu berechnen, beschränkt dabei aber die Anzahl der in Betracht gezogenen Randbedingungen derart, dass die Qualität bzw. Zulässigkeit der gefundenen Lösung kaum einschätzbar ist. Insbesondere werden durch die statische Berechnung einer Lösung etwaige dynamisch auftretende Engpässe nicht gefunden.

Der von BIERWIRTH vorgeschlagene Ansatz nutzt ein Optimierungsverfahren, um eine Lösung für ein stark vereinfachtes Problem zu suchen, während der hier vorgeschlagene Ansatz eine oder mehrere von einem menschlichen Planer oder aber von einem Optimierungsverfahren entwickelte bzw. ausgewählte Alternativen für die Belegungsplanung sehr detailliert bewerten kann. Das heißt, beide Verfahren können sich bei der Problemlösung durchaus ergänzen.

Die ausschließliche Verwendung von Optimierungsverfahren zur Lösungsfindung kann zu »ungültigen« Werksbelegungen führen, weil dynamisch auftretende Engpässe u.U. nicht gefunden werden. Die ausschließliche Entwicklung von Lösungsalternativen durch einen menschlichen Planer kann dazu führen, dass günstige Lösungsalternativen nicht untersucht werden.

Lösungsvoraussetzungen

Die in der Lösungsstrategie skizzierte Modellstruktur soll zahlreiche Produktionswerke umfassen können, denen wiederum zahlreiche Fahrzeugbaureihen zugeordnet werden können. Es lassen sich folgende wesentliche Anforderungen ableiten:

- ▷ Die Erzeugung von Aufträgen auf Basis vorgegebener (zukünftiger) Fahrzeuganzahlen und Eigenschaftenverteilungen. Die Aufträge müssen der für den Planungszeitraum gültigen Produktstruktur entsprechen (soweit bekannt) und alle Eigenschaften enthalten, die für die Zuordnung zu den Werken relevant sind.
- ▷ Die Einplanung von Aufträgen unter Berücksichtigung von Fertigungs- und Kapazitätsrestriktionen. Insbesondere, wenn ein Modell in mehreren Werken gefertigt werden kann, ist zunächst eine Auswahl unter den alternativen Werken zu treffen. Hierbei ist zu beachten, dass nicht notwendigerweise alle Werke alle Eigenschaften verbauen können (z.B. Rechtslenker könnten nur in einem Werk gebaut werden können).
- ▷ Die Bewältigung hoher Anzahlen von Aufträgen und Eigenschaften. Der lange Untersuchungszeitraum im strategischen Umfeld bringt eine sehr große Anzahl zu bearbeitender Aufträge mit sich, ebenso die »Breite« der Betrachtung (viele Werke, viele Baureihen).
- ▷ Die multikriterielle Bewertung von Engpässen zur Ursachenanalyse. Die große Anzahl von Einflussfaktoren, die die Ergebnisse beeinflussen, machen es erforderlich, selektiv nach Ursachen suchen zu können. Es muss also möglich sein, Anfra-

gen zu stellen, wie z.B.: Berechne die Lieferzeit für Aufträge eines Modells, einer Baureihe, welche in einem bestimmten Werk gefertigt wurden und eine bestimmte Ausstattung (Eigenschaft) enthalten.

3.2. Einsatz im taktischen Bereich

Taktische Aufgabenstellungen sind durch einen mittelfristigen Horizont (Monate bis Jahre) gekennzeichnet und beinhalten z.B. die Auswahl bzw. Reservierung von Ressourcen und setzen den Rahmen für die kurzfristige, operative Planung.

3.2.1. Bedarfs- und Kapazitätsmanagement

Fragestellung

Eine wesentliche Aufgabe der Logistik eines Automobilherstellers ist die Sicherstellung einer ausreichenden Teileversorgung¹⁴ bei gleichzeitig geringen Kosten für Bestände (vgl. [Hoo04, S. 5]). Um diese sicherzustellen, ist zum einen die Kenntnis der zukünftigen Bedarfe und zum anderen die Kenntnis der relevanten Kapazitäten erforderlich. Aus mehreren Gründen sind beide Teilaufgaben nicht leicht zu lösen:

1. Jedes Automobil besteht aus mehreren tausend Teilen, die zum Zeitpunkt der Fertigung dieses Fahrzeugs in der richtigen Qualität, an der richtigen Stelle des fertigen Werkes vorhanden sein müssen¹⁵.
2. Die »Zusammensetzung« eines Fahrzeugs aus Teilen ändert sich beständig. Jeden Tag werden dutzende Teile eines Automobils geändert. Zu jeder Änderung gibt es einen »Einsatztermin«. Beginnend ab diesem Termin ist anstatt des bisherigen Teils ein neues Teil zu verbauen. Die Bestimmung der benötigten Teile eines Fahrzeugs wird auf Basis des beauftragten Ausstattungsumfanges sowie des Tages vorgenommen, an dem das Fahrzeug gebaut werden soll¹⁶.
3. Die Kapazitäten der Lieferanten der Teile sind in der Mehrzahl nicht exakt bekannt¹⁷. Es kann also z.B. nicht ohne Rückfrage zuverlässig bestimmt werden, ob ein Zulieferer eine Steigerung der Produktion um einen bestimmten Prozentsatz durchführen kann¹⁸.

¹⁴Unter »ausreichender« Versorgung soll hier verstanden werden, dass keine Versorgungsengpässe auftreten.

¹⁵In Anlehnung an die vier »r's« der Logistik (vgl. [Pfo04, S. 12]):

Die Logistik hat dafür zu sorgen, dass ein Empfangspunkt gemäß seines Bedarfs von einem Lieferpunkt mit dem richtigen Produkt (in Menge und Sorte), im richtigen Zustand, zur richtigen Zeit, am richtigen Ort, zu den dafür minimalen Kosten versorgt wird.

¹⁶Weitere Faktoren sind möglich, z.B. das Werk, an dem das Fahrzeug gebaut wird.

¹⁷Ein Grund hierfür ist unter anderem, dass Zulieferer nur ungenügende Informationen über ihre Produktionsplanung und -kapazitäten an den Automobilhersteller geben in der Furcht, dieser könnte diese Informationen bei den nächsten Einkaufsverhandlungen gegen den Zulieferer verwenden.

¹⁸Es werden mit dem Zulieferer i.d.R. Flexibilitätsbandbreiten vereinbart, die der Zulieferer abdecken können soll. Der Zulieferer ist allerdings im Rahmen seiner internen Optimierung bestrebt, nur soviel Kapazität vorzuhalten, dass eine ausreichende Versorgung des Kunden gewährleistet werden kann.

4. Kapazitätsüberschreitungen können teilweise erst spät erkannt werden. Da die genaue Bestimmung des Teilebedarfes erst möglich ist, wenn der endgültige¹⁹ Kundenauftrag vorliegt, können Abweichungen in den Teilebedarfen von den prognostizierten Werten erst spät erkannt werden.
5. Die Auswirkungen von Änderungen des Produktionsprogramms auf die Teilebedarfe ist nur schwer zu bestimmen, da zum Zeitpunkt der Verschiebung häufig noch keine Kundenaufträge vorliegen.

Die hier betrachtete Planungsaufgabe ist die Überprüfung der Machbarkeit²⁰ einer Programmplanung mit dem Schwerpunkt auf der Überprüfung der Machbarkeit einer Änderung am Produktionsprogramm. Eine solche Überprüfung erfordert die Bestimmung der durch die Änderung erzeugten Bedarfsschwankung und einer Überprüfung der Kapazitätssituation. Eine Programmveränderung kann z.B. durch Anpassungen der Verteilung der Produktionsvolumina auf die Märkte hervorgerufen werden, die durch eine Aktualisierung von Prognoseinformationen ausgelöst wird. Werden Volumina zwischen Märkten verschoben, steht einem Minderbedarf in Höhe des reduzierten Marktvolumens auf Markt »A« ein Mehrbedarf in Höhe des zusätzlichen Volumens für Markt »B« zur Verfügung. Diese scheinbar triviale Berechnung wird dadurch erschwert, dass sich die Ausstattung der Fahrzeuge in Markt »A« von der in Markt »B« unterscheidet (vgl. [Ohl00, S. 202]). Handelt es sich z.B. um eine Verschiebung vom Markt Spanien in den Markt Kanada, kann der Bedarf für Standheizungen um mehrere hundert Prozent ansteigen. Neben diesen nahe liegenden Ausstattungsunterschieden können hunderte weitere Eigenschaften für Kapazitätsengpässe sorgen.

Eine verwandte Fragestellung ist die Verschiebung des Eigenschaftsmixes für einen Markt. Auch hier werden die Prognosen durch den Vertrieb rollierend aktualisiert, ebenso können Veränderungen zu Teileengpässen führen. Die Bedarfsverschiebung ist insofern leichter zu erkennen als bei der Verschiebung ganzer Fahrzeuge, weil die betroffene Eigenschaft explizit benannt wird. Aber auch hier können durch eine Eigenschaft, bedingt durch die Baubarkeitsregeln (s.u.) für Aufträge, weitere Eigenschaften »erzwungen« werden.

Beispiel 3.3 zeigt ein typisches System von Einbauraten²¹ und Konstruktionsregeln²², wie es bei vielen Automobilherstellern zu finden ist. Die Notwendigkeit der Regeln ist offensichtlich: Nicht jede Felge kann mit jedem Reifen kombiniert werden. Wird die Einbaurate einer bestimmten Felge erhöht, muss auch die Einbaurate der Reifen steigen, die auf diese Felge passen. Im Gegenzug müssen die Einbauraten anderer Reifen und Felgen sinken, da ein Fahrzeug stets mit genau einem Reifen- und Felgensatz bestückt ist. Eine Erhöhung der Einbaurate für eine Eigenschaft bewirkt eine Veränderung der Einbauraten in einer anderen Familie. Eine Veränderung einer »Stellschraube« zieht die Veränderung vieler weiterer Parameter nach sich und

Außergewöhnliche Bedarfsschwankungen können also zwar innerhalb des ausgehandelten Flexibilitätsrahmens liegen, aber trotzdem zu einer Lieferunfähigkeit der Lieferanten führen.

¹⁹Der Auftrag darf durch den Kunden nicht mehr geändert werden können.

²⁰Sind die Kapazitäten ausreichend?

²¹Die *Einbauraten* legen den Anteil von Aufträgen bzw. Fahrzeugen fest, die eine gewisse Eigenschaft aufweisen (z.B. 35 % enthalten ein Schiebedach).

²²Diese Regeln werden auch als *Baubarkeitsregeln* bezeichnet.

Anzahl Fahrzeuge		1000			
Felgen	Stahl	16"	17"	19"	
Anzahl	200	200	200	400	
Einbaurrate	0,2	0,2	0,2	0,4	
Reifen	205	215	225	245	
Anzahl	200	200	200	400	
Einbaurrate	0,2	0,2	0,2	0,4	
Regeln:					
	Stahl nur mit 205				
	16" nur mit 215 oder 225				
	17" nur mit 225 oder 245				
	19" nur mit 245				
	Modell "Sport" nur mit 19"				
	Modell "Sport" mit 19" und 245 erhält Kotflügelverbreiterung				
oder:					
	205 nur mit Stahl				
	215 nur mit 16"				
	225 nur mit 16" oder 17"				
	245 mit 17" oder 19"				
	Modell "Sport" nur mit 19" auf 245 Felgen				
	Modell "Sport" mit 19" und 245 erhält Kotflügelverbreiterung				

Abbildung 3.3.: Konstruktionsregeln am Beispiel

muss so durchgeführt werden, dass wiederum eine »stimmige«²³ Verteilung von Eigenschaften erreicht wird.

Lautet die Fragestellung: »Ist eine Erhöhung der Einbaurrate für 19-Zoll-Felgen für die Baureihe X im Markt Frankreich um 10 %-Punkte machbar«, ist zu prüfen, ob Kapazitätsober- bzw. untergrenzen²⁴ verletzt werden. Um eine Kapazitätsüber- bzw. -unterschreitung feststellen zu können, muss bestimmt werden, welche Teile zusätzlich benötigt werden und bei welchen Teilen ein Minderbedarf entsteht.

Dabei verteilt sich eine Steigerung einer Einbaurrate für eine Baureihe und einen Markt nicht unbedingt gleichmäßig auf alle Modelle für diesen Markt. So ist wahrscheinlich die Einbaurrate für die breitesten verfügbaren Reifen in diesem Beispiel bei der Spitzenmotorisierung weitaus höher als bei der Basismotorisierung. Diese Verteilung auf die Modelle einer Baureihe ist entscheidend für die Bestimmung der tatsächlichen Bedarfsveränderung, die durch die Änderung der Einbaurrate 19-Zoll-Felgen hervorgerufen wird, da die Reifen für die Basismotorisierung einen niedrigeren Geschwindigkeitsindex²⁵ benötigen als für die Spitzenmotorisierung. Um den neuen Gesamtbedarf für Reifen berechnen zu können, muss die Verteilung der Reifen/Felgenkombinationen auf die verschiedenen Motorisierungsvarianten bekannt sein²⁶.

²³»Stimmig« heißt hier, dass die Eigenschaften so auf Aufträge verteilt werden können, dass unter Berücksichtigung des Regelwerkes baubare Aufträge spezifiziert werden können.

²⁴Ist mit einem Zulieferer eine Mindestabnahmemenge vereinbart worden, kann auch eine Senkung des Bedarfs Probleme verursachen.

²⁵Die höchste zugelassene Geschwindigkeit, mit der die Reifen betrieben werden dürfen.

²⁶Aufgrund der teilweise sehr hohen Anzahl von Modellen innerhalb einer Baureihe, sollte eine Prognose der Verteilung der Einbaurrate jeder Eigenschaft auf die Modelle automatisiert erfolgen, z.B. auf Basis

Für die Varianten der fraglichen Baureihe können zudem Regeln definiert sein, die den Einbau zusätzlicher Teile in das Fahrzeug bewirken. Zum Beispiel kann die Kombination 19-Zoll-Felgen mit 245er-Reifen mit dem Einbau einer Kotflügelverbreiterung im Modell »Sport« verbunden sein (siehe Abbildung 3.3). Neben den vielfältigen Verschiebungen zwischen Teilebedarfen, die durch eine Betrachtung bezogen auf einen Zeitpunkt sichtbar gemacht werden können, ist für eine Machbarkeitsaussage einer Einbauratenänderung die dynamische Entwicklung zu beachten. Es verändern sich die Absatzzahlen der Baureihen über die Zeit, neue Varianten kommen hinzu, andere Varianten entfallen, die Konstruktionsregeln verändern sich über die Zeit und schließlich ändert sich potenziell die Zuordnung von Teilen zu Eigenschaften. So kann z.B. eine ausreichende Kapazität in einem Monat durch einen Wechsel des Reifenzulieferers und den Entfall einer möglichen Reifengröße für die 19-Zoll-Felge plötzlich ein Engpass werden.

Lösungsstrategie

Automobilhersteller, die ihren Lieferanten Prognosen über die zu erwartenden Abnahmemengen in einem Bereich mitteilen, in dem noch keine Aufträge vorhanden sind, verfügen notwendigerweise über ein Verfahren, die Teilebedarfe auf Basis von prognostizierten Abnahmemengen zu schätzen. Selbst bei Neuentwicklungen, wie dem auf der Nutzung von Markov-Netzwerken basierenden IT-System »EPL« (EPL steht für »Eigenschaftsplanung«) durch die Volkswagen AG (vgl. [GDM03, S. 2]), gelingt die vollständige Abbildung der vielfältigen Zusammenhänge der Eigenschaften mit den damit verknüpften Teilebedarfen nur unvollständig (vgl. [GDM03, S. 3]).

Eine Alternative zu einer direkten mathematischen Repräsentation aller Einbauraten und Regeln, die offenbar (siehe oben) aufgrund der sehr großen Modelle zumindest derzeit nicht vollständig möglich ist, ist die Verwendung von »Planaufträgen«, die entsprechend der gewünschten Einbauraten erzeugt und gegenüber Konstruktionsregeln geprüft werden können. Diese Planaufträge können genutzt werden, um durch eine Stücklistenauflösung – auch unter Nutzung bereits vorhandener Stücklistensysteme – die Teilebedarfe in gewünschter Weise zu berechnen.

Eine gewünschte Änderung von Einbauraten führt bei dieser Vorgehensweise zur Erzeugung einer neuen Menge von Planaufträgen mit eben diesen geänderten Einbauraten und der Berechnung der neuen Teilebedarfe auf dieser Basis. Gegenüber einem mathematischen Modell, welches nur eine Teilmenge der Konstruktionsregeln und Teilebedarfe berücksichtigt, bietet der Ansatz »Planaufträge« einige Vorteile:

- ▷ Die erzeugten Aufträge können leicht gegen alle Konstruktionsregeln geprüft werden, auch unter Nutzung vorhandener Lösungen für »echte Aufträge«.
- ▷ Die Berechnung der Teilebedarfe kann auch auf Teilen durchgeführt werden, die nicht im Kreis der enger überwachten Teile im Bedarfs- und Kapazitätsmanagement stehen. Es lassen sich z.B. große Bedarfsschwankungen für solche Teile ausweisen.
- ▷ Die Qualität der errechneten Lösung ist leichter nachvollziehbar. Entsprechen die Einbauraten der Planaufträge den Vorgaben und entsprechen die Planaufträge allen

der Verteilung in der unmittelbaren Vergangenheit.

Konstruktionsregeln, so ist die berechnete Lösung gültig.

- ▷ Die Planaufträge können in einem Simulationsmodell eingesetzt werden, um z.B. eine Überschreitung von Werksrestriktionen zu prüfen (siehe Kapitel 3.2.2).

Teilebedarfe, die durch eine Kombination von Eigenschaften mit sehr geringen Einbauraten entstehen, können durch eine solche Lösung allerdings nur schlecht prognostiziert werden. Eine Einschränkung, die in der Struktur des zugrunde liegenden Problems begründet liegt und daher nicht nur das Verfahren auf Basis der Planauftragserzeugung betrifft.

Lösungsvoraussetzungen

Die Erzeugung von Planaufträgen nach vorgegebenen Einbauraten und unter Beachtung von Konstruktionsregeln ist ein wesentliches Element des hier vorgestellten Ansatzes. Die Anforderungen an eine planauftragsbasierte Bedarfsberechnung überschneiden sich in vielen Bereichen mit denen an einen Bewertungsansatz für Auftragsabwicklungsprozesse, wie z.B. in Forderungen nach einer sehr detaillierten Produktabbildung und einer sehr effizienten Umsetzung in einem IT-Tool²⁷.

Die Forderung nach einem laufzeiteffizienten Werkzeug ist für die Beantwortung dieser Fragestellung von besonderer Relevanz, da aufgrund der hohen Anzahl möglicher Auftragskonfigurationen für eine Bedarfsprognose eine sehr große Anzahl von Aufträgen zu erzeugen ist.

Werden die Bedarfsberechnung und die oben vorgestellte Werksprogrammplanung als Bausteine einer umfassenderen Lösung gesehen, werden zusätzlich zur Planauftragserzeugung weitere Bereiche des Ansatzes bzw. Werkzeugs genutzt, insbesondere die Simulationskomponente.

3.2.2. Werksprogrammplanung

Fragestellung

Die Programmplanung eines Unternehmens hat die Aufgabe, unter Beachtung einer Reihe von Kriterien, wie z.B. Kosten, Lieferzeit, Liefertreue, zwischen Marktnachfrage und verfügbaren Ressourcen zu vermitteln.

Eine der Herausforderungen der Programmplanung besteht darin, die Implikationen der Fertigungsrestriktionen der Werke auf die Gesamtplanung zu bewerten. Fertigungsrestriktionen beschränken die Anzahl der Fahrzeuge mit bestimmten Eigenschaften oder Kombinationen von Eigenschaften und sind insbesondere in ihren Kombinationswirkungen schwer abzuschätzen.

Diese Kombinationswirkungen können dazu führen, dass eine Menge von Aufträgen nicht vollständig eingeplant werden kann, obwohl die Kapazitäten, bezogen auf einzelne Eigenschaften, nicht ausgeschöpft sind.

²⁷Meyer definiert *Effizienz* ([Mey97, S. 9]):

Efficiency is the ability of a software system to place as few demands as possible on hardware resources, such as processor time, space occupied in internal and external memories, bandwidth used in communication devices.

Für die Einplanung von Aufträgen »gegen« eine Menge von Restriktionen sind Lösungen verfügbar, die zudem Gleichverteilungskriterien für die Verteilung der Aufträge über einen zeitlichen Abschnitt berücksichtigen können. So ist z.B. eine bei Toyota im Einsatz befindliche Lösung publiziert [PC99, S. 242], die neben der Berücksichtigung von Restriktionen eine Gleichverteilung für ausgewählte (ca. 20) Eigenschaften erreicht. Zur Optimierung dieser Gleichverteilung wird bei diesem Ansatz eine sog. *Zielfunktion* über der Auftragsreihenfolge definiert, deren Maximierung mittels der »Goal-Chasing-Methode« durchgeführt wird (siehe auch [MNZ02, S. 1269-1273]).

Für die Berücksichtigung von Werksrestriktionen bei der Programmplanung müssen, um den oben erwähnten Optimierungsansatz nutzen zu können, bereits alle Aufträge vorliegen, eine Vorbedingung, die nur selten erfüllt ist²⁸. Eine weitere Hürde für den Einsatz optimierender Verfahren ist die im mittel- und langfristigen Bereich sehr große Anzahl von Aufträgen, die zu bearbeiten sind. Die oben zitierte Beschränkung auf 20 aus mehreren tausend Teilen, aus denen ein Fahrzeug besteht, ist erforderlich, um eine für Toyota akzeptable Rechenzeit zu erzielen (vgl. [PC99, S. 242]). Eine erhebliche Ausweitung des Betrachtungshorizontes ist neben dem Fehlen von Aufträgen also u.U. in Bezug auf die erforderliche Berechnungszeit nicht mehr praktikabel.

Lösungsstrategie

Ein Schlüsselement des Lösungsvorschlags zur Lösung der obigen Fragestellung liegt in der Erzeugung von Aufträgen, die in der Anzahl und der Eigenschaftenverteilung der Programmplanung entsprechen. Durch die sukzessive Einplanung dieser Aufträge unter Beachtung der Restriktionen lässt sich prüfen, inwieweit die geplanten Volumina unter Beachtung von Fertigungsrestriktionen und Lieferantenkapazitäten erreichbar sind.

Weiterhin kann auf Basis vorhandener Aufträge eine Optimierung der Verteilung durchgeführt werden, um eine gleichmäßige Auslastung von Ressourcen zu erreichen. Allerdings sind der Optimierung durch die, im Vergleich zu kurzfristigen Planungen, große Anzahl von Aufträgen, wie oben bereits dargestellt, enge Grenzen gesetzt.

Lösungsvoraussetzungen

Der im Folgenden vorgestellte Ansatz soll durch die ganzheitliche Betrachtung, über die Einplanung des Auftrages hinweg bis hin zur Übergabe an den Kunden, eine Bewertung der kundenwirksamen Auswirkungen einer Programmplanung möglich machen. Eine hohe Eignung des Ansatzes zur Unterstützung der Werksprogrammplanung soll durch folgende Eigenschaften bzw. Möglichkeiten begründet werden:

- ▷ Erzeugung von Aufträgen entsprechend vorgegebener Volumina und Eigenschaftenverteilungen.
- ▷ Einplanung von Aufträgen unter Berücksichtigung von Fertigungs- und Kapazitätsrestriktionen der Werke und Zulieferer.

²⁸Ist ein Fahrzeugmodell am Markt sehr erfolgreich, kann es vorkommen, dass zu dem Zeitpunkt, an dem die Programmplanung i.d.R. durchgeführt wird, bereits so viele Aufträge vorliegen, dass man auf die Verwendung von Prognosen für diesen Zeitraum vollständig verzichten kann.

- ▷ Verarbeitung einer großer Anzahlen von Aufträgen in kurzer Zeit (ohne Optimierung der Auftragsreihenfolge).
- ▷ Multikriterielle Bewertung der Ergebnisse.

Für eine im laufenden Betrieb einsetzbare Lösung sind weitere Komponenten erforderlich, wie eine angepasste Benutzeroberfläche, angepasste Auswertungen sowie Schnittstellen zu Systemen der Automobilhersteller, die die erforderlichen Planungsdaten enthalten. Diese Komponenten sind ohne weitere Schwierigkeiten implementierbar, aber nicht Bestandteil der hier vorgestellten Lösung.

3.3. Einsatz im operativen Bereich

Aufgaben im operativen Bereich sind einerseits geprägt durch die zeitliche Nähe²⁹ des Zeitraumes der Fragestellung und zum anderen durch unmittelbare Auswirkungen einer Entscheidung auf das Unternehmen. Im operativen Kontext steht daher weniger die Gestaltung des Auftragsabwicklungsprozesses im Vordergrund, als die Bewertung einer Planungsalternative.

3.3.1. Distributionsplanung

Fragestellung

Wie viele weitere Planungsaktivitäten im Bereich des Auftragsabwicklungsprozesses, wird die Distributionsplanung für einen Zeitabschnitt mehrfach durchgeführt. Lange bevor der eigentliche Transport stattfindet, z.B. wenn für das Folgejahr mit den Transportdienstleistern Verträge geschlossen werden sollen, ist zu ermitteln, wie viele Fahrzeuge über eine Relation bewegt werden sollen, um die hierfür erforderlichen Transportkapazitäten einzukaufen.

Diese Abschätzung der Transportvolumina kann aufgrund zusätzlicher Informationen, die mit Fortschreiten der Zeit zur Verfügung stehen, präzisiert werden. So kann zum Zeitpunkt des »Einfrierens« des Programms, viele Wochen vor der tatsächlichen Produktion der Fahrzeuge, für einen Zeitraum – auf Basis des Programms – ermittelt werden, wie viele Fahrzeuge für welche Relation geplant werden müssen.

Obwohl auch die langfristige Planung und Bewertung von Distributionsnetzwerken, z.B. zum Zeitpunkt der Vergabe von Verträgen an Transportdienstleister, als Anwendungsgebiet für den vorzustellenden Ansatz in Frage kommt, soll hier auf die Unterstützung im kurzfristigen Bereich eingegangen werden. Unter »kurzfristigem« Bereich soll die Unterstützung bei der konkreten Steuerung der Fahrzeugdistribution verstanden werden.

Durch die langfristige Transportplanung werden die für eine Relation in Frage kommenden Transportmittel festgelegt, wobei die Verteilung der Volumina auf die Relationen an die Absatzprognosen des Herstellers angelehnt wird. Durch Abweichungen

²⁹Gablers »Lexikon Controlling und Kostenrechnung« definiert den Begriff *operative Planung* als: »Begriff: Bei der o.P. handelt es sich i.d.R. um die kurzfristige Planung der Leistungserstellung und Leistungsverwertung im Rahmen gegebener Kapazitäten. Es erfolgt eine detaillierte Festlegung der zur Erreichung der strategischen Ziele notwendigen Aktivitäten. In einem divisionalisierten Unternehmen regelt die o.P. die Aktivitäten der Geschäfts- und Zentralbereiche.«

der Marktnachfrage von den prognostizierten Werten entstehen Ressourcenengpässe, die durch planerischen Eingriff aufzulösen sind.

Ein typisches Beispiel für eine solche Engpassressource ist die Stellfläche in einem Lager. Auf der einen Seite muss für eine Stellfläche garantiert werden, dass die Menge der Zuflüsse abzüglich der Menge der Abflüsse zu jedem Zeitpunkt kleiner ist als die Kapazität der Stellfläche. Auf der anderen Seite sollte die Stellflächenkapazität nicht nur zu einem kleinen Teil ausgeschöpft werden. Ist eine Entscheidung zu fällen, ob eine Stellfläche eine Anzahl weiterer Fahrzeuge aufnehmen kann, ist ein Blick auf den gegenwärtigen Befüllungsgrad nicht ausreichend – weitere Faktoren müssen beachtet werden:

1. Für welche Relationen wird die Stellfläche genutzt?
2. Welche Volumina werden gemäß dem aktuellen Planungsstand in der nächsten Zeit auf die Stellfläche zulaufen?
3. Mit welcher Varianz ist in Bezug auf die Abweichung Planung/tatsächlicher Zulauf zu rechnen?
4. Welche Volumina werden voraussichtlich abtransportiert?
5. Mit welcher Varianz ist in Bezug auf die Abweichung Planung/tatsächlicher Abtransport zu rechnen?
6. Gibt es weitere Nutzer der Stellfläche, die die verfügbare Kapazität weiter reduzieren?

Das gewünschte Ergebnis sollte eine Entscheidungsgrundlage liefern, ob eine gewisse Menge an Fahrzeugen in Richtung dieser Stellfläche in Bewegung gesetzt werden kann, oder eben nicht. Im Falle einer Fehleinschätzung kann z.B. ein Zug, der die Fahrzeuge transportiert, nicht entladen werden. Zum einen entstehen dabei Kosten für die länger als geplant belegten Waggons bzw. Gleise, zum anderen können die Fahrzeuge nicht mehr, oder nur unter Inkaufnahme weiterer Kosten, per LKW direkt zu den Händlern transportiert werden.

Ein kennzeichnendes Element der hier angesprochenen Kategorie von Fragestellungen ist, dass zu dem Zeitpunkt, an dem eine Entscheidung zu treffen ist, Kundenaufträge nur unvollständig vorliegen und somit eine Beantwortung der Fragestellung durch eine Analyse der vorliegenden Aufträge nicht möglich ist. Das heißt, es ist auf der Basis von Prognosen zu entscheiden, welche Belastung auf das Distributionssystem zukommt.

Lösungsstrategie

Auf Basis eines Modells eines Distributionsnetzwerkes, welches neben der statischen Struktur auch die Regeln abbildet, wie Fahrzeuge durch das Netzwerk zu bewegen sind, kann bei vorhandenen Aufträgen und Informationen über die gegenwärtige Befüllung von Lagerflächen auf nahe liegende Weise eine Prognose über zukünftige Entwicklungen gegeben werden: Werden die vorliegenden Aufträge als »Systemlast« in das Modell gegeben, kann im Modell die Belastung der einzelnen Ressourcen gemessen werden.

Herausforderungen bei der Abbildung von Distributionsnetzwerken in der Automobilindustrie sind zum einen die Größe der Netzwerke und komplexe Steuerungsregeln und zum anderen die Notwendigkeit Aufträge zur Verfügung zu stellen, ohne auf eine ausreichende Anzahl »echter« Kundenaufträge zurückgreifen zu können.

Lösungsvoraussetzungen

Neben der bereits oben genannten Anforderung, dem Erzeugen von Aufträgen auf Basis von Prognosen, ist hier die Möglichkeit der Abbildung komplexer Netzwerke und komplexer Steuerungsregeln erforderlich, um die oben skizzierte Strategie zur Unterstützung von Planungsprozessen im Bereich der Distribution umzusetzen.

3.3.2. Störungsmanagement

Fragestellung

Niedrige Bestände und die Sicherheit der Teileversorgung stehen in einem offensichtlichen Zielkonflikt: Wird die Versorgung eines Teils unterbrochen, kann die Produktion von Fahrzeugen, für die dieses Teil benötigt wird, nur durch Rückgriff auf Bestände aufrecht erhalten werden. Sind die Bestände sehr klein, führt ein Versorgungsengpass schnell zu Schwierigkeiten in der Produktion³⁰.

Die Ursachen für eine Versorgungsunterbrechung sind vielfältig und hängen wesentlich vom Standort eines Werkes ab. So unterscheiden sich z.B. die Neigung der Belegschaften der Werke und Zulieferer zum Streik, das Risiko, dass LKWs in Staus warten müssen und die Wahrscheinlichkeit, dass durch Qualitätsprobleme Teile nicht verwendbar sind.

Tritt ein Teile-Engpass auf, muss schnellstmöglich eine Strategie gefunden werden, die negativen Auswirkungen dieses Engpasses zu minimieren. Die Ableitung einer solchen Strategie ist eine komplexe Aufgabe, da sehr viele Faktoren in Betracht gezogen werden müssen. Die Quelle der Komplexität besteht in der Schwierigkeit abzuleiten, welche Auswirkungen eine Änderung der Fertigungsreihenfolge hat. Werden betroffene Aufträge aus dem Programm genommen, kann die verbleibende Auftragsreihe bereits nicht mehr baubar sein, weil Reihenfolgekriterien verletzt werden³¹. Ein »Vorziehen« von Aufträgen kann neue Probleme induzieren, da für diese vorgezogenen Aufträge möglicherweise die Teileversorgung nicht sichergestellt werden kann.

Eine Reaktion auf den Engpass sollte möglichst die vorhandenen Kapazitäten nutzen, die Liefertermine der Fahrzeuge nicht oder möglichst wenig verändern und keine zusätzlichen Engpässe provozieren.

Lösungsstrategie

Es sind viele Strategien zur Vermeidung bzw. Bewältigung von durch Teileengpässe verursachten Schwierigkeiten denkbar:

³⁰Ein Beispiel ist der mehrtägige Produktionsstillstand bei der Volkswagen AG im Werk Wolfsburg, der durch den Ausfall einer Presse zustande gekommen ist (Meldung der Financial Times Deutschland vom 25.5.05 [im Internet unter <http://www.ftd.de>]).

³¹Existiert z.B. ein Reihenfolgekriterium »max. jedes dritte Fahrzeug mit Schiebedach« und in der bisherigen Auftragsfolge ist die maximal mögliche Anzahl von Fahrzeugen mit Schiebedach ausgeschöpft, führt das Entfernen von Aufträgen ohne Schiebedach zu einem Konflikt.

- ▷ Wenn knappe Teile bei einem Zulieferer vorhanden sind, aber eine Anlieferung über den »Standard-Lieferweg« zu lange dauert, kann ein schnellerer (und i.d.R. teurerer) Transport, z.B. per Flugzeug, in Erwägung gezogen werden.
- ▷ Aufträge, die von einem Engpass betroffen sind, können entweder zum nächstmöglichen Zeitpunkt gefertigt werden, an dem die Teile wieder verfügbar sind, oder aber neu eingeplant werden³². Im ersten Fall verzögert sich der Liefertermin für alle Aufträge, die das Engpassteil erhalten sollen (viele Kunden bekommen ihr Fahrzeug ein bisschen später – der Dominoeffekt). Im letzten Fall wird nur ein Teil der betroffenen Aufträge verschoben, diese werden aber massiv verzögert (wenige Kunden erhalten ihr Fahrzeug stark verspätet).
- ▷ Betrifft der Engpass nicht alle Aufträge, die gefertigt werden könnten, können vom Ausfall nicht betroffene Aufträge »vorgezogen« werden³³, oder aber es wird die Anzahl der an diesen Tagen produzierten Fahrzeuge verringert.
- ▷ Anstatt der nicht verfügbaren Teile können temporär andere Teile zur späteren Umrüstung verbaut werden³⁴.

Eine mögliche Unterstützung der Bearbeitung von Versorgungsstörungen umfasst zwei Schritte: Erstens die Festlegung der möglichen und sinnvollen Reaktionen auf einen Engpass im Vorfeld und zweitens die Unterstützung im konkreten Störfall. Sowohl für die Festlegung einer Sammlung von Maßnahmen für die Störungsbehandlung als auch für die konkrete Unterstützung im Falle eines Engpasses ist IT-Unterstützung erforderlich.

Die Festlegung, welche Maßnahmen vom Fachpersonal ergriffen werden können, hängt eng mit wichtigen Rahmenbedingungen des Auftragsabwicklungsprozesses zusammen. Ist es z.B. möglich, das Personal in einem von einer Versorgungsstörung betroffenen Werk kürzer arbeiten zu lassen, um einen aufgetretenen Rückstand zu einem späteren Zeitpunkt aufzuholen? Sind damit Zusatzkosten verbunden? Können Aufträge vorgezogen werden und wenn ja, um welche Zeitspanne? Können Aufträge in ein anderes Werk verschoben werden?

Lösungsvoraussetzungen

Der im Folgenden vorgestellte Ansatz soll sich sowohl für die Festlegung der Rahmenbedingungen, die für ein Störungsmanagement gelten sollen als auch für eine Unterstützung bei der Lösung von Fragestellungen, die beim tatsächlichen Eintreten eines Störfalles entstehen (operativer Einsatz, siehe unten), eignen.

Beide Aufgaben erfordern insbesondere die detaillierte Abbildung von Aufträgen, wobei im Fall der operativen Unterstützung alle von dem jeweiligen Hersteller verwalteten Eigenschaften repräsentiert werden müssen, da im vorhinein nicht bekannt ist,

³²Der Auftrag kann z.B. wie ein neuer Kundenauftrag behandelt werden.

³³Das »Vorziehen« von Aufträgen bringt es mit sich, dass Fahrzeuge vor dem bislang geplanten Fertigungsbeginn produziert werden und somit Teilebedarfe früher als geplant auftreten. Dies kann z.B. bei auftragsbezogen gefertigten Teilen, wie z.B. dem Kabelbaum, zu neuen Problemen führen.

³⁴Eine bestimmte Reifen/Felgen-Kombination kann z.B. durch eine andere ersetzt werden. So kann das Fahrzeug gefertigt werden und wird nachträglich auf die vom Kunden gewünschten Reifen/Felgen umgerüstet.

welches Teil von einer Störung betroffen ist. Um eine Strategie für das Störungsmanagement festzulegen, müssen verschiedene alternative Strategien abgebildet werden können, um die tauglichsten für den operativen Betrieb auswählen zu können. Zudem müssen Fertigungsrestriktionen und Kapazitäten Berücksichtigung finden, um mögliche adverse Effekte durch eine etwaige Verschiebung von Aufträgen aufdecken zu können.

Der vorgestellte Ansatz soll die Unterstützung des Störungsmanagements auf den angesprochenen zwei Ebenen ermöglichen. Zum einen sollen geeignete Regeln zur Verschiebung von Aufträgen im Vorfeld durch Experimentieren mit verschiedenen Ansätzen ausgewählt werden können und zum anderen soll die Möglichkeit zur Verfügung gestellt werden, alternative Reaktionen auf einen im »Tagesgeschäft« auftretenden Engpass zu untersuchen. Hierfür ist die Voraussetzung zu schaffen, »echte« Aufträge und Kapazitätsinformationen in eine Untersuchung einzubeziehen.

4. Anforderungen an einen Modellierungs- und Bewertungsansatz

4.1. Ausgangssituation

Neben den zweifellos wichtigen psychologischen Faktoren, die einen Veränderungsprozess in einem Unternehmen beeinflussen, sowohl auf Seiten des Marktes bzw. der Kunden als auch auf der Seite der betroffenen Belegschaft der Unternehmen, macht die hohe Komplexität¹ der Prozesse der Automobilhersteller Veränderungen in ihren Auswirkungen schwer überschaubar: Komplexität schafft Intransparenz.

Beispiel: Häufig sind mehrere Fachbereiche von einer Änderung betroffen: Interessenskonflikte entstehen, wenn eine Änderung (voraussichtlich) nicht für alle Beteiligten positive Wirkungen erzeugt. Zudem lassen sich die Auswirkungen auf die Erlössituation eines Unternehmens mit bisherigen Methoden kaum einschätzen, da eine geeignete Methode bislang fehlt.

Die Verfügbarkeit eines Ansatzes zur detaillierten und quantitativen Bewertung einer Prozessveränderung vor deren Umsetzung könnte diese Situation grundlegend ändern (vgl. [SK05, S. 315]): Es ließen sich die Auswirkungen für die Prozessbeteiligten in den Fachabteilungen im Vorfeld klären und unbegründete Vorbehalte ausräumen; die Quantifizierung negativer Auswirkungen erlaubte einen Interessenausgleich. Neben der Be- bzw. Entlastung unternehmensinterner Ressourcen ließen sich auch Anforderungen an externe Ressourcen quantifizieren. Die Möglichkeit einer Kosten-/Nutzenbetrachtung würde kostenträchtige Fehlschläge vermeiden und legte das Fundament zur Umsetzung von Konzepten, die sich bislang nicht als gewinnträchtig nachweisen ließen.

Die Bewertung verschiedener Ausprägungen des Auftragsabwicklungsprozesses bzw. von Teilen hiervon am Modell ist vorteilhaft, denn eine Untersuchung am Realsystem² ist mit hohen Kosten und Risiken behaftet (vgl. [LK91, S. 2], [Rai98]).

HOMMES zählt (vgl. [HvR00], vgl. auch [ATH05]) allein über 350 Ansätze zur Repräsentation von Geschäftsprozessen³; eine Teilmenge hiervon ist auch zur Bewertung von Geschäftsprozessen geeignet. Soll eine dieser Repräsentationen eingesetzt werden, um einen höchst komplexen Prozess zu analysieren, kann es erforderlich sein, die

¹Für Definitionen von Komplexität bzw. komplexen Systemen vgl. [WSS01], [Sha75, S. 37], [Neu95].
BAR-YAM über *Komplexe Systeme* [BY97, S. 1].

²Eine Definition von »System« wird in Kapitel 4.2 gegeben.

³JACOBSON definiert [JEJ95, S. 3]:

A business process is the set of internal activities performed to serve a customer.

Der Auftragsabwicklungsprozess ist demzufolge ein *Geschäftsprozess*.

Komplexität des Prozesses in einem entsprechenden Modell einzufangen; das Modellierungsparadigma muss es also erlauben, komplexe Prozesse adäquat abzubilden⁴.

Neben der Komplexität des abzubildenden Systems spielt die Komplexität auf einer anderen Ebene eine für die Aufgabenstellung bedeutsame Rolle: Komplexität, die durch die Konstruktion von Software entsteht, die eine Abbildung des Auftragsabwicklungsprozesses erlauben soll. BOOCH widmet in [Boo94, S. 3-26] dem Thema Komplexität von Software ein ganzes Kapitel und argumentiert unter Verweis⁵ auf [FPB95, S. 183]:

The complexity of software is an essential property, not an accidental one. Hence descriptions of a software entity that abstract away its complexity often abstract away its essence. Mathematics and the physical sciences made great strides for three centuries by constructing simplified models of complex phenomena, deriving properties from the models, and verifying those properties experimentally. This worked because the complexities ignored in the models were not the essential properties of the phenomena. It does not work when the complexities are the essence.

Beide Formen der Komplexität – die Komplexität des Auftragsabwicklungsprozesses und die Komplexität der Software, die Modelle von Auftragsabwicklungsprozessen bewertet – sind zu beherrschen, wenn die Aufgabenstellung dieser Arbeit bewältigt werden soll.

Die Unterstützung der Modellierung von Auftragsabwicklungsprozessen ist auf viele Weisen denk- und kategorisierbar. Eine Kategorisierung kann den Formalisierungsgrad als Maßstab nehmen, eine andere die Eignung als Dokumentationsmedium, eine dritte die Möglichkeit zur automatisierten Bewertung durch einen Computer. Im Rahmen dieser Arbeit soll eine vierte Kategorie im Vordergrund stehen: Der Nutzen im Hinblick auf die Entwicklung eines grundlegenden Verständnisses über die Wirkzusammenhänge innerhalb des Auftragsabwicklungsprozesses.

Die oben aufgestellte Behauptung, der Auftragsabwicklungsprozess sei ein komplexer Prozess, gründet sich u.a. auf folgende Faktoren:

- ▷ Der hohen Anzahl von Teilprozessen.
- ▷ Der hohen Zahl von Verbindungen zwischen den Teilprozessen.

⁴SHANNON [Sha75, S. 36] definiert u.a. folgende Charakteristika für komplexe Systeme:

- ▷ **Change.** ... *No real World System remains static over a long period of time. Elements enter and leave the system either through a birth-and-death process or by passing through the system boundary.*
- ▷ **Environment.** *Each system has its own environment and is in fact a subsystem of some broader system. ...*
- ▷ **Counterintuitive behaviour.** ... *Cause and effect are often not closely related in time and space. Obvious solutions may actually intensify a problem rather than solve it.*
- ...
- ▷ **Interdependency.** *No activity in a complex system takes place in total isolation. Each event is influenced by its predecessors and effects its successors. ...*
- ▷ **Organisation.** *Virtually all complex systems consist in highly organized elements or components. Parts combine into hierarchies of subsystems, which interact to carry out the function of the system.*

⁵Booch zitiert nur den ersten Satz.

- ▷ Der hohen Anzahl von Eigenschaften⁶ des Produktes, die in Wechselwirkungen mit den Teilprozessen stehen⁷.
- ▷ Der großen Menge von Aufträgen, die durch den Prozess laufen und wiederum die Teilprozesse beeinflussen.

Wichtige Ziele der Modellierung von Prozessen sind u.a. die Dokumentation und die Schaffung von Transparenz (vgl. [Kuh95, S. 38], [Eve95, S. 37]). Obwohl die Dokumentation von Prozessen die Gewinnung von Transparenz unterstützt, soll der Aspekt der Dokumentation insbesondere die Dokumentation von nicht oder nur schwer formalisierbaren Inhalten (siehe [FvL03, S. 32]) nicht im Vordergrund stehen, da hier bereits zahlreiche, in der Praxis bewährte (z.B. die am Fraunhofer IML entwickelte »Dortmunder Prozesskettennotation«, die auch in dieser Arbeit Verwendung findet⁸) Repräsentationen zur Verfügung stehen. Um die Möglichkeit der Prozessdokumentation zu bieten, wird ein Instrumentarium integriert, welches die letztgenannte Prozesskettennotation zur Dokumentation nutzbar macht (siehe Kapitel 7.5).

Um die Erzeugung von Transparenz über die Dokumentation hinausgehend unterstützen zu können, sollen die Modelle quantitativ bewertbar und experimentierfähig gemacht werden. Das heißt, die Bewertung eines Modells soll Kennzahlen liefern, die im Kontext des zu analysierenden Prozesses verwendet werden. Zudem soll durch leichte Veränderbarkeit wichtiger Prozessparameter die Möglichkeit des Experimentierens geschaffen werden, um Erkenntnisse über das System zu gewinnen. Diese Anforderungen legen eine rechnergestützte Bewertung der Prozesse nahe. Neben einer Methode⁹ zur Beschreibung von Auftragsabwicklungsprozessen ist also Software erforderlich, die die Bewertung der jeweiligen Modelle erlaubt.

Fasst man den Auftragsabwicklungsprozess als »Geschäftsprozess« auf, kommen zu seiner Beschreibung und Bewertung zahlreiche Modellierungsansätze und Werkzeuge für die Interpretation dieser Modelle in Betracht. UTHMANN differenziert diese Werkzeuge in »Simulationstools« und »Geschäftsprozessmodellierungstools« (vgl. [vU01, S. 33]), eine Unterscheidung, die auch von KOPPERGER et al [KMN⁺01] getroffen wird.

Viele dieser Werkzeuge erfüllen die Anforderung, dass alle mittels eines Computers lösbaren Aufgaben in den von den Werkzeugen unterstützten Sprachen formulierbar und berechenbar sind¹⁰. Es folgt, dass die oben geschilderte Modellierungs- und Be-

⁶Unter den »Eigenschaften« eines Auftrages sollen »Ausstattungen« oder »Optionen« verstanden werden, die von den Kunden ausgewählt werden können. Hinzu kommen weitere Eigenschaften, die die Automobilhersteller aus technischen oder sonstigen Gründen hinzufügen. Ein Beispiel für eine kundenwählbare Eigenschaft ist ein Navigationssystem. Der Hersteller könnte z.B. bei Fahrzeugen, die für einen Überseemarkt bestimmt sind, die Eigenschaft »Transportschutz Übersee« hinzufügen. Aus der Menge der Eigenschaften und der Information über den Fahrzeugtyp kann bestimmt werden, welche Teile für die Fertigung eines Fahrzeuges benötigt werden.

⁷Also den Prozesszustand verändern.

⁸Siehe hierzu z.B. [KLS02].

⁹Unter einer *Methode* soll eine »detaillierte und systematische Handlungsvorschrift, wie nach bestimmten Prinzipien ein vorgegebenes Ziel erreicht werden kann«, verstanden werden [SGR97].

¹⁰Programmiersprachen, die diese Eigenschaft besitzen, werden *berechnungsuniversell* genannt (vgl. z.B. [Min67, S. 135], [HU90, S. 156], [Jon87, S. 9]). Die meisten allgemeinen Programmiersprachen besitzen diese Eigenschaft (vgl. [Aab04, S. 9], [Eng88, S. 75]), aber auch viele Petri-Netz basierte Ansätze (vgl. [Jen96], [vdA92, S. 7]), die häufig als Basis für Schemata zur Geschäftsprozessbeschreibung dienen (vgl. [vU01, S. 159]). Für eine Definition von *Sprache* siehe z.B. [HU90, S. 1-2].

wertungsaufgabe, sofern sie überhaupt mit einem Computer gelöst werden kann, mit bisherigen Methoden bereits lösbar ist.

Der Bedarf für einen neuen Ansatz entsteht dadurch, dass die Aufgabe der Modellierung und Bewertung von Auftragsabwicklungsprozessen bislang nicht mit akzeptablem Aufwand bzw. nicht zeitgerecht gelöst werden kann – das Fehlen eines Werkzeugs zur Beantwortung der oben angedeuteten Fragestellungen führte zur Unterstützung der Erarbeitung des hier vorgestellten Ansatzes durch die Volkswagen AG, durch die Nutzung des im Rahmen dieser Arbeit vorgestellten Werkzeugs in zahlreichen Projekten und durch den Zugang zu Informationen über die Prozesse im Bereich der Auftragsabwicklung der Volkswagen AG.

Die Auftragsabwicklung in Automobilunternehmen umfasst viele Teilprozesse, in denen wiederum komplexe Aufgaben durchgeführt werden. Die Gestaltung und Bewertung solcher Prozesse ist nur möglich, wenn diese komplexen Prozesse auch im Computermodell angemessen repräsentiert werden können¹¹. Durch die hohe Zahl von Kundenaufträgen, die in der Praxis – also im Realsystem – auftreten, wird die gestellte Aufgabe weiter erschwert (vgl. Kapitel 3.2.1).

Die Gestaltung eines experimentierfähigen Modells beschränkt sich nicht auf die Abbildung des Auftragsabwicklungsprozesses, sondern erfordert die Erweiterung des Modells um relevante Teile des Umfeldes. Die *Quellen* und *Senken*¹² des Auftragsabwicklungsprozesses sind jeweils die *Kunden* der Automobilhersteller (siehe Abbildung 2.2) und sind ebenso Bestandteil des zu modellierenden Bereiches, wie die Lieferanten des Herstellers¹³.

Um die Anforderungen an einen Gestaltungs- und Bewertungsansatz motivieren zu können, werden in den folgenden Kapiteln Beispiele zur Hilfe genommen, die für den jeweiligen Bereich typische alternative Ausprägungen (vgl. z.B. [Hoo04], [HP04], [Sta01b]) des Auftragsabwicklungsprozesses aufzeigen sollen; diese Aufzählung von Varianten erhebt keinen Anspruch auf Vollständigkeit. Wiederum steht das Ziel im Vordergrund, einen Ansatz zu finden, der den Anforderungen aus den Anwendungsgebieten in Kapitel 3 genügt. Die im Weiteren zitierten konkreten Beispiele wurden stets aus dem Umfeld dieser Anwendungsgebiete entnommen.

4.2. Allgemeine Anforderungen

Allgemeine Anforderungen an die Modellierung von Geschäftsprozessen sind vielfach in der Literatur zu finden ([BRS95], [Kug03], [Sch98d], [Kac02]). Die Grundsätze ordnungsmäßiger Modellierung (GoM) nach BECKER/ROSEMANN/SCHÜTTE, die hier aufgrund der hohen Verbreitung ausgewählt wurden, sind aus den Grundsätzen

¹¹Einige Beispiele hierfür sind in Kapitel 3 aufgeführt; so ist z.B. eine sehr detaillierte Repräsentation von Aufträgen bzw. der Produkte erforderlich, um die dort angesprochenen Fragestellungen beantworten zu können.

¹²*Quellen* und *Senken* stellen die Schnittstellen des Modells zur nicht spezifizierten Umgebung dar und beschreiben die Charakteristika dieser Umgebung soweit erforderlich (vgl. [BBT⁺99, S. 41], [KLS02, S.11]).

¹³Während die *Kunden* als Quellen bzw. Senken des Prozesses stets mit einzubeziehen sind, ist die Abbildung der Lieferanten nur in den Fällen erforderlich, in denen diese relevante Ressourcen für eine Untersuchung darstellen. Die Möglichkeit zur Abbildung der Lieferanten als optionaler Modellbestandteil ist daher notwendig.

ordnungsmäßiger Buchführung abgeleitet und stellen sechs Grundsätze in den Mittelpunkt (vgl. [BRS95, S. 438], [BSGI00, S. 13-17]):

1. **Grundsatz der Richtigkeit.** Ein Modell sollte syntaktisch und semantisch korrekt sein. Die syntaktische Richtigkeit ist gegeben, wenn ein Modell vollständig und konsistent bzgl. des zugrundeliegenden Metamodells ist. Die Forderung nach semantischer Richtigkeit bezieht sich auf das Verhältnis von Modell zum abgebildeten System. Das Modell soll das abgebildete System adäquat repräsentieren.
2. **Grundsatz der Relevanz.** Die Elemente und Beziehungen eines Modells sind genau dann relevant, wenn der Nutzeffekt der Modellverwendung sinken würde, falls das Modell weniger Informationen enthalten würde. Um den Nutzeffekt beurteilen zu können, muss der Zweck der Modellierung vorher festgestellt worden sein.
3. **Grundsatz der Wirtschaftlichkeit.** Modellerstellung und -nutzung wird wirtschaftlichen Kriterien unterzogen.
4. **Grundsatz der Klarheit.** Ein Modell sollte intuitiv verständlich sein (strukturiert, übersichtlich, lesbar).
5. **Grundsatz der Vergleichbarkeit.** Vergleiche zwischen Modellen sollen Vergleiche der abgebildeten Systeme erlauben.
6. **Grundsatz des systematischen Aufbaus.** Modelle, die in getrennten Sichten erzeugt werden, müssen integriert werden können.

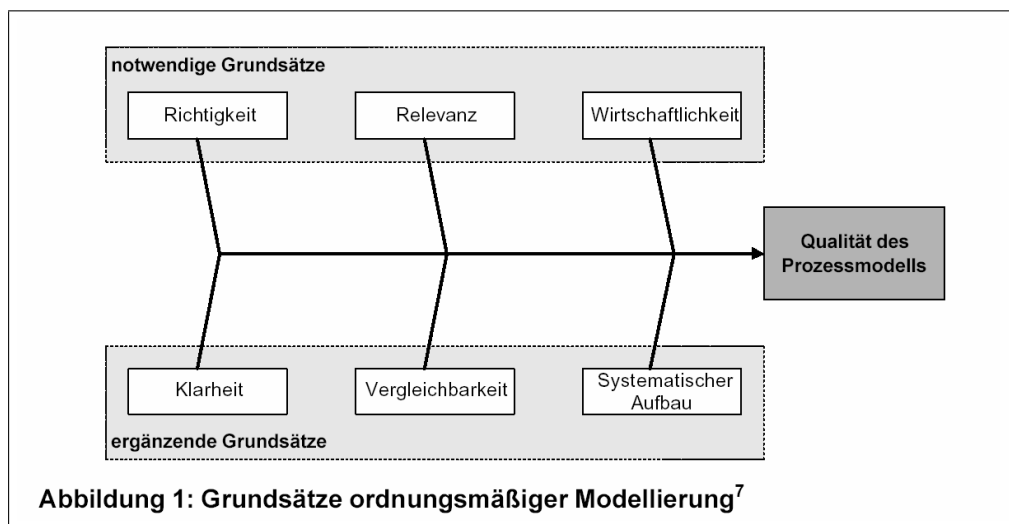


Abbildung 4.1.: Grundsätze ordnungsmäßiger Modellierung (GoM I)

In der obigen Definition der GoM (im Folgenden GoM I) aus dem Jahr 2000¹⁴ werden Modelle als Abbilder der *Realität* gesehen. Dieser *abbildungsorientierte* Modellbegriff (Modelle als Abbilder von Ausschnitten der Realität) wurde von einem der Autoren (SCHÜTTE) der obigen Definition bereits vor Erscheinen des hier zitierten Abschlußberichtes kritisiert (vgl. [Sch99a, S. 1-2], später zusammen mit BECKER, vgl. [SB98]), insbesondere da die Realität nicht subjektunabhängig erkennbar sei ([Sch98c, S. 8]). Die von Schütte vorgeschlagenen »gemäßigt konstruktivistisch« (vgl. [Sch95, S. 213-241], [Sch98c, S. 119-126]) geprägten Grundsätze ersetzen die Punkte »Richtigkeit« und »Relevanz« gegen:

1. **Grundsatz der Konstruktionsadäquanz.** Die Modellkonstruktion muss dem Problem angemessen und nachvollziehbar sein.
2. **Grundsatz der Sprachadäquanz.** Die verwendete Sprache muss zur Beschreibung einer Problemsituation angemessen sein, die syntaktischen Regeln der Sprache sind zu befolgen.

Die Interpretation der verbleibenden Aspekte der GoM I wird in Richtung einer *konstruktionsorientierten* Sichtweise verändert (im Folgenden GoM II). BECKER und SCHÜTTE definieren den Modellbegriff auf Basis dieser Sichtweise (vgl. [BS04, S. 65-67]):

Ein Modell ist die Repräsentation eines Objektsystems eines Originals für Zwecke eines Subjekts. Es ist das Ergebnis einer Konstruktion eines Subjekts des Modellierers, das für eine bestimmte Adressatengruppe Modellnutzer eine Repräsentation eines Originals zu einer Zeit als relevant mit Hilfe einer Sprache deklariert. Ein Modell setzt sich somit aus der Konstruktion des Modellierers, dem Modellnutzer, einem Original, der Zeit und einer Sprache zusammen.

Der verbreitet genutzte Modellbegriff (vgl. z.B. [Tho05, S. 8]) von STACHOWIAK (vgl. [Sta73, S.131-133]) dagegen stützt sich auf drei Merkmale von Modellen: Das *Abbildungsmerkmal*, das *Verkürzungsmerkmal* und das *pragmatische Merkmal*. Insbesondere das Abbildungsmerkmal wird von SCHÜTTE als problematisch gesehen, weil aus konstruktivistischer Perspektive keine Abbildungsrelation im mathematischen Sinne definiert werden kann ([Sch99b, S. 2-4]), aufgrund der nicht intersubjektiven Erkennbarkeit des abzubildenden Systems ([Sch99a, S. 1]).

Der Verzicht auf die abbildungsorientierte Sichtweise auf Modelle bringt in gewisser Hinsicht eine pragmatische Herangehensweise an die Modellierung mit sich. Modelle haben zweckmässig zu sein und den Erwartungen der Modellnutzer entgegenzukommen. Auf der anderen Seite erscheint der praktische Umgang mit der konstruktivistisch geprägten Sicht auf die Erstellung von Modellen umständlich, weil ein Bezugssystem – die »Realität« – fehlt. Das Bezugssystem für den konstruktionsorientiert vorgehenden Modellierer bzw. Modellnutzer ist stets über den Weg einer Konsensbildung herzustellen¹⁵.

Zudem ist beim Sprechen über Modelle stets die Konstruiertheit dieser Modelle zu berücksichtigen, das Denken über »Abbildungen« zu vermeiden. Das dies sehr schwie-

¹⁴Zitiert aus dem Endbericht des Projektes, aus dem die GoM I hervorgegangen sind (vgl. [BSGI00, S. 13-17]). In eben der zitierten Form ist die GoM I aber auch schon 1995 zu finden (vgl. [BRS95, S. 438]).

¹⁵Differenzierte Darstellung in [Sch95, S. 228].

rig ist, kann schon an obiger Modelldefinition von SCHÜTTE und BECKER gesehen werden. Es ist dort von *Repräsentationen* die Rede, die als »strukturerhaltende Abbildungen« [Bro06] definiert werden können. Das heißt, bereits bei der konstruktionsorientierten Definition von »Modellen« ist es SCHÜTTE und BECKER nicht gelungen, den Abbildungsbegriff inhaltlich zu vermeiden.

Um die Intuitivität des abbildungsorientierten Modellbegriffes zu erhalten, ist es erforderlich, die Subjektivität des Betrachters mit der vorausgesetzten Existenz einer »Realität« zusammenzubringen. Dieser Standpunkt des »aufgeklärten Realismus« sei nach SCHÜTTE/BECKER zwar mit den GoM II vereinbar, erlaube aber nicht die Definition von Abbildungen im mathematischen Sinn, eben wegen der Subjektivität der Wahrnehmung.

Von Abbildungen zwischen Original und Modell kann aber dennoch gesprochen werden, wenn die Subjektivität der Betrachter als *Theoriebildung* über das Original aufgefasst wird (vgl. [Gel02], [Sta73, S. 56]). Kommt es im Zuge dieser Theoriebildung zu Widersprüchen, die sich z.B. bei der Arbeit mit Modellen ergeben, können diese im Zuge einer Konsensbildung, z.B. zwischen Modellierer und Modellnutzer, ausgeräumt werden. Das heißt, der Prozess der Konsensbildung erfolgt ganz analog zum konstruktivistischen Ansatz.

Im Folgenden sollen die Grundsätze ordnungsmäßiger Modellierung in der durch SCHÜTTE und BECKER modifizierten Form (also GoM II) genutzt werden, aber dies zusammen mit einem abbildungsorientierten Modellbegriff.

4.3. Kommunikation und Dokumentation

Die Gestaltung von Auftragsabwicklungsprozessen bzw. von Teilprozessen involviert Fachpersonal und Management aus verschiedensten Bereichen. Das verbreitete »Silo-denken« in den Unternehmen erschwert insbesondere die Gestaltung des Auftragsabwicklungsprozesses, da hier ausgesprochen viele Bereiche beteiligt sind bzw. werden müssen.

Die Neigung der von einer Prozessänderung Betroffenen, eher das (vermutete) lokale Optimum in ihrem jeweiligen Bereich zu suchen, denn das Optimum für das Gesamtunternehmen im Auge zu haben, ist ein ernstes Hindernis für die Umsetzung neuer Konzepte. Der vorgeschlagene Ansatz kann zu einem dazu dienen, durch Bewertung von Gestaltungsvorschlägen für einen neuen Prozess Transparenz zu schaffen (Kapitel 4.4), kann aber zum anderen dabei helfen, den »Betroffenen« einen neuen Prozess zu erläutern¹⁶.

Vor der Diskussion über Ergebnisse einer Untersuchung am Modell steht die Kommunikation des neuen Prozesses und die Motivation der Annahme, dass das präsentierte Modell den neuen Prozess so abbildet, dass dieser mit Hilfe des Modells untersucht werden kann. Um eine Diskussion über (Teil-) Prozesse anhand eines Modells überhaupt führen zu können, ist ein gewisser Detaillierungsgrad erforderlich.

Ist es das Ziel, die mittlere Lieferzeit für BTO-Fahrzeuge zu reduzieren (Beispiel aus

¹⁶Ein neuer Prozess ist nicht immer für alle Beteiligten vorteilhaft. Transparenz über die Auswirkungen auf die jeweilig Betroffenen hilft die Auswirkungen zu konkretisieren bzw. sogar zu quantifizieren und schafft somit eine verbesserte Diskussionsgrundlage, um etwaige Vorbehalte aufzulösen (siehe unten).

Kapitel 3.1.1), ist für eine Diskussion auf der Ebene »Top-Management« der Blick auf den Gesamtprozess und die Erreichung der gesetzten Ziele in Bezug auf Gesamtkosten und Gesamtnutzen gerichtet¹⁷. Die Diskussion auf der Ebene der Fachbereiche fokussiert auf die Ausgestaltung des Prozesses im jeweiligen Teilbereich. Soll ein Modell als Grundlage hierfür dienen, müssen die hier relevanten Prozessparameter abgebildet werden. Zusätzlich muss belegt werden, dass die *Systemlast* (siehe [VS02, B5-16]) für einen Teilbereich korrekt modelliert wurde.

Anforderung 1 [GOM: KLARHEIT, SYSTEMATISCHER AUFBAU]

Um bei der Beantwortung einer Fragestellung als Diskussionsgrundlage für die Kommunikation von Prozessalternativen in den Fachbereichen nutzbar zu sein, müssen mindestens folgende Modellelemente vorhanden sein:

1. *Produktaufbau (Modelle in den jeweiligen Märkten, Eigenschaftenfamilien, Eigenschaften, Konstruktionsregeln)*
2. *Einplanung von Aufträgen (Überprüfung von verfügbaren Kapazitäten, Beachtung von Restriktionen für die jeweiligen Werke)*
3. *Fertigung (Beachtung von Materialverfügbarkeit, Ermittlung der Durchlaufzeit abhängig von Modell und Ausstattung)*
4. *Zulieferer (Berücksichtigung von Kapazitäten, Flexibilität, Störungen, Vorlaufzeiten)*
5. *Distribution (Fahrpläne, Kapazitäten, Regeln für die Transportmitteldisposition)*

Entscheidend für die Zusammenstellung dieser Parameter ist es, dass für die jeweils zur Diskussion stehenden Prozessabschnitte die jeweilige Systemlast aus dem Modell abgeleitet werden können muss. Aus Sicht der Werke ist die Zusammensetzung der Eigenschaften in den Aufträgen relevant. Aus Sicht der Distribution sind die Menge und die Verteilung der Fahrzeugmodelle auf die Märkte relevant. Das heißt, die Systemlasten sind jeweils in der angemessenen Form für die Prozessabschnitte zur Verfügung zu stellen.

Anforderung 2 [GOM: SYSTEMATISCHER AUFBAU]

Die Systemlasten für Teilprozesse des Auftragsabwicklungsprozesses müssen aus dem Modell abgeleitet (berechnet) werden können.

4.4. Bewertung der Ergebnisse

Sowohl für den Einsatz als Bewertungsansatz für die Prozessgestalter als auch für die Nutzung zur Kommunikation mit den Betroffenen einer Prozessänderung, ist Akzeptanz für die Ergebnisse von höchster Bedeutung. Mittel zur Erreichung dieser Akzeptanz sind die Nachvollziehbarkeit der Ergebnisse und die Möglichkeit, die für Teilprozesse relevanten Sichten auf den Gesamtprozess zu bieten.

¹⁷Die Frage, wie es zur Bewertung von Prozessalternativen kommt, ist im Vorfeld zu klären. Das heißt, ein Fokus auf dem Gesamtprozess bringt nicht notwendig eine Reduktion des Detaillierungsgrades der Betrachtung mit sich.

Wird z.B. die »Leistung« eines Werkes an seiner Auslastung bemessen, muss die Auslastung des Werkes auch für den zukünftigen Auftragsabwicklungsprozess ausweisbar sein. Prozessveränderungen, wie in Kapitel 3.1.1 beschrieben, können zu einer stark veränderten Situation im Hinblick auf die Werksauslastung führen. Eine stärker marktorientierte »Fahrweise« eines Werkes führt zu einer stärker schwankenden Auslastung. Diese Erkenntnis lässt sich ohne Weiteres aus dem Erfahrungswissen des Fachpersonals ableiten und eben dieses Erfahrungswissen erlaubt diesem Personenkreis die Prognose, dass die Prozessveränderung eine Verschlechterung einiger Kennzahlen im Bereich »Werk« zur Folge hat. Ein Ansatz, der die Akzeptanz für einen veränderten Prozess erhöhen soll, muss also die Effekte auf die für den Teilprozess relevanten Kennzahlen quantifizieren und somit die Gestaltung neuer Bewertungsmaßstäbe unterstützen.

Anforderung 3 [GOM: SYSTEMATISCHER AUFBAU]

Die Bewertung eines neuen Prozessentwurfs anhand von relevanten Kennzahlen muss sowohl für den Gesamtprozess (ganzheitliche Betrachtung) als auch die jeweiligen Teilprozesse ermöglicht werden. Das heißt, die Abbildung der Teilprozesse muss so detailliert erfolgen können, dass eine angemessene Beurteilung der Auswirkungen auf die betrachteten Teilprozesse möglich wird.

Die in den Unternehmen genutzten Kennzahlen sind nur schwer vollständig und unternehmensübergreifend in einem Analyse-Werkzeug in vordefinierter Form abzulegen. Neben zwischen den Unternehmen unterschiedlichen Benennungen werden viele Kennzahlen nicht einmal innerhalb eines Unternehmens einheitlich genutzt. Ein Analysewerkzeug sollte dieser Tatsache Rechnung tragen und die Möglichkeit vorsehen, Auswertungen durch den Anwender definierbar zu machen.

Anforderung 4 [GOM: KLARHEIT, KONSTRUKTIONSDÄQUANZ, VERGLEICHBARKEIT]

Liegen für ein konkretes Unternehmen einheitlich definierte Kennzahlen in Bezug auf den Auftragsabwicklungsprozess vor, sollten diese Definitionen in analoger Weise für Analysen des Modells zur Verfügung stehen.

Aus dieser Anforderung leiten sich Anforderungen an das Datenmodell ab, da die Eingangsgrößen für die Kennzahlen entweder direkt verfügbar sein oder zumindest abgeleitet werden können müssen. Die Möglichkeit, den »Lebenszyklus¹⁸« eines jeden Auftrages im Anschluss an eine Bewertung durch ein Werkzeug zu betrachten, ist einerseits in der Analysephase nützlich, um gewisse Effekte¹⁹ leichter nachvollziehen zu können (z.B. durch Auflistung aller Aufträge, die eine bestimmte Lieferzeit überschreiten) und schafft andererseits Vertrauen in die Ergebnisse²⁰.

¹⁸Unter dem *Lebenszyklus* eines Auftrags soll die Zeitspanne von der Auslösung der Beauftragung durch den Kunden bis zur Auslieferung eines Fahrzeugs verstanden werden – der OTD-Prozess. Die Auftragsabwicklung wird auf die kundenrelevanten Aspekte fokussiert betrachtet. Aus Sicht der jeweiligen Hersteller kann der Lebenszyklus, z.B. aus technischen Gründen, anders definiert sein.

¹⁹Ergebnisse, die eine tiefergehende Analyse erfordern.

²⁰»Vertrauen« in die Ergebnisse ist eine subjektive Größe, für dessen Erreichung die Validierung der Ergebnisse nur ein Baustein ist. So spielt z.B. die Nachvollziehbarkeit der Ergebnisse eine wichtige Rolle.

Anforderung 5 [GOM: KLARHEIT, KONSTRUKTIONSDÄQUANZ, VERGLEICHBARKEIT]

Der Lebenszyklus eines Auftrags sollte detailliert nachvollziehbar sein (z.B.: Welche Ausstattung eines Fahrzeugs wurde von einem Händler bestellt, welche Ausstattung wünschte der Kunde, inwieweit konnte der Auftrag an den Kundenwunsch angepasst werden, welchen Status hatte der Auftrag zum Zeitpunkt der »Übernahme« durch einen Kunden, wann wurde der Auftrag gefertigt, welchen Transportweg hat das Fahrzeug zum Kunden genommen?).

4.5. Systemlast

Dass die Abbildung der Aufträge von zentraler Bedeutung für die Modellierung des Auftragsabwicklungsprozesses ist, bedarf keiner weiteren Begründung. Der Wunsch eines Kunden, der letztlich zur Erteilung eines Auftrags führt, lässt sich durch den gewünschten Fahrzeugtyp und die Liste der gewünschten Ausstattungen charakterisieren. Unter der *Systemlast* soll die Menge aller Kundenaufträge verstanden werden, die vom Auftragsabwicklungsprozess zu bearbeiten sind.

Die Anforderungen an diese Systemlast sind stark abhängig von der jeweils untersuchten Fragestellung und dem Abstraktionsgrad, der gewählt wurde, um die Fragestellung zu beantworten. Als wesentliche Kostentreiber innerhalb der Automobilunternehmen stehen die Ressourcen des Auftragsabwicklungsprozesses (z.B. die Werke [OLCB01, S. 9]) häufig im Fokus von Untersuchungen. Die Systemlast ist so zu gestalten, dass diese Ressourcen »realistisch« belastet werden, also dass zum Beispiel die Verteilung von Aufträgen auf Werke derjenigen im abzubildenden Systems ausreichend nahe kommt. In den folgenden Abschnitten werden Bedingungen abgeleitet, die hierzu zu erfüllen sind.

4.5.1. Quellen für Aufträge

Die Erzeugung von im jeweiligen Kontext korrekten Aufträgen ist unabdingbar, um qualitativ hochwertige Aussagen über das Zusammenspiel zwischen Bedarf und Kapazität im Rahmen der Auftragsabwicklung zu machen (vgl. Kapitel 3.2.1). Diese Aufträge können für eine Untersuchung entweder extern zugeführt werden (z.B. »echte« Kundenaufträge), oder für die jeweilige Aufgabestellung erzeugt werden. Die externe Bereitstellung von Aufträgen kommt für Fragestellungen in Betracht, die auf Basis einer Bedarfssituation aus der Vergangenheit oder unmittelbaren Zukunft²¹ beantwortet werden können.

Dem Charakter nach sind jedoch Untersuchungen über den Auftragsabwicklungsprozess in die Zukunft gerichtet – Aufträge aus der Vergangenheit können somit nur mit starken Einschränkungen²² verwendet werden.

²¹ Aufträge von Kunden liegen i.d.R. in ausreichender Anzahl nur für wenige Wochen in die Zukunft vor (siehe Kapitel 2.1). Ausnahmen sind Hersteller, die über volle Auftragsbücher mit einem längeren Horizont verfügen können. Für Volumenmodelle tritt dieser Fall nicht mehr auf.

²² Die Überführung von Aufträgen, die gemäß den Regeln einer Produktstruktur erzeugt, gegen tausende Regeln geprüft wurden (s.u.) und einer Menge von Einbauraten entsprechen, in eine andere Produktstruktur mit anderen Konstruktionsregeln und anderen Einbauraten ist nicht weniger komplex als die Neuerzeugung der Aufträge gemäß den Vorgaben der Ziel-Produktstruktur. Nur in sehr einfachen

Anforderung 6 [WERKZEUG]

Es ist also die Anforderung zu stellen, dass ein Werkzeug für den gewünschten Einsatzzweck im Rahmen einer Simulation Aufträge erzeugen kann.

4.5.2. Baubarkeit von Aufträgen

Automobilhersteller definieren in den Katalogen, die den Kunden überreicht werden, implizit eine Menge von Regeln, die die Menge der von Kunden wählbaren Konfigurationen eines Auftrags einschränkt. So lässt sich bei einem Hersteller nennenswerter Größe nicht jeder von diesem Hersteller angebotene Motor mit jeder Karosserie kombinieren²³.

Die Gestaltung von Verkaufsprospekten bzw. von Produktkonfiguratoren für Automobile berücksichtigt offenbar vor allem Marketingaspekte; die Formulierung eindeutiger Konfigurationsregeln für Aufträge²⁴ steht nicht im Vordergrund. Als Startpunkt für die Auftragskonfiguration werden von den Herstellern i.d.R. Kombinationen aus Baureihe/Karosserieform/Motorisierung ausgewählt. Da die Menge der zulässigen Motor/Karosserie-Kombinationen für eine Baureihe überschaubar ist, können diese leicht explizit aufgelistet werden.

Mit dieser Auswahl einer »Grundvariante« ist ein Auftrag aus Sicht des Herstellers vollständig beschrieben, da aus jeder *Eigenschaftenfamilie* (z.B. Radios, Getriebe, Sitze) eine Eigenschaft gewählt ist: Die *Grundausrüstung*. Diese Konfiguration kann nun durch die Änderung von Eigenschaften weiter an den jeweiligen Kundenwunsch²⁵ angepasst werden. Jede Veränderung dieser (Grund-) Konfiguration sollte wiederum zu einem gültigen Zustand²⁶ führen.

Für jede Eigenschaft, die durch einen Kunden ausgewählt werden kann, wird vom Hersteller eine Liste von Einschränkungen verwaltet. Diese Regeln haben die Form Eigenschaft A *nicht mit* oder aber *nur mit* Eigenschaft B. Typische Beispiel: Basismotorisierung *nicht mit* Automatikgetriebe, V12-Motor *nur mit* Automatikgetriebe. Die große Anzahl verfügbarer Eigenschaften in Kombination mit einer großen Menge verschiedener Motor- und Karosseriekombinationen führen zu hunderten Regeln, denen ein Auftrag genügen muss, um baubar zu sein.

In einem Großkonzern kann die Anzahl der Regeln über die gesamte Produktpalette mehrere zehntausend betragen (vgl. [GDM03, S. 1]). Wie bereits in Kapitel 1 ange-

Fällen können durch Modifikation bestehender Aufträge nutzbare neue Aufträge erzeugt werden.

²³Diese und alle weiteren Feststellungen zur Auftragskonfiguration in diesem Kapitel lassen sich durch Experimentieren mit den im Internet verfügbaren Produktkonfiguratoren nachvollziehen. Diese Produktkonfiguratoren können auf den Web-Seiten der jeweiligen Hersteller gefunden werden (z.B. <http://www.daimlerchrysler.de>, <http://www.bmw.de>, <http://www.volkswagen.de>).

²⁴Für eine formale Definition von Aufträgen siehe Kapitel 6.

²⁵Unter *Kundenwunsch* soll diejenige Konfiguration verstanden werden, die der Kunde »eigentlich« erwerben möchte. Dieser Kundenwunsch ist allerdings von höchst virtuellem Charakter, da er nicht verlässlich ermittelt werden kann. Der Händler verändert durch die Führung des Verkaufsgespräches diesen Wunsch in Richtung einer Konfiguration, die tatsächlich geliefert werden kann bzw. für den Händler besonders profitabel ist. Ähnliche Einschränkungen gelten für die Auftragskonfiguration am Computer. Hier ist zwar kein beeinflussender Händler anwesend (zumindest nicht zwingend), aber durch die Vorgehensweise bei der Auftragskonfiguration werden bereits viele nicht lieferbare Eigenschaftskombinationen ausgeschlossen.

²⁶Einer Konfiguration, die den Regeln des jeweiligen Herstellers für Aufträge entspricht.

deutet, hat die im Rahmen dieses Regelwerks korrekte Spezifikation von Aufträgen großen Einfluss auf die Qualität von Aussagen, die über einen Auftragsabwicklungsprozess getroffen werden können. An einem praktischen Beispiel soll diese Notwendigkeit verdeutlicht werden.

Beispiel 4.1 *Es sollen die Auswirkungen auf die Versorgungssituation einer vorgezogenen Markteinführung der Basismotorisierung eines Modells auf einem Überseemarkt untersucht werden. Als kritisch bekannt gelte die Versorgungslage für Getriebe. Soll eine Aussage über die Versorgungslage gewonnen werden, muss der durch die vorgezogene Markteinführung veränderte Bedarf bestimmt werden. Auf Basis der Regel »Basismotorisierung nicht mit Automatikgetriebe« lässt sich einfach ableiten, dass der Bedarf für Schaltautomaten von der Markteinführung nicht beeinflusst wird. Wird hingegen diese Regel nicht berücksichtigt, scheint dieser Bedarf zu steigen – eine offensichtlich fehlerhafte Aussage.*

Es existieren also Fragestellungen in Bezug auf den Auftragsabwicklungsprozess, die es erforderlich machen, Aufträge im oben dargestellten Sinne regelkonform zu erzeugen. Sobald Kapazitäten von Lieferanten Bestandteil eines Modells werden, kann auf die Regelkonformität von Aufträgen nicht verzichtet werden, da etwaige Ressourcenengpässe Einfluss auf die Einplanung von Aufträgen bzw. die Möglichkeit zur termingerechten Fertigung nehmen.

Anforderung 7 [GOM: KONSTRUKTIONSADEQUANZ; WERKZEUG]

Die Aufträge im Modell des Auftragsabwicklungsprozesses dürfen keine der Konstruktionsregeln für Aufträge verletzen.

4.5.3. Verteilung von Eigenschaften auf Aufträge

Die oben formulierte Anforderung nach Regelkonformität²⁷ der Aufträge ist ein Schritt auf dem Weg zu einer realistischen Ressourcennutzung in einem Simulationsmodell. Eine weitere Anforderung ist die realistische Verteilung von Eigenschaften auf die Aufträge, d.h. der Anteil von Aufträgen mit einer bestimmten Eigenschaft (z.B. 35 % haben ein Schiebedach) muss den Vorgaben aus dem Modell entsprechen.

Die Anteile von Aufträgen bzw. Fahrzeugen, die mit bestimmten Eigenschaften ausgestattet sind, fallen abhängig vom Markt höchst unterschiedlich aus, da die Automobilhersteller teilweise speziell auf einen Markt zugeschnittene Verkaufsprogramme betreiben, die bestimmte landesspezifische Vorlieben der Kunden reflektieren. Die Ressourcenbelastung im Realsystem entsteht durch die Mischung von Aufträgen aus den jeweiligen Märkten und ist im Modell insofern nachzubilden, dass die Belastungen der als im Modell relevant erachteten Ressourcen nachgebildet werden. Auf die Zusammenstellung von Eigenschaften in einem Auftrag wird an verschiedenen Stellen des Auftragsabwicklungsprozesses Bezug genommen:

1. **Auftragseinplanung:** Bei der Einplanung von Aufträgen kann die Verfügbarkeit der gewünschten Eigenschaften geprüft werden²⁸. Ist mindestens eine der Eigen-

²⁷Bezogen auf Konstruktionsregeln.

²⁸Eine solche Prüfung kann auf Basis von Eigenschaften durchgeführt werden (nicht mehr als x Fahrzeuge mit einer bestimmten Eigenschaft pro Tag oder Woche) oder auf der Basis von Teilen, die benötigt werden, um Fahrzeuge mit einer bestimmten Eigenschaft zu fertigen.

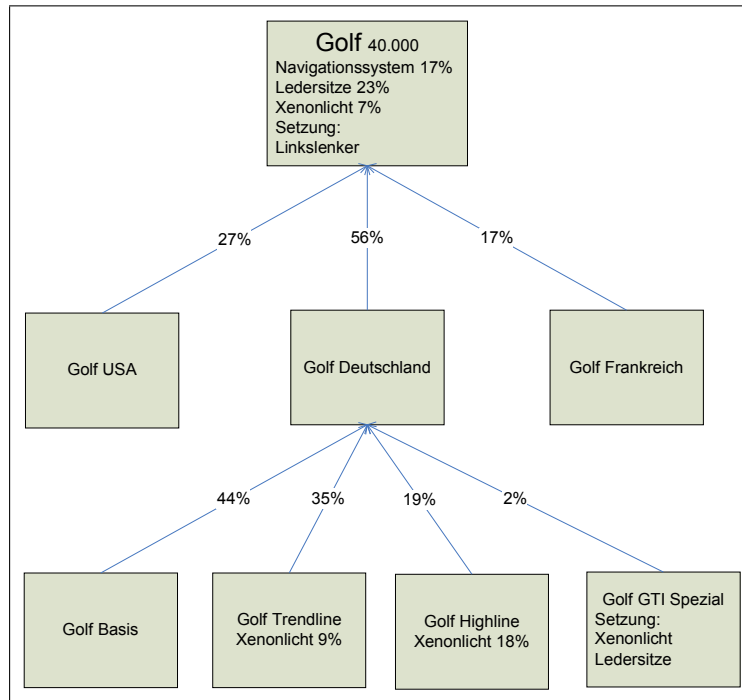


Abbildung 4.2.: Anteile von Eigenschaften auf Märkten

schaften nicht verfügbar, wird eine Einplanung zu einem früheren oder späteren Zeitpunkt geprüft. Die Verteilung von Eigenschaften auf Aufträge wirkt also auf die Verteilung von Aufträgen im Werk.

2. **Berechnung Teilebedarfe:** Die Bestimmung, ob ein bestimmtes Teil in einem Automobil verbaut wird, wird durch die Zuordnung von einem Eigenschaftsmuster zu einem Teil erreicht. Das heißt, ein Teil wird in einem Fahrzeug verbaut, wenn eine Menge von Eigenschaften in einem Auftrag vermerkt ist. Die Teilebedarfe im Realsystem und wünschenswerterweise auch im Modell des Realsystems hängen also von der Häufigkeit bestimmter Eigenschaftskombinationen in den Aufträgen ab (vgl. [GDM03, S. 2]).
3. **Auftragsanpassung:** Erscheint ein Kunde bei einem Händler mit einem Fahrzeugwunsch, wird der Händler nach einem Blick in seinen Auftragsbestand zunächst versuchen, ein bereits bestelltes Fahrzeug an den Wunsch des Kunden anzupassen. Ob eine solche Anpassung möglich ist, wird entscheidend von den Regeln zur Auftragsanpassung bestimmt. Die Eigenschaften eines Auftrags bestimmen also, zu welchem Maß dieser Auftrag an den Kundenwunsch angepasst werden kann.

Die genannten Anforderungen zur Auftragsgenerierung sind zu erfüllen, wenn die zu bearbeitende Fragestellung eine realistische Verteilung von Eigenschaften auf Aufträge erfordert, wie z.B. das in Kapitel 3.2.1 genannte Beispiel. In vielen Fällen ist eine weniger detaillierte Repräsentation von Aufträgen ausreichend. So ist im Zusam-

menhang mit Fragestellungen zur Distribution von Fahrzeugen unter Umständen eine Beschränkung auf wenige Eigenschaftsfamilien ausreichend.

Anforderung 8 [GOM: SPRACHADÄQUANZ]

Die Repräsentation von Aufträgen muss flexibel an den Anwendungs- bzw. Untersuchungszweck anpassbar sein.

Erforderliche Eingangsdaten für die Erzeugung von Aufträgen sind neben den Konstruktionsregeln für die Aufträge die Einbauraten. Werden Einbauraten für eine Untersuchung aus Vergangenheitsdaten abgeleitet, können also »echte« Aufträge genutzt werden. Einbauraten lassen sich sehr spezifisch für alle Eigenschaften, für jedes Modell in jeder Motorisierung einer Baureihe angeben, auf jedem Markt, auf dem das Modell verkauft wurde (siehe Beispiel 4.2). Liegen die Vergangenheitsdaten nicht vor, z.B. wenn die Einführung einer neuen Baureihe Untersuchungsgegenstand ist, wird die Information abstrakter vorliegen, wie in Beispiel 4.3²⁹ für eine beschränkte Zahl Eigenschaften. Informationen über die Entwicklung von Einbauraten einiger »kritischer« Eigenschaften werden z.B. durch die Vertriebsorganisationen der Hersteller bereitgestellt und dienen als Planungsbasis bis zum Vorliegen konkreterer Informationen³⁰.

Beispiel 4.2 *Einbauraten »Navigationssystem« 12,7 % für Baureihe: »Audi A4«, Karosserieform »Limousine«, Ausstattungslinie: »GLS«, Motor: »1,8«, Markt: »Frankreich«.*

Einbauraten »Feuerlöscher« 43,2 % für Baureihe: »Audi A4«, Karosserieform »Limousine«, Ausstattungslinie: »GLS«, Motor: »1,8«, Markt: »Frankreich«.

Beispiel 4.3 *Einbauraten »Navigationssystem« 11,9 % für Baureihe: »Audi A4«, Markt: »Frankreich«.*

Einbauraten »Feuerlöscher« 43,2 % für Baureihe: »Audi A4«.

Die Informationen über Einbauraten können für ein Modell unterschiedlich spezifisch vorliegen und sollten in dieser Form von der Software, die den Gestaltungs- und Bewertungsansatz umsetzt, »akzeptiert« werden. Aus diesen Eingangsdaten – statistische Informationen über »historische« oder zukünftige Aufträge – müssen Aufträge erzeugt werden, die für die Bewertung eines Modells als »Systemlast« dienen.

Anforderung 9 [GOM: SPRACHADÄQUANZ]

Die Eingangsdaten für die Auftrags erzeugung sollten sowohl sehr detailliert als auch sehr abstrakt spezifizierbar sein.

²⁹Eine detailliertere Prognose der Einbauraten erfordert das Abschätzen von Tausenden von Einbauraten und ist sowohl aufgrund des Aufwands als auch aufgrund der hierdurch kaum verbesserten Prognosegenauigkeit nicht zweckmäßig. Ursache für die geringere Prognosegenauigkeit ist u.A., dass die Anzahl der Fahrzeuge, für die diese Prognose angefertigt wird, geringer wird, je spezifischer der Prognosegegenstand wird. Werden z.B. von einem Fahrzeugmodell einer bestimmten Karosserieform mit einer bestimmten Motor/Getriebekombination in einem Markt zwei Stück pro Woche abgesetzt, ist es nur schwer möglich, die Wochentage für die Fertigung dieser zwei Fahrzeuge korrekt zu prognostizieren.

³⁰Zum Beispiel durch das Eintreffen von Kundenaufträgen für einen prognostizierten Zeitraum.

Anforderung 8 verlangt die Möglichkeit zur flexiblen Festlegung der Auftragsstruktur. Anforderung 9 verlangt, dass die Eingangsdaten, die für die Erzeugung der Aufträge genutzt werden, in unterschiedlichen Detaillierungsgraden spezifiziert werden können sollen³¹.

Anforderung 10 [WERKZEUG; GOM: SPRACHADÄQUANZ]

Die durch Analyse der vom Verfahren erzeugten Aufträge bestimmten Einbauraten müssen die Einbauraten im Modell in einer für den jeweiligen Einsatzzweck ausreichenden Genauigkeit annähern.

Neben den spezifizierten Einbauraten sind bei der Erzeugung der Aufträge auch die in Kapitel 4.5.2 angesprochenen Konstruktionsregeln zu beachten. Ist z.B. im Beispiel 4.3 im Markt Frankreich für das Basismodell der Einbau eines Navigationssystems nicht zulässig³², muss die Einbauraten für Navigationssysteme in den anderen Modellen erhöht sein, da über die Gesamtheit aller A4 im Markt Frankreich die Einbauraten 11,9 % beträgt.

Die oben formulierte Anforderung 7 beeinflusst also zum einen die Erzeugung eines einzelnen Auftrags und bewirkt zum anderen u.U. eine Veränderung von Einbauraten für andere Fahrzeugklassen.

4.6. Abbildung von Werken

Eine wesentliche Kennzahl bei der Bewertung von Auftragsabwicklungsprozessen ist eine im Hinblick auf die Kosten optimale Auslastung der Werke eines Unternehmens (vgl. [OLCB01, S. 9]). Vorbedingung für eine Bewertung der Auslastung ist das Vorliegen von Informationen über die Verteilung von Aufträgen über die Zeit. Die Verteilung von »echten« Aufträgen im Realsystem entsteht durch Regeln, die von den jeweiligen Unternehmen genutzt werden, um die Aufgabe zu erfüllen.

Diesen Regeln liegt ein Zielsystem zu Grunde, welches abhängig vom Automobilhersteller sehr unterschiedlich gestaltet sein kann und wesentlich von den Charakteristika der Märkte des jeweiligen Herstellers und des bzw. der Produkte(s) bestimmt wird. So kann z.B. bei einem Unternehmen die kurzfristige Befriedigung eines Kundenwunsches im Fokus stehen; bei einem anderen Unternehmen die gleichmäßige Auslastung der Fertigung.

Ein Hersteller im Premiumsegment mit hohem BTO-Anteil kann aufgrund der hohen Variationsbandbreite der Aufträge kaum gleichartige Aufträge zusammenfassen³³. Obwohl die Produktion großer Mengen gleichartiger Produkte für Premiumhersteller kaum möglich ist, gibt es dennoch einen gewissen Spielraum für die Einplanung eines Auftrags, um zumindest eine größtmögliche Gleichverteilung der Belastung in den Werken und bei den Zulieferern zu erreichen. Wird dem Kunden ein Liefertermin in Form eines Intervalls (z.B. eine Lieferwoche) avisiert, ist es möglich, zwischen mehreren Tagen einen solchen auszuwählen, an dem das Fahrzeug unter logistischen Gesichtspunkten am günstigsten zu fertigen ist³⁴. Ein Hersteller mit niedrigem BTO-

³¹Zum Beispiel kann der Absatz für eine ganze Modellreihe für ein Jahr spezifiziert sein, oder aber für jeden Markt, auf dem diese Modellreihe verkauft wird, oder für jeden Tag.

³²Die Regel lautet also »Navigationssystem nicht im Basismodell«.

³³Zum Beispiel 500 rote Fahrzeuge in der Ausstattung »GLS« für den Markt Frankreich.

³⁴Der Spielraum zur Optimierung wird durch eine »ungenau« Aussage in Richtung Kunde erkaufte.

Anteil kann eine gleichmäßige Auslastung der Ressourcen durch Produktion großer Lose identisch konfigurierter Fahrzeuge erreichen³⁵.

Die unterschiedlichen Strategien der Automobilhersteller zur Einplanung von Aufträgen beeinflussen offensichtlich Kenngrößen wie Lieferzeiten und Bestände bereits gefertigter Fahrzeuge.

Anforderung 11 [GOM: SPRACHADÄQUANZ; WERKZEUG] *Aussagen zu Kenngrößen wie Lieferzeiten und Bestände lassen sich nur dann ableiten, wenn die Algorithmen³⁶ zur Einplanung ebenso geeignet abgebildet werden können (siehe Kapitel 4.6.2) wie die Ressourcen, auf denen diese Algorithmen arbeiten.*

4.6.1. Ressourcen

Nimmt der Automobilhersteller einen Auftrag zur Einplanung entgegen, entscheidet die Qualität der Informationen über die Verfügbarkeit der für den Auftrag benötigten Ressourcen über Faktoren wie Termintreue und Kosten. Können z.B. Teile von einem Zulieferer nicht in geplanter Menge geliefert werden, werden u.U. einige Aufträge nicht wie geplant gebaut³⁷. Diese Umplanung von Aufträgen führt jedenfalls zur verzögerten Fertigstellung der betroffenen Aufträge.

Weiterhin wird die geplante Reihenfolge von Teileabrufen durcheinander gebracht. Die im abzubildenden System gemessenen Diskrepanzen zwischen geplanter Liefermenge und tatsächlicher Liefermenge sind ein Maß für die Informationsqualität in diesem Bereich des Auftragsabwicklungsprozesses.

Anforderung 12 [GOM: SPRACHADÄQUANZ] *Die Abweichungen zwischen angenommener (geplanter) und tatsächlicher Kapazität einer Ressource müssen im Modell abgebildet werden können.*

Eine mögliche Strategie, um mit schlechter Informationsqualität umzugehen, ist z.B. der Aufbau von Beständen. Ein »guter« Auftragsabwicklungsprozess trägt dazu bei, die benötigten Bestände zu minimieren. Ist obige Anforderung erfüllt, lässt sich die Wirkung einer solchen Strategie am Modell prüfen.

Für die Ressource »Personal« ist die Informationsqualität bei der Modellierung weniger problematisch, da der Hersteller diese Ressource zumeist selbst steuert³⁸. Informationen über Störungen der Personalverfügbarkeit, wie z.B. Streiks, sind nahezu

³⁵Die Fertigung von Honda ist z.B. auf die Herstellung einer möglichst großen Anzahl identischer Fahrzeuge ausgerichtet (Quelle: »The Detroit News« Online Archiv, 22.4.2004).

³⁶AHO, ULLMAN ([AU92, S. 5] definieren einen Algorithmus:

An algorithm is a precise and unambiguous specification of a sequence of Steps that can be carried out mechanically.

³⁷In der Regel wird die Fertigung eines Auftrags nur dann begonnen, wenn von einer Verfügbarkeit aller benötigten Teile ausgegangen werden kann. Ausnahmen sind Teile, die nachgerüstet werden können. Ist z.B. das vom Kunden gewünschte Sportlenkrad nicht verfügbar, kann das Fahrzeug zunächst mit einem anderen Lenkrad ausgerüstet und produziert werden. Diese Nachrüstung kann in einigen Fällen kostengünstiger sein, als die Umplanung von Aufträgen.

³⁸Ausnahmen sind Fälle, in denen die Montage extern geleistet wird von Firmen wie Karmann, Magna Steyr oder Valmet.

sofort verfügbar und werden in die aktuelle Planung eingearbeitet. Daher ist die Informationsqualität in diesem Bereich leichter sicherzustellen als bei der Teileversorgung (siehe Kapitel 4.7.).

4.6.2. Algorithmen

Neben der Abbildung von Daten, wie z.B. Informationen über die Produktstruktur³⁹, Lieferbeziehungen und verfügbaren Kapazitäten, muss ein Ansatz zur Gestaltung und Bewertung von Auftragsabwicklungsprozessen die Möglichkeit bieten, die planerischen Aktivitäten innerhalb des Prozesses abzubilden und einer Bewertung zugänglich zu machen. Erst die detaillierte Abbildung von solchen Algorithmen erlaubt es, Modelle von Auftragsabwicklungsprozessen so zu erstellen, dass daraus Rückschlüsse auf die dynamische Entwicklung eines Realsystems in der angestrebten Weise möglich werden. So setzt z.B. die Disposition von Kapazitäten wichtige Rahmenbedingungen für die Auftragsabwicklung und beeinflusst zahlreiche Kennzahlen des Prozesses.

Einplanung von Aufträgen

Für die Einplanung von Aufträgen ist eine Vielzahl von Verfahren bei Automobilherstellern im Einsatz bzw. wird für einen möglichen Einsatz untersucht (vgl. z.B. [Hoo04, S. 7]). Die Unterschiede resultieren aus der unterschiedlichen Positionierung im Markt, sind aber auch auf die Schwierigkeiten bei der Anpassung etablierter Prozesse zurückzuführen (siehe Kapitel 1). Beispiele für Unterschiede in der Einplanung sind:

1. Einplanung in einen zentralen Auftragspool oder direkt in ein Werk. Verfügt ein Hersteller über mehrere Werke, in denen ein Auftrag gefertigt werden kann, besteht die Möglichkeit, den Auftrag direkt einem Werk zuzuordnen, oder aber den Auftrag zunächst in einem zentralen Auftragspool zu belassen und zu einem späteren Zeitpunkt ein Werk auszuwählen.
2. Einplanung auf Basis von Monaten, Wochen oder Tagen. Der Termin für die Fertigung kann bereits bei der Einplanung tagesgenau festgelegt oder aber zunächst einer Fertigungswoche oder einem Fertigungsmonat zugeordnet werden.
3. Der Bezugspunkt für die Einplanung bzw. Verwaltung von Aufträgen kann Fertigungsbeginn, Fertigungsende oder der Kundenliefertermin sein.
4. Die Kapazitätsprüfung bei der Einplanung kann gegen vertraglich zugesicherte Kapazitäten der Lieferanten oder gegen von den Lieferanten zeitnah erstellte Kapazitätsaussagen erfolgen.
5. Bei Einplanung »kurz« vor Fertigung des Auftrags können zusätzliche Regeln die Ressourcenverfügbarkeit weiter einschränken. So kann beispielsweise die Schwankungsbandbreite zwischen einzelnen Tagen limitiert werden.
6. Für gewisse Aufträge können Prioritätsregeln gelten. Für einen exklusiven Kundenkreis können Aufträge zu einem Zeitpunkt in die Auftragsreihenfolge eingeschoben werden, an dem dies für »normale« Aufträge nicht mehr möglich wä-

³⁹Für eine Definition von »Produktstruktur« siehe Kapitel 6.2.1.

re. Mit einer solchen abweichenden Behandlung sichert sich der Hersteller die Möglichkeit, bei Bedarf besonders kurze Lieferzeiten für BTO-Aufträge zu realisieren.

Dieser Ausschnitt der Unterschiede⁴⁰ zwischen den Einplanungsverfahren für Aufträge deutet bereits darauf hin, dass eine Repräsentation dieser Unterschiede in einem Modell nicht vernachlässigt werden kann, ohne die Aussagekraft des Modells stark zu beeinträchtigen. Dies gilt insbesondere in den Fällen, in denen die oben genannten Unterschiede gerade den Untersuchungsgegenstand bilden.

Anforderung 13 [GOM: SPRACHADÄQUANZ] *Die Einplanung von (Kunden-) Aufträgen muss flexibel an unterschiedliche Vorgaben anpassbar sein. Diese Flexibilität umfasst z.B.*

- ▷ *die Festlegung der Bestellzeitpunkte (wann bestellen Kunden bzw. Händler),*
- ▷ *die Form der Kapazitätsprüfung (werden Einplanungen gegen die tatsächlichen Zuliefererkapazitäten geprüft),*
- ▷ *Priorisierungsregelungen für bestimmte (z.B. verspätete) Aufträge oder*
- ▷ *Priorisierungsregelungen für die Einplanung in die Werke (welches Werk wird unter welchen Bedingungen gewählt).*

Berechnung Auftragsstatus

Eingeplanten Aufträgen können zahlreiche Status zuerkannt werden. Zum Beispiel kann ein Auftrag entweder noch an einen Kundenwunsch angepasst werden oder eben nicht (der Auftrag ist »eingefroren«). Der Auftrag bekommt im ersten Fall den Status »änderbar«, im zweiten Fall den Status »nicht änderbar«. Die Zeitpunkte, an denen der Status geändert wird (Statuspunkte), werden genutzt, um die Aufträge den jeweiligen Beständen zuzurechnen; im konkreten Fall dem »änderbaren Auftragsbestand« bzw. dem »nicht änderbaren Auftragsbestand«. Planungsverfahren können die Ableitung eines Auftragsstatus bzw. einer Statushistorie erfordern, um die Rahmenbedingungen für bestimmte Aufgaben festzulegen. Zum Beispiel können im Fall eines Lieferengpasses Aufträge bevorzugt behandelt werden, die bereits zuvor durch einen Engpass verzögert wurden.

Anforderung 14 [GOM: SPRACHADÄQUANZ] *Für die Entscheidungsfindung im Modell muss die Ableitung des gegenwärtigen Auftragsstatus und der Auftragshistorie (z.B. welche Prozesse wurden bis zu einem Zeitpunkt durchlaufen) unterstützt werden. Informationen über das Durchschreiten von Statuspunkten durch einzelne Aufträge sollen für die Berechnung von Kennzahlen zugänglich sein.*

⁴⁰Da eine vollständige Aufzählung aller möglichen Auftragseinplanungsverfahren nicht möglich ist, werden im Beispiel einige typische Unterschiede im Hinblick auf die Aufgabenstellung in Kapitel 3 aufgezählt.

Störungsmanagement

Abweichungen der tatsächlichen von der geplanten Ressourcenverfügbarkeit (Störungen) können in einem Auftragsabwicklungsprozess an vielen Punkten auftreten (siehe Kapitel 3.3.2). Diese Abweichungen können als kurzfristige Schwankungen im Bereich von wenigen Prozenten auftreten, die z.B. durch vorhandene Bestände ausgeglichen und somit kaum sichtbar werden, oder im anderen Extrem als wochenlanger Ausfall einer Ressource, die nicht ersetzt werden kann.

Um die »Robustheit« eines Auftragsabwicklungsprozesses gegen die genannten Abweichungen prüfen zu können, sollen zwei Modellierungsansätze gefordert werden:

Anforderung 15 [GOM: SPRACHADÄQUANZ] *Störungen sollen »explizit« als vom Modellierer spezifizierte Abweichung der tatsächlichen von der geplanten Kapazität spezifiziert werden können, wie auch als stochastischer Einfluss.*

Die Spezifizierung stochastischer Einflüsse ist prinzipiell auch mit der ersten geforderten Methode durch eine entsprechende Modelländerung möglich. Das heißt, der Modellierer berechnet stochastisch veränderte Kapazitäten. Da dies offensichtlich sehr umständlich ist, werden zwei Methoden zur Störungsmodellierung gefordert.

Kapazitätsverwaltung und -anpassung

Die verfügbare Kapazität eines Werkes wird vom Automobilhersteller in kurz-, mittel- und langfristigen Planungszyklen an den prognostizierten Bedarf angepasst (siehe Kapitel 2.2.1).

Die Langfristplanung mit einem Horizont von mehreren Jahren ordnet Baureihen einzelnen Werken zu und beeinflusst Entscheidungen über den Neubau bzw. die Schließung von Standorten (siehe Kapitel 3.1.2). Neben den Kapazitäten des Automobilherstellers werden indirekt auch die Kapazitäten der Zulieferer beeinflusst. Auf Basis der langfristigen Kapazitätsplanung des Automobilherstellers werden von der für die Beschaffung zuständigen Teileinheit des Herstellers Verhandlungen mit Lieferanten geführt, die daraufhin ihrerseits eine Kapazitätsplanung durchführen und – bei positivem Ausgang der Verhandlungen – Fertigungseinrichtungen aufbauen oder sogar Werke in einem Lieferantenpark des Automobilherstellers errichten. Die langfristige Planung setzt also den Rahmen für die mittel- und kurzfristige Planung.

Die mittelfristige Kapazitätsplanung nimmt Einfluss auf Personal- bzw. Schichtplanung und kann bereits auf Basis von Daten aus der Programmplanung eine Kapazitätsplanung durchführen. Die Verlagerung der Produktion in ein anderes Werk wird dabei i.d.R. nicht mehr möglich sein. Dies auch wegen der mit den Zulieferern getroffenen Vereinbarungen. Kapazitätsveränderungen in größerem Rahmen lassen sich durch Veränderungen der Schichtanzahl erreichen, u.U. durch Verhandlungen mit den Tarifpartnern. Im kurzfristigen Bereich können Änderungen an der Arbeitszeit nur innerhalb der tariflich vereinbarten Korridore durchgeführt werden.

Die Abstimmungsprozesse innerhalb dieser »Kapazitätsnetzwerke« sollen im Weiteren nur insoweit in den Modellierungsansatz integriert werden, als dass kurzfristige Anpassungsmaßnahmen Berücksichtigung finden. Zum einen können die Umfeldparameter, die für die langfristige Kapazitätsplanung eine Rolle spielen, wie z.B. Änderungen der politischen Rahmenbedingungen an einem Werksstandort, nur schwer in

ein Simulationsmodell integriert werden; zum anderen sind Fragestellungen, wie die Errichtung eines neuen Werkes, nicht im Kontext des Auftragsabwicklungsprozesses angesiedelt.

Das heißt, die Entscheidungen der langfristigen Kapazitätsplanung sollen nicht *automatisiert* in einem Modell getroffen werden (müssen)⁴¹. Sehr wohl können aber die *Auswirkungen* einer solchen Entscheidung mittels des hier vorgestellten Ansatzes bewertet werden. Die Auswirkungen der Entscheidung – also z.B. die Errichtung eines neuen Werksstandortes – werden in das Modell integriert, die Entscheidungsfindung aber nicht. Im kurzfristigen Bereich hingegen:

Anforderung 16 [GOM: SPRACHADÄQUANZ] *Entscheidungen der kurzfristigen Kapazitätsplanung sollen automatisiert im Modell ableitbar sein, da diese Entscheidungen als Bestandteil des Auftragsabwicklungsprozesses aufgefasst werden können.*

4.7. Abbildung von Zulieferern

Zulieferer sind für die Bewertung eines Auftragsabwicklungsprozesses von hoher Bedeutung, da ein wesentlicher Teil der Dynamik des Prozesses aus der Diskrepanz zwischen geplanter Teilekapazität und tatsächlicher Teilekapazität und daraus resultierenden Verschiebungen in der Auftragswarteschlange resultiert. Die Kapazität eines Zulieferers sowie die Flexibilität, diese Kapazität anzupassen, haben nachhaltigen Einfluss auf die Fähigkeit des Automobilherstellers, auf Störungen in der Teileversorgung in einer Weise zu reagieren, die für Kunden keine oder nur geringe negative Auswirkungen hat.

Eine hohe Flexibilität des Zulieferers ist einerseits vorteilhaft im Hinblick auf die Erreichung einer guten Liefertreue, auf der anderen Seite aber kann diese Flexibilität ein Kostentreiber sein – Fertigungsanlagen und Personal werden vom Zulieferer für den *eventuellen* Einsatz vorgehalten. Es kommt also auf die richtige Balance zwischen Flexibilität und Kosten an. Eine »realistische« Abbildung dieser Flexibilität im modellierten Auftragsabwicklungsprozess ist ein Schlüssel zur vergleichenden Bewertung der Kosten und resultierenden Prozesskennzahlen, die entstehen, wenn bestimmte Vorgaben im Hinblick auf die Flexibilität mit den Zulieferern vereinbart werden⁴².

Ebenso wie die Kapazität der Automobilhersteller wird die Kapazität der Zulieferer durch die »eigenen« Kapazitäten einerseits und durch die Kapazitäten der Lieferanten⁴³ andererseits limitiert. Für die Gestaltung und Bewertung des Auftragsabwicklungsprozesses sind insbesondere die resultierenden Kapazitäten des 1st-Tier-Lieferanten⁴⁴ von Belang, daher ist es hinreichend, eben diese Kapazitäten »realistisch« abzubilden⁴⁵.

⁴¹ Anders formuliert: Die langfristige Kapazitätsplanung ist nicht Bestandteil des betrachteten Realsystems.

⁴² Die Flexibilitätsvorgaben schlagen sich nahe liegend in den Preisen nieder, die der Lieferant in Rechnung stellt.

⁴³ Hier: Die Lieferanten der Lieferanten.

⁴⁴ Ein Lieferant, der den Automobilhersteller unmittelbar beliefert. Dessen Lieferant wiederum, der Rohmaterial für die Fertigung der Türinnenverkleidung herstellt, wird als *2nd-Tier-Lieferant* bezeichnet, usw..

⁴⁵ Durch diese Beschränkung werden viele interessante Fragestellungen ausgeschlossen, wie z.B. die Frage nach Beständen bei den Lieferanten, die erforderlich sind, um ein bestimmtes Flexibilitätsmaß zu

Zulieferer können genau ein oder auch mehrere Werke mit einem Teil beliefern; mehrere Zulieferer können das gleiche Teil an ein Werk liefern. Auch bei einer Beschränkung der Abbildung auf 1st-Tier Lieferanten ist es daher erforderlich, die verfügbare Kapazität nach geeigneten Regeln auf die jeweiligen Abnehmer zu verteilen. Der Verteilungsschlüssel kann sich von den prognostizierten Bedarfen ableiten, d.h. beliefert ein Unternehmen mehrere Werke, wird im Falle einer Verknappung dasjenige Werk eine größere Zuteilung erhalten, das im Vorfeld einen größeren Bedarf angemeldet hat.

Anforderung 17 [GOM: SPRACHADÄQUANZ] *Die Zuteilung von Kapazitäten durch einen Lieferanten soll sich an den prognostizierten Bedarfen der jeweiligen Kunden ausrichten.*

Der Verlauf der Bedarfe über die Zeit für *ein* Teil ist offenbar proportional zur produzierten Anzahl Fahrzeuge, zudem ist sowohl der Automobilhersteller als auch der Zulieferer an einer gleichmäßigen Einbauhäufigkeit für ein Teil interessiert, da in beiden Fällen Anlagen bzw. Personal so gleichmäßig ausgelastet werden⁴⁶.

Da Eigenschaften nicht stochastisch unabhängig über die Aufträge verteilt sind, sondern nach bestimmten marktabhängigen Kundenpräferenzen zusammengestellt worden sind, kann die Gleichverteilung auf Basis einer Eigenschaft zu einer Ungleichverteilung einer anderen führen. So führt auch eine »bestmögliche« Gleichverteilung zu einer Schwankung in den Teilebedarfen, die die Schwankung der Anzahl der Aufträge übersteigen kann.

Im Interesse einer kostenoptimalen Fertigung ist der Zulieferer daran interessiert, diese Schwankungen in seiner Fertigung zu vermeiden und ist folglich gezwungen, Bestände aufzubauen.

Anforderung 18 [GOM: SPRACHADÄQUANZ] *Die Modellelemente zur Abbildung von Zulieferern sollen es ermöglichen, Nachfrageschwankungen, die die Fertigungskapazität in einem Zeitraum überschreiten, durch Maßnahmen, wie z.B. Vorproduktion oder erforderlichenfalls Nachproduktion, auszugleichen.*

4.8. Abbildung der Distribution

Transporte von Gütern spielen in Bezug auf die Auftragsabwicklung an mindestens zwei Stellen eine Rolle: Zum einen beim Transport der gefertigten Fahrzeuge vom Werk zum Kunden und auf der anderen Seite bei der Versorgung der Produktionswerke mit Teilen. Erstere Rolle ist direkter Bestandteil des im Fokus stehenden Prozesses und ist somit notwendiger Bestandteil eines Modells des Auftragsabwicklungsprozesses. Die Teileversorgung als angrenzender Prozess (siehe Abbildung 2.2) setzt wesentliche Rahmenbedingungen für die Auftragsabwicklung und soll daher ebenfalls betrachtet werden.

erreichen. Diese Fragestellungen lassen sich aber durch eine Erweiterung des vorgestellten Ansatzes um weitere Elemente der Lieferkette erreichen (siehe Kapitel 9).

⁴⁶Für einige besonders aufwändig zu verbauende Teile werden zum Teil Obergrenzen für die Häufigkeit in der Auftragssequenz spezifiziert. Beispiel: »Nur jedes sechste Fahrzeug mit Schiebedach«.

4.8.1. Distribution gefertigter Fahrzeuge

Die Aufgabenstellungen im Bereich der Distribution der Fahrzeuge von den Werken zu den jeweiligen Zielorten umfassen eine Anzahl von Teilaufgaben, die unter Berücksichtigung sowohl der Wünsche der Kunden nach zeitgerechter Lieferung, als auch des Wunsches des Automobilherstellers nach einer möglichst kostengünstigen Abwicklung dieser Transportaufgabe zu bewältigen sind. Der Aspekt der kostengünstigen Abwicklung gewinnt besondere Relevanz unter Beachtung der Kapitalbindungskosten, die bei hochwertigen Fahrzeugen die Transportkosten insbesondere bei Überseetransporten übersteigen können. Weiterhin können die Bestandskosten für Fahrzeuge, die noch keinen Kunden gefunden haben, diesem Prozessabschnitt zugeordnet werden (vgl. Kapitel 3.1.1). Maßnahmen zur Senkung dieser Kosten sind aber nur teilweise in diesem Prozessabschnitt angesiedelt.

Die Transportzeit übersteigt in vielen Fällen die Dauer für die eigentliche Fertigung der Fahrzeuge, insbesondere, wenn die Fahrzeuge über Landesgrenzen hinweg transportiert werden müssen und ist daher eine nahe liegende Größe, die für eine etwaige gewünschte Lieferzeitverkürzung betrachtet werden kann. Über diese einfachen Zusammenhänge hinaus, wirken die Distributionsprozesse aber auch zurück in vorgelegte Prozessabschnitte. So kann z.B. eine nicht mehr ausreichende Lagerfläche in einem Markt eine Änderung der Programmplanung nach sich ziehen. Weiterhin kann die Positionierung von Lagerflächen darüber entscheiden, ob Kunden mit einem Lagerfahrzeug oder aber mit einem Build-to-Order-Fahrzeug versorgt werden können.

Die Distributionsprozesse stehen nicht immer im Fokus, wenn Veränderungen am Auftragsabwicklungsprozess zu bewerten sind. Die Abbildung der Distributionsprozesse im Modell muss es jedoch auch in diesem Fall erlauben, Aussagen zur Lieferzeit und -treue zu ermöglichen, muss also hinreichend detailliert sein, um eine Durchlaufzeit für den Distributionsprozess für individuelle Fahrzeuge so abzuleiten, dass die gewünschte Qualität der Aussagen erreicht werden kann. Ist beispielsweise ausschließlich die Lieferzeit zu genau einem Importeur die Fragestellung, kann die Abbildung eines Distributionsweges ausreichend sein. Sind die Verteilungsprozesse in der Fläche Bestandteil der Fragestellung, müssen u.U. hunderte Distributionswege modelliert werden.

Steht die Distribution im Fokus der Betrachtung, ist wahrscheinlich eine detaillierte Abbildung sowohl der Distributionswege als auch der Dispositionsregeln erforderlich.

Anforderung 19 [GOM: SPRACHADÄQUANZ] *Die Abbildung des Distributionsnetzwerkes für gefertigte Fahrzeuge muss die Repräsentation der jeweils relevanten Transportwege, Transportmittel und Dispositionsverfahren für Transportmittel erlauben, bis hin zur detaillierten Abbildung aller Distributionskanäle eines Automobilherstellers.*

4.8.2. Distributionsprozesse in der Teileversorgung

Die Zuverlässigkeit, mit der Teile an die jeweiligen Verbrauchs- bzw. Lagerorte transportiert werden, entscheidet sowohl über erforderliche Bestandshöhen als auch, wenn sich die Bestände als nicht ausreichend erweisen, über Liefertreue und Lieferzeit, die von einem Automobilhersteller erreicht werden. Weiteren Einfluss auf erforderliche Bestände hat die Form der Belieferung. Werden Teile täglich oder nur wöchentlich

geliefert, ist eine Just-in-Time oder Just-in-Sequence Belieferung vorgesehen (vgl. [Gra00, S. 7], [VA02, S. 18-19])?

Über Belieferungsform, Lagerort, gewünschte Bestandshöhen, Art der Bestandsverwaltung usw. müssen vor dem Anlauf der Fertigung eines Automobils Entscheidungen getroffen werden, die in Summe wesentlichen Einfluss auf die Sicherheit der Teileversorgung und in Folge dessen auf die Auftragsabwicklung haben. Neben der potenziellen Verschlechterung von Kennzahlen der Auftragsabwicklung wirken Probleme in der Teileversorgung auch in die Lieferkette zurück:

Eine Veränderung der Auftragsreihenfolge, die durch einen Versorgungsengpass ausgelöst wird, verändert auch den Teilebedarf im Vergleich zur Situation mit unbeeinflusster Reihenfolge. Die »Unruhe«, die durch einen Engpass hervorgerufen wird, kann neue Teileengpässe nach sich ziehen.

Aufgrund des hohen Kostendrucks, unter dem die Zulieferindustrie steht, wird häufig eine Verlagerung der Produktion in Länder betrieben, die günstigere Lohnkosten aufweisen. Die veränderten Distributionswege setzen neue Rahmenbedingungen sowohl für die Planungsprozesse vor der Produktion als auch für die Produktion selbst.

Aufgrund der in vielen Fällen höheren Entfernung ausländischer Zulieferer-Standorte⁴⁷, werden die Möglichkeiten der Reaktion auf kurzfristige Änderungen des Fertigungsprogramms reduziert: Beträgt die Transportzeit z.B. für auftragsbezogen gefertigte Kabelbäume drei Tage, ist die »Reaktionszeit« für eine Veränderung der Auftragsreihenfolge eben min. drei Tage⁴⁸, wenn nicht nur die Reihenfolge von Aufträgen geändert werden soll, für die der Kabelbaum gesichert verfügbar ist.

Neben dieser Einschränkung der Flexibilität der Planung steigt mit der zu überbrückenden Entfernung auch die Wahrscheinlichkeit, dass es zu Störungen in der Transportabwicklung kommt, wodurch die Fertigung des Automobilherstellers beeinträchtigt werden kann, wenn nicht durch entsprechend erhöhte Bestände Vorsorge getroffen wurde.

Anforderung 20 [GOM: SPRACHADÄQUANZ] *Die Abbildung des Distributionsnetzwerkes der Teileversorgung (»Inbound-Distribution«) muss die Repräsentation der jeweils relevanten Transportwege und -mittel ermöglichen. Für die Ableitung der erforderlichen Transportvolumina im Modell müssen Behälter für Teile zusammen mit Angaben zu deren Fassungsvermögen und Ausmaßen angegeben werden können.*

4.9. Modellgröße, Laufzeit

Der hohe Detaillierungsgrad der Modelle, die zur Bewertung von Auftragsabwicklungsprozessen genutzt werden, bringt hohe Anforderungen an die Software mit sich, die für die Bewertung der Modelle genutzt wird. Wenn einige Millionen Aufträge während der Bewertung erzeugt werden⁴⁹ und jeder dieser Aufträge wiederum aus über

⁴⁷Gemeint ist hier die Belieferung deutscher Werke durch in das Ausland verlagerte Zulieferbetriebe.

⁴⁸Zu dieser Transportzeit addiert sich die Zeit vom Auftragsingang beim jeweiligen Zulieferbetrieb bis zur Verladung der Teile.

⁴⁹Unter der Annahme, dass die Betrachtung einzelner Aufträge erforderlich ist, um Aussagen über den Auftragsabwicklungsprozess abzuleiten (vgl. 5).

hundert Eigenschaften zusammengesetzt ist⁵⁰, entstehen Datenmengen, deren Beherrschung eine wichtige Rahmenbedingung bei der Gestaltung einer Bewertungslösung ist.

Anforderung 21 [GOM: WIRTSCHAFTLICHKEIT; WERKZEUG] *Es muss die Bewertung von Modellen von Auftragsabwicklungsprozessen möglich sein, in denen mehrere Millionen Aufträge mit jeweils hunderten von Eigenschaften Gegenstand der Betrachtung sind.*

Neben der Möglichkeit, »große« Modelle überhaupt bewerten zu können, spielt natürlich auch die dafür benötigte Zeit eine wichtige Rolle für die praktische Anwendbarkeit eines Bewertungsansatzes. Während bei Fragestellungen im Hinblick auf die strategische Ausrichtung eines Unternehmens eine Bewertungsdauer von z.B. 24 Stunden wahrscheinlich akzeptabel ist, wenn eine hohe Ergebnisqualität dagegen steht, kann für die Unterstützung operativer Fragestellungen eine Bewertungsdauer von 20 Minuten bereits zu lang sein.

Eine »harte« Anforderung in Form einer Obergrenze für die Bewertungsdauer ist daher nur schwer anzugeben, zumal der im Rahmen dieser Arbeit entwickelte Prototyp (siehe Kapitel 7) die zahlreichen Möglichkeiten zur Parallelverarbeitung von Daten nur rudimentär nutzt. Trotzdem soll eine Anforderung zumindest an die Dimension der Laufzeit gestellt werden, um eine wirtschaftliche Einsetzbarkeit des Ansatzes zu erlauben.

Anforderung 22 [GOM: WIRTSCHAFTLICHKEIT; WERKZEUG] *Ein Werkzeug zur Bewertung von Auftragsabwicklungsprozessen soll ein Modell mit einer Millionen Aufträge mit jeweils über 100 Eigenschaftenfamilien in unter 24 Stunden bewerten können.*

Ein Berechnungszeitraum von 24 Stunden sollte im strategischen oder taktischen Umfeld akzeptabel sein, mit 100 Eigenschaftenfamilien eines Produktes sind auch sehr hohe Anforderungen an den Detaillierungsgrad zu erfüllen. Selbst bei großen Automobilherstellern läßt sich entweder durch die Beschränkung des Untersuchungszeitraumes, durch Betrachtung nicht aller Fertigungswerke, durch Beschränkung auf eine Baureihe oder einer Marke, die Anzahl der abzubildenden Aufträge unter einer Million halten.

Da bei operativen Fragestellungen, aufgrund des hier kurzen betrachteten Zeitabschnittes, sehr viel geringere Auftragsanzahlen eine Rolle spielen, kann die Erfüllung obiger Anforderung auch eine Nutzbarkeit für operative Fragestellungen mit sich bringen. Da diese Anforderungen bereits von der prototypischen Umsetzung erfüllt werden sollen, ist eine Verbesserung der Leistung in Bezug auf die Laufzeit ohne Weiteres möglich.

⁵⁰Gebhardt gibt an, dass der Volkswagen Golf durch die Festlegung jeweils einer Eigenschaft aus ca. 200 Eigenschaftenfamilien konfiguriert wird, wobei jede dieser Eigenschaftenfamilien zwischen vier und acht Eigenschaften enthalte (vgl. [GDM03, S. 1]).

4.10. Automatisierbarkeit des Modellaufbaus

Modelle, die die oben formulierten Anforderungen erfüllen, sind zwangsläufig sehr umfangreich. Buchstäblich tausende von Fahrzeugklassen mit jeweils hunderten von Eigenschaften sind dabei nicht nur für die Berechnung eines Modells eine Herausforderung, sondern auch für den Aufbau der Modelle. Viele der zugrunde liegenden Daten ändern sich häufig, so dass ein Modell, welches zur Unterstützung operativer Aufgaben erstellt wurde, nur eine Gültigkeit von wenigen Tagen hat⁵¹. Der wirtschaftliche Einsatz solcher Modelle ist daher leichter möglich, wenn der Modellaufbau weitestgehend automatisch erfolgen kann. Die Anforderung der Automatisierbarkeit gilt in verstärktem Maß, wenn ein Modell nicht nur zur einmaligen Beantwortung einer Fragestellung genutzt wird, sondern zur permanenten Planungsunterstützung.

Nahezu alle Schritte der Bearbeitung eines Auftrags, von der Aufnahme des Kundenwunsches bis hin zur Auslieferung des Fahrzeugs an den Kunden, können bei den größeren Automobilherstellern von Computersystemen unterstützt werden. Eine Nutzung dieser Daten ist für den Aufbau eines Modells des Auftragsabwicklungsprozesses nahe liegend. Für den operativen Einsatz des Gestaltungs- und Bewertungsansatzes kann der automatische Modellaufbau bzw. mindestens die automatische Modellaktualisierung⁵² als unabdingbar angesehen werden, da für den operativen Einsatz besonders hohe Anforderungen bezüglich der Aktualität der verwendeten Daten zu stellen sind.

Anforderung 23 [GOM: WIRTSCHAFTLICHKEIT; WERKZEUG] *Der Aufbau bzw. die Wartung eines Computermodells des Auftragsabwicklungsprozesses sollte weitestgehend automatisch erfolgen können.*

4.11. Zusammenfassung

Die oben formulierten Anforderungen können zu folgendem Anforderungsprofil zusammengefasst werden:

1. [GOM: SPRACHADÄQUANZ] Es müssen für die Abbildung von Auftragsabwicklungsprozessen geeignete Elemente, z.B. zur Beschreibung der Produktstruktur, von Werken oder von Zulieferern, vorhanden sein.
2. [GOM: KLARHEIT und KONSTRUKTIONSDÄQUANZ] Es muss möglich sein, die Auftragsabwicklung detailliert und nachvollziehbar abzubilden, um z.B. Kennzahlen analog zu denen im abzubildenden System am Modell berechnen zu können.
3. [GOM: WIRTSCHAFTLICHKEIT] Beim Modellaufbau müssen auch sehr große Datenmengen, möglichst ganz- oder halbautomatisch, in das Modell aufgenommen werden können. Die Zeit zur Modellbewertung (Rechenzeit des Simulationswerkzeugs) muss möglichst kurz sein.

⁵¹Jede Änderung der dem Modell zugrunde liegenden Daten (eine Änderung im »Realsystem«) kann dazu führen, dass die Aussagen, die aus dem Modell gewonnen werden, nicht mehr gültig sind.

⁵²Unter Modellaktualisierung soll hier verstanden werden, dass im Modell bereits vorhandene Daten, wie z.B. Absatzzahlen, lediglich verändert nicht aber neu erzeugt werden.

4. [GOM: VERGLEICHBARKEIT] Die detaillierte Nachvollziehbarkeit des Prozessdurchlaufs einzelner Aufträge und die möglichst direkte Übernahme von Kennzahldefinitionen aus dem abgebildeten System, sollen die Vergleichbarkeit (im Sinne der GoM) von Modellen erlauben.
5. [WERKZEUG] Das Werkzeug muss den Aufbau umfangreicher Modelle erlauben, z.B. durch Unterstützung eines (halb-) automatisierten Aufbaus und muss zudem diese Modelle in einer angemessenen Zeit bewerten können.

Im folgenden Kapitel soll untersucht werden, welcher Modellierungsansatz genutzt werden kann und welches Bewertungswerkzeug in Frage kommt, um mit Hilfe der Modelle die jeweiligen Fragestellungen zu beantworten.

5. Abgleich des Anforderungsprofils mit existierenden Ansätzen

Es kann also das Ziel formuliert werden, einen Modellierungsansatz zu finden, der einerseits die gestellten Anforderungen erfüllt und für den andererseits bereits ein IT-Werkzeug am Markt zur Verfügung steht, welches eine Bewertung von Modellen von Auftragsabwicklungsprozessen erlaubt, die unter Nutzung dieses Ansatzes erzeugt wurden.

Gelingt es nicht, eine solche existierende »integrierte« Lösung zu finden, ist es erforderlich, eine solche zu schaffen, z.B. indem existierende »Bausteine«, wie z.B. Modellierungssprachen und Bewertungswerkzeuge, geeignet zusammengestellt und erweitert werden. Im Folgenden soll zunächst festgelegt werden, wie Modelle von Auftragsabwicklungsprozessen abgebildet werden können, um im Anschluss aufzuzeigen, auf welcher Grundlage ein Bewertungswerkzeug umgesetzt werden kann.

Um Modelle von Auftragsabwicklungsprozessen in verschiedenen Ansätzen bewerten zu können, muss die Semantik¹ so definiert sein, dass eine maschinengestützte Analyse ermöglicht wird (vgl. [BBT⁺99, S. 4]). Auf der anderen Seite sollten die Modelle auch Anwendern der Anwendungsdomäne verständlich sein – Programmiersprachen² sind daher zu ihrer Erstellung allein nicht geeignet (vgl. [FP97, S. 13]).

5.1. Auswahl eines Modellierungsansatzes

Um eine Modellierungssprache und im Anschluss ein Bewertungswerkzeug auswählen zu können, ist zunächst zu bestimmen, welche Modellierungsansätze für eine weitere Betrachtung in Frage kommen. Dazu ist zunächst die Frage zu klären, ob die Bewertung von Auftragsabwicklungsprozessen durch Nutzung *analytischer Methoden* möglich ist. Eine analytische Lösung einer Aufgabe macht *alle* Systemzustände transparent³ und ist somit – soweit verfügbar – zu bevorzugen.

Allerdings sind analytische Methoden nur für die Untersuchung hinreichend einfacher Modelle⁴ geeignet. LAW und KELTON schätzen ein:

However, most real-world systems are too complex to allow realistic models to be evaluated analytically, and these models must be studied by means of simulation.

¹Semantik ist die Lehre von der inhaltlichen Bedeutung einer Sprache und ihrer Zeichen. In der Informatik beschreibt die Semantik die Beziehung zwischen einer Anweisung in einer Programmiersprache und der Zustandsänderung, die ein Computer ausführen soll (vgl. [Bro02]).

²Auch Sprachen, die zur Beschreibung von Unternehmensprozessen entworfen wurden, können als Programmiersprachen aufgefasst werden. Gemeint sind hier allerdings »übliche« Programmiersprachen, wie z.B. Java oder C++.

³Zum Beispiel können Verklemmungen (Deadlocks) erkannt werden (vgl. [BBK01, S. 4], [Tep04, S. 15], [GSS02, S. 1282]).

⁴Siehe hierzu [LK91, S. 1], [GNS03, S. 1013], [AM02, S. 1302], [JGG⁺02, S. 1306], Beispiele für den Einsatz analytischer Methoden siehe [FD03], [DV03], [BM02].

Für die vorliegende Aufgabe kommen analytische Methoden als umfassender Lösungsansatz also nicht in Betracht, da hier gerade sehr komplexe Fragestellungen untersucht werden sollen. *Simulation* hingegen kann zur Analyse von Modellen genutzt werden, die einer analytischen Analyse nicht oder aber mit vertretbarem Aufwand zugänglich sind (vgl. [Sch97, S. 6]). Die Vor- und Nachteile der Methode »Simulation« werden in der Literatur sehr ausführlich beschrieben und sollen an dieser Stelle nicht noch einmal wiederholt werden (vgl. u.a. [LK91], [BCN96], [ZPK00], [Pag91]). Stellvertretend wird hier die VDI-Richtlinie 3633 zitiert (vgl. [VDI93, S. 2]), in der als Gründe für den Einsatz von Simulation genannt werden:

- ▷ Steigende Produktkomplexität und Variantenvielfalt,
- ▷ steigende Anforderungen an Qualität,
- ▷ steigende Anforderungen an Servicezeit und -grad,
- ▷ schnellere Produktwechsel,
- ▷ kleinere Losgrößen und
- ▷ steigender Kostendruck.

Als wesentliche Vorteile der Methode werden an selber Stelle die Möglichkeiten gesehen, zur Untersuchung

- ▷ real (noch) nicht existierender Systeme,
- ▷ real existierender Systeme ohne direkten Betriebseingriff,
- ▷ mehrerer Gestaltungsvarianten bei geringem Arbeitsaufwand,
- ▷ des Systemverhaltens über lange Zeiträume hinweg (Zeitraffung)
- ▷ und von Anlaufvorgängen, Einschwingphasen und Übergängen zwischen definierten Betriebszuständen.

Zur Auswahl eines Simulationsansatzes soll das Klassifizierungsschema aus VDI-Richtlinie 3633 Blatt 1 genutzt werden (siehe Tabelle 5.1).

Kriterium	Ausprägung
Experimentierbarkeit	experimentierbar / nicht experimentierbar
Beschreibungsmittel	symbolisch / physisch / hybrid / formal, abstrakt / gedanklich / graphisch, textuell
Beschreibungsart	digital / analog
Beschreibungsmedium	Software / ...
Zufallsverhalten	stochastisch / deterministisch
Zeitverhalten	dynamisch / statisch
Zeitablauf	kein Zeitablauf / diskret / quasikontinuierlich / kontinuierlich
Zeitablaufsteuerung	ereignisgesteuert / prozessorientiert / aktivitätsorientiert / transaktionsorientiert / zeitgesteuert

Tabelle 5.1.: Klassifizierung von Modellen (Quelle: [VDI93, S.14])

Die ersten Festlegungen sind offensichtlich: Modelle sollen **experimentierfähig**, **textuell**⁵, **digital** und als **Software** repräsentierbar sein.

Modelle, die keine Zufallsvariablen enthalten, werden als *deterministisch* bezeichnet (vgl [BCN96, S. 12]). Eine Menge von Eingabedaten erzeugt stets dieselbe Ausgabe. *Stochastische Modelle* dagegen enthalten Zufallsvariablen. Die Ausgabe eines stochastischen Modells wird durch Nutzung von stochastischen Verteilungen berechnet⁶.

Zahlreiche Vorgänge innerhalb des Auftragsabwicklungsprozesses unterliegen stochastischen Einflüssen. Störungen in der Materialverfügbarkeit machen es erforderlich, Aufträge, für die die benötigten Materialien nicht vollständig verfügbar sind, zurückzustellen. Diese tatsächliche Materialverfügbarkeit weicht von der geplanten Verfügbarkeit, z.B. aufgrund von durch die jeweilige Verkehrslage verzögerten LKW-Lieferungen, ab. Es ist daher sinnvoll, die Belieferung eines Werkes als stochastischen Prozess in einem Modell abzubilden. Es sollte also ein Ansatz ausgewählt werden, der die Abbildung **stochastischer** Prozesse ermöglicht.

Statische Modelle repräsentieren Systeme zu einem bestimmten Punkt in der Zeit; dynamische Modelle repräsentieren die Veränderung eines Systems über die Zeit (vgl. [BCN96, S. 12], [MRW04, S. 27-29]). Die Verwendung eines statischen Ansatzes schließt die Berücksichtigung einer Veränderung über die Zeit aus. Bezieht sich eine Fragestellung über einen Zeitraum (z.B. »Tritt in den nächsten zwei Monaten ein Engpass in der Teileversorgung ein«), kann ein statisches Verfahren mit Mittelwerten über diesen Zeitraum zu einer Antwort gelangen. Im gewählten Beispiel ist die Tragfähigkeit einer Aussage über Mittelwerte begrenzt (siehe auch [BG97b, S. 30]).

Liegt der über zwei Monate berechnete Mittelwert der verfügbaren Kapazität über dem Mittelwert der nachgefragten Kapazität, kann ein Engpass dennoch nicht ausgeschlossen werden (vgl. [CB02, S. 64]). Soll weiterhin z.B. der Status eines Auftrages über die Zeit verfolgt werden (Anforderung 14), ist die Verwendung eines statischen Verfahrens ausgeschlossen. Es werden also **dynamische** Modelle bevorzugt.

Kontinuierliche Modelle werden insbesondere dort eingesetzt, wo das Systemverhalten durch Differentialgleichungen angemessen repräsentiert werden kann (vgl. [WBN⁺02, S. 4], [LK91, S. 109]). Typische Einsatzgebiete sind die Abbildung biologischer oder chemischer Prozesse. Die Systeme werden durch zeitabhängige Zustandsvariablen abgebildet. Durch Bildung zeitlicher Ableitungen können Aussagen über die Zustandsgrößen gemacht werden. Die Simulation besteht im Lösen der Differentialgleichungen für vorgegebene Eingangsvariablen [Pag91, S. 9].

In *diskreten Modellen* finden Zustandsänderungen⁷ im Gegensatz zur kontinuierlichen Simulation ausschließlich zu bestimmten Zeitpunkten statt. Zwischen diesen

⁵In der VDI-Richtlinie wird ein *symbolisches Modell* aus einem *gedanklichen Modell* abgeleitet und sei nicht experimentierfähig. Auch wenn symbolische Modelle im Sinne der Computeralgebra aufgefasst werden (vgl. [ea95, S. 111]), lassen sich diese Modelle textuell repräsentieren.

⁶Die Verwendung »echter« Zufallszahlen in stochastischen Verteilungen bringen einen wesentlichen Nachteil mit sich: Die Ergebnisse sind nicht reproduzierbar. Daher finden sog. *Pseudozufallszahlen* Verwendung. Pseudozufallsgeneratoren produzieren reproduzierbare Zahlenfolgen, wenn sie mit identischen Parametern aufgerufen werden. Kriterien für die Güte eines solchen Generators sind die Unkorreliertheit beliebiger Teilfolgen, möglichst große Periodizität und die Gleichverteilung der erzeugten Zahlen [Knu81, S.2].

⁷Der *Systemzustand* ist die Menge der Zustandsvariablen, die notwendig ist, um das System zu einem bestimmten Zeitpunkt zu beschreiben [LK91, S. 10].

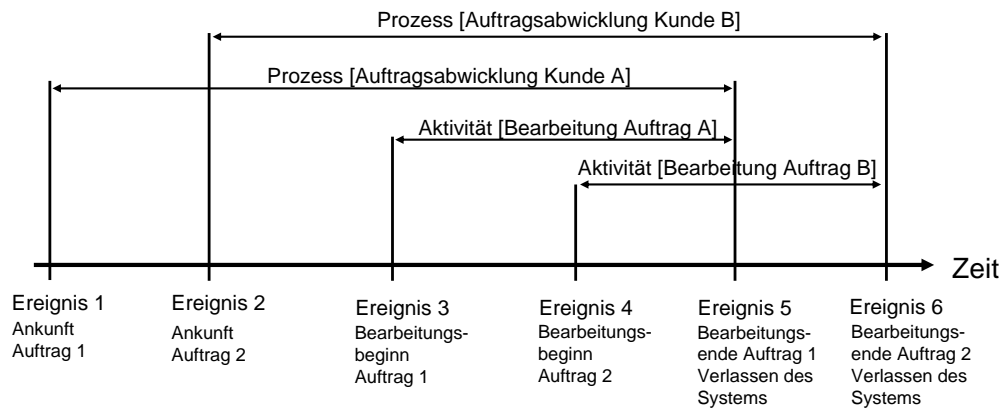


Abbildung 5.1.: Events, Aktivitäten, Prozesse (Quelle: [Pag91, S. 27])

Zeitpunkten bleibt der Systemzustand unverändert. Anders formuliert: Der Systemzustand ändert sich an einer abzählbaren Menge von Zeitpunkten [LK91, S. 7].

Teilprozesse innerhalb des Auftragsabwicklungsprozesses werden häufig durch Ereignisse beeinflusst, die zu bestimmten Zeitpunkten stattfinden, während kontinuierliche Vorgänge weniger offensichtlich eine Rolle spielen⁸. Die große Mehrzahl der gut durch an diskreten Zeitpunkten stattfindende Zustandsänderungen beschreibbaren Teilsysteme führt zu der Entscheidung, **diskrete** Zustandsänderungen als Grundlage für den Modellierungsansatz vorzusehen. Diese Entscheidung lässt den Weg zur Modellierung von Teilsystemen mit kontinuierlichen Zustandsänderungen offen, da sich die beiden Ansätze kombinieren lassen (siehe [WBN⁺03a], [WBN⁺03b], [WBN⁺04], [Kof03b], [ZPK00, S. 203]).

Bei diskreten Modellen lassen sich drei Kategorien unterscheiden (vgl. [EHP94, S. 12], [Pag91, S. 26-27]):

Ereignisorientierte Sicht. Die ereignisorientierte Sicht fokussiert auf die Vorgänge, die zu den jeweiligen Ereigniszeitpunkten durchzuführen sind und den Systemzustand verändern. Die »Ausführung«⁹ eines Ereignisses verbraucht keine Zeit, erzeugt aber möglicherweise weitere Ereignisse, die zum gleichen Moment oder später ausgeführt werden. Das jeweils nächste Ereignis wird einer Warteschlange entnommen, die nach dem Ereigniszeitpunkt sortiert ist.

Aktivitätsorientierte Sicht. Als »Aktivität« wird in diesem Zusammenhang ein zeitliches Intervall verstanden, in dem eine Aufgabe durchgeführt wird. In ereignisorientierter Sichtweise kann eine »Aktivität X« durch zwei Ereignisse abgebildet werden: »Start Aktivität X« und »Ende Aktivität X«.

⁸Beispiele für Teilsysteme, die sich für eine kontinuierliche Beschreibung eignen, sind das Kunden- bzw. Marktverhalten innerhalb eines Modells.

⁹Die mit einem Ereignis verbundenen Aktionen (Anweisungen zur Veränderung des Systemzustands) werden ausgeführt.

Prozessorientierte Sicht. Ein Prozess kann als die Menge aller Aktivitäten aufgefasst werden, die in einem Teilsystem durchgeführt werden. Diese Aktivitäten werden über den gesamten *Lebenszyklus* des Prozesses beschrieben.

Der ereignisorientierte Beschreibungsansatz eignet sich also als Grundlage sowohl für eine aktivitätsorientierte Modellierung als auch für eine prozessorientierte Modellierung. THIELE führt Geschäftsprozesse als Beispiel für Systeme auf, die sich als »Discrete Event Systems« (DES) beschreiben lassen¹⁰ (vgl. [Thi03, S. 2-3]). SCHEER charakterisiert die Auftragsabwicklung als »betont ereignisorientiert« [Sch90, S. 366]. Als Basis für einen Modellierungsansatz soll daher ein **ereignisorientierter** Ansatz dienen, der aber auch eine prozessorientierte Sicht auf ein Modell bieten sollte, um zeitverbrauchende Vorgänge natürlich beschreiben zu können.

Zusammengefasst sollen also Modelle experimentierbar, textuell, digital, Software, stochastisch, diskret und ereignisgesteuert sein.

5.2. Existierende Modellierungssprachen

Welche Modellierungssprachen bzw. Bewertungswerkzeuge können nun potenziell genutzt werden, um Auftragsabwicklungsprozesse zu modellieren bzw. Modelle von Auftragsabwicklungsprozessen zu bewerten? Es existieren hunderte von Notationen, um Geschäftsprozesse zu modellieren (vgl. [HvR00]), wobei diese Notationen für unterschiedlichste Zielsetzungen entwickelt wurden. Beispiele für solche Modellierungssprachen sind¹¹ (vgl. [Pet02, S. 16]):

1. **Dortmunder Prozesskettennotation.** Methode und Notation zur Unternehmensbeschreibung (vgl. [Kuh95], [KLS02]). Die ProC/B-Prozesskettennotation formalisiert eine Teilmenge der Dortmunder Prozesskettennotation, macht sie einer maschinengestützten Analyse zugänglich, unterstützt eine objektorientierte, prozessorientierte und hierarchische Modellierung und erlaubt die Bewertung von logistischen Netzwerken in vielfacher Hinsicht (vgl. [BBT⁺99, S. 4], [Fis03, S. 8], [iHSV03], [DV03]).
2. **UML.** Die *Unified Modeling Language* wurde in erster Linie zum Einsatz in der Softwareentwicklung entwickelt, findet aber auch Einsatz in der Unternehmens- bzw. Geschäftsprozessmodellierung (vgl. [Pet02, S. 16], vgl. auch [EP00], [JEJ95]). Besondere »Profile« (z.B. das EDOC Profil [Enterprise Distributed Object Computing] der Object Management Group [<http://www.omg.org>]) unterstützen die Erstellung von Unternehmensmodellen.
3. **UEML.** Die *Unified Enterprise Modeling Language* wurde mit der Zielsetzung entwickelt, Wissen über ein Unternehmen in einer vereinheitlichten und erweiterbaren Weise in einem Modell zu repräsentieren. UEML befindet sich noch im Entwicklungsstadium und wird noch nicht verbreitet eingesetzt (vgl. [Die04, S. 129]).

¹⁰Für eine formale Definition von DES siehe [Thi03, S. 18].

¹¹Die Auswahl wurde u.a. auf Basis des Verbreitungsgrades getroffen.

4. **BPMN.** Eine weitere Modellierungssprache, deren Entwicklung durch die »Object Management Group« koordiniert wird, ist die *Business Process Modeling Notation* (BPMN). Ein wesentliches Ziel bei der Entwicklung von BPMN war die mögliche »Ausführbarkeit« der Modelle (vgl. [Whi04a, S. 5]). Hierzu können aus BPMN-Modellen BPEL4WS-Modelle (Business Process Execution Language for Web Services) erzeugt werden. Die OMG entwickelt derzeit ein »UML Business Process Definition Meta-Modell« (siehe Abbildung 5.3), in welchem BPMN eine Diagramm-Sicht darstellt (vgl. [Whi04a, S. 11]). BPMN wird von einem Industriekonsortium unterstützt, dem u.a. IBM und Microsoft angehören.
5. **ARIS.** Die »Architektur integrierter Informationssysteme« basiert auf einem Konzept, welches unterschiedliche Sichten (*Daten, Steuerung, Funktionen* und übergeordnet *Organisation* [Sch90, S. 7]) auf ein Unternehmen miteinander vereinigt. Zur Abbildung von Geschäftsprozessen können Ereignisgesteuerte Prozessketten (EPK) genutzt werden (vgl. [Sch01, S. 125]). Methoden der objektorientierten Modellierung können mit EKP genutzt werden (vgl. [NZ98a], [NZ98b]).
6. **IDEF.** Die »Integrated Definition Methods« bilden eine Sammlung von Notationen¹², die jeweils verschiedene Schwerpunkte der Beschreibung setzen. PETIT beschreibt Entsprechungen für IDEF Notationen in der Unified Modeling Language (vgl. [Pet02, S. 98-99]):
 - IDEF0 ~ UML Activity Diagramme.
 - IDEF1 und IDEF1x ~ UML Class und Object Diagramme.
 - IDEF2 ~ UML Collaboration, Activity und State Diagramme.
 - IDEF3 ~ UML Activity, State und Use Case Diagramme.
 - IDEF4 ~ UML Class, Object, Activity und State Diagramme.
7. **PSL.** Die *Process Specification Language* wurde unter anderem mit dem Ziel entwickelt, als Austauschformat zwischen anderen Prozessbeschreibungen dienen zu können ([SGT⁺00, S. 4]). Besonderer Wert bei der Gestaltung der PSL wurde auf die Definition einer Ontologie gelegt (vgl. [SGT⁺00, S. 8]).

Das hier beispielhaft erläuterte Dortmunder Prozesskettenparadigma beschreibt Unternehmen anhand ihrer Prozesse und benennt als Parameter eines Prozesskettenelementes (vgl. [Kuh95, S. 43-46]):

- ▷ **Quellen und Senken.** Input und Output des Systems.
- ▷ **Prozesse.** Was geschieht im System?
- ▷ **Ressourcen.** Welche Ressourcen werden von den Prozessen benötigt? Diese Ressourcen können als Betriebsmittel eines Unternehmens gesehen werden. Nach

¹²IDEF0 »Function Modeling Method«, IDEF1 »Information Modeling Method«, IDEF1X »Data Modeling Method«, IDEF2 »Simulation Model Design«, IDEF3 »Process Description Capture Method«, IDEF4 »Object Oriented Design Method«, IDEF5 »Ontology Description Capture Method« (vgl. <http://www.ideal.com>, [KMN⁺01, S. 51], [Pet02, S. 96]), für weitere IDEF Notationen (IDEF 6 - IDEF 14) siehe [Díe04, S. 110] oder [Pet02, S. 96].

KUHN werden diese auch als die sechs »knappen Betriebsmittel der Logistik« bezeichnet:

1. Personal
2. Flächen
3. Bestände
4. Bearbeitungsmittel
5. Arbeitshilfsmittel
6. Organisationsmittel

- ▷ **Strukturen.** Informationen über Layout, Aufbauorganisation und technische Kommunikationsstruktur.
- ▷ **Lenkungsebenen.** Regeln, Steuerungsvorschriften, Strategien, nach denen die Abläufe innerhalb der Prozessketten bestimmt werden. Die fünf Ebenen der Lenkung sind nach KUHN (vgl. [Kuh95, S. 44]):

1. Normative Gestaltungsebene
2. Administrative Gestaltungsebene
3. Dispositive Gestaltungsebene
4. Netzwerk
5. System/Ressourcen/Kapazität.

WINZ und QUINT definieren¹³ (vgl. [WQ97, S. 17]):

Unter einem Prozesskettenelement wird eine geordnete Abfolge von Aktivitäten verstanden, die einen definierten Input (Leistungsobjekte vom Lieferanten) in einen definierten Output (transformierte Leistungsobjekte an den Kunden) überführen. Die Prozesskette besteht aus den einzelnen Prozesskettenelementen, die entlang der Zeitachse miteinander verknüpft sind.

Geschäftsprozessmodellierungssprachen ermöglichen die Notation von Auftragsabwicklungsprozessen und in Teilen auch deren Bewertung; so können beispielsweise Modelle in der ProC/B-Prozesskettennotation sowohl analytisch (vgl. [FD03, S. 11]) als auch mittels Simulation (vgl. [BBT⁺99, S. 4]) bewertet werden. Es existieren also bereits Modellierungs- und Bewertungsansätze, die grundsätzlich eine Abbildung von Auftragsabwicklungsprozessen erlauben und darüber hinaus die Bewertung, z.B. durch die Berechnung geeigneter Kennzahlen, unterstützen (siehe Kapitel 4.1).

Neben dem oben bereits genannten ProC/B-Paradigma existieren z.B. auch für Ereignisgesteuerte Prozessketten (EPK) und die IDEF Notation¹⁴ die Möglichkeit, detaillierte Simulationsmodelle mit einem Prozessmodell in der jeweiligen Notation zu

¹³Orthographie geändert.

¹⁴Die für den Entwurf von Simulationsmodellen geschaffene Notation IDEF2 wird allerdings in keinem nennenswerten Umfang mehr genutzt (vgl. [IEE98, S. 3]). LEONG setzt stattdessen IDEF0, IDEF1X und IDEF3 für diesen Zweck ein (vgl. [Leo00, S. 5]).

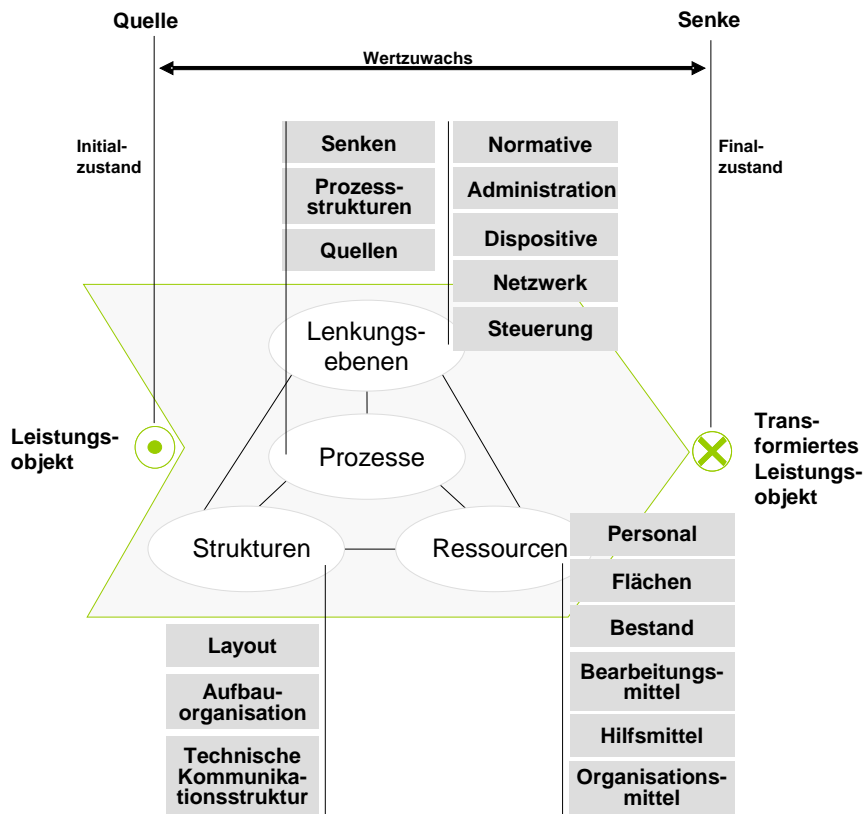


Abbildung 5.2.: Prozesskettenelement des Dortmunder Prozesskettenparadigmas (angelehnt an: [Kuh95, S. 47] und [WQ97, S. 18])

verbinden bzw. Informationen in die Prozessmodelle zu integrieren, so dass eine Bewertung mittels Simulation möglich ist (vgl. [DV03], [CK04, S. 17], [Leo00]).

Einige der oben genannten Modellierungssprachen sind bereits internationale Standards (PSL) oder befinden sich im Prozess der Standardisierung (BPMN, UML Profile for Business Process Definition (vgl. [CIM05, S. 1-2]). Die von der OMG koordinierte Entwicklung der Sprachen UML und BPMN hebt sich aber in vielerlei Hinsichten von den konkurrierenden Ansätzen ab.

Unter den Methoden zur Unterstützung der Analyse- und Designphase in der Softwareentwicklung wird seit einigen Jahren UML am häufigsten mit einem Anteil von 70 % in der objektorientierten Softwareentwicklung eingesetzt¹⁵ (vgl. [Pen03], siehe auch [Gus02, S. 169]). Die *Unified Modeling Language* ist aus der Verschmelzung der Notationen *OMT* von RUMBAUGH und *Booch* von BOOCH [RJB99] unter Mitwirkung von JACOBSON entstanden und wird neben der Softwareentwicklung z.B. auch im Bereich der Unternehmensmodellierung (Enterprise Modeling) eingesetzt (vgl. [Pan03, S. 382], vgl. auch [RC03, S. 1613], [JEJ95]).

¹⁵Eine Definition erfolgt in Abschnitt 5.3.

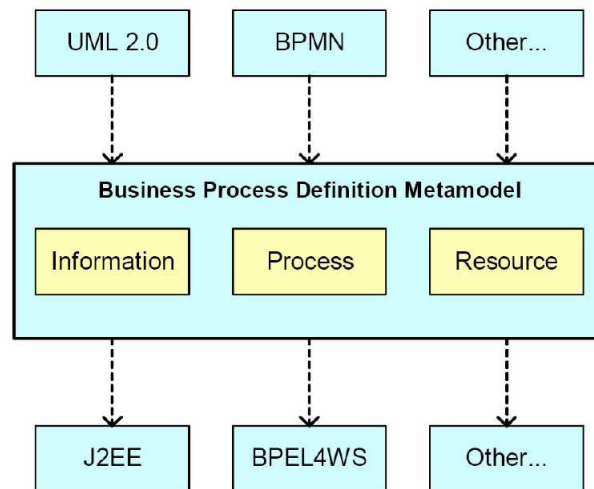


Abbildung 5.3.: Business Process Definition Meta-Modell der Object Management Group (Quelle: [OMG04, S. 2])

UML ist zwar in erster Linie für die Unterstützung der Softwareentwicklung konzipiert, erlaubt aber auch die Modellierung von Unternehmensprozessen. Die von der OMG geplante Integration von UML und BPMN in einem »Business Process Definition Meta-Modell« erlaubt ferner die Erweiterung von UML-Modellen durch in der BPMN formulierte weitere Sichten. Sowohl UML als auch BPMN erlauben dabei die Spezifikation von ausführbaren Modellen¹⁶.

Die Werkzeuge, die eine Modellierung unter Verwendung der oben genannten Modellierungssprachen unterstützen – »Geschäftsprozessmodellierungstools« – bieten aber nur eine unzureichende Unterstützung der detaillierten Abbildung von Prozessen im vorliegenden Einsatzgebiet (vgl. [KMN⁺01, S. 63-64], [HR98, 1364]). Die geringe Eignung der oben genannten Modellierungssprachen zur direkten Abbildung von Auftragsabwicklungsprozessen kann bereits durch das Fehlen einer für den vorliegenden Fall sehr wesentlichen Eigenschaft aufgezeigt werden: Einem Konzept zur effizienten Beschreibung der Produktstruktur (siehe Kapitel 4.5). Die hiervon »betroffenen« Anforderungen sind: 1, 5, 7, 8, 9, 10.

Dieses Fehlen spezifischer Elemente für die Abbildung von Auftragsabwicklungsprozessen bei »allgemeinen« Modellierungssprachen ist ebenso nahe liegend, wie hinderlich. Um der Forderung nach einer adäquaten Sprache nachzukommen (GOM: SPRACHADÄQUANZ) und somit eine wirtschaftliche Modellierung zu ermöglichen oder zumindest zu erleichtern (GOM: WIRTSCHAFTLICHKEIT), muss also entweder eine neue Modellierungssprache geschaffen werden, welche die benötigten spezifischen Elemente enthält, oder aber eine bestehende Modellierungssprache erweitert werden.

Neben der Eignung einer Sprache zur Abbildung von Auftragsabwicklungsprozessen, spielt die Möglichkeit zur *effizienten* Bewertung eine wichtige Rolle. Das heißt,

¹⁶UML ab der Version 2.0.

es ist nicht ausreichend, dass die Modelle in einem theoretischen Sinne (»in endlicher Zeit«) bewertet werden können, sondern dass die zur Bewertung benötigte Zeit eine praktische Nutzung des Ansatzes erlaubt (GoM: Wirtschaftlichkeit).

Eine Modellierungssprache kann daher nicht sinnvoll ausgewählt werden, ohne sicherzustellen, dass in dieser Sprache formulierte Modelle auch im obigen Sinne angemessen bewertet werden können. Die Auswahl einer »Plattform« für die Bewertung von Modellen von Auftragsabwicklungsprozessen führt zu konkretisierten Anforderungen, die die Auswahl einer Modellierungssprache erlauben.

5.3. Festlegung der Softwareplattform

Im folgenden Kapitel soll eine geeignete Softwarebasis ausgewählt werden, die sowohl die Modellierung als auch die Bewertung von Diskrete-Event-Modellen von Auftragsabwicklungsprozessen hinsichtlich der gewünschten Zielgrößen erlaubt. Die hier gesuchte Software soll eine Basis bieten¹⁷, um Modelle, die diesen Anforderungen genügen, mittels Simulation zu bewerten.

Die Vorgehensweise für den Suchprozess wurde an VDI 3633 Blatt 4 (vgl. [VDI97b], siehe auch [BBH⁺01, S. 2], [TVVB02, S. 68], [JJE02, S. 307])¹⁸, angelehnt.

1. **Problemdefinition.** Suche nach einer geeigneten Basis¹⁹ für die Implementierung von Software zur Bewertung von Modellen von Auftragsabwicklungsprozessen²⁰.
2. **Zwingende Anforderungen.**
 - ▷ [GOM: WIRTSCHAFTLICHKEIT, KLARHEIT] Direkte Anwendbarkeit objektorientierter Analyse- und Designmethoden (OOA/OOD, Definition unten). Die Realisierung eines Werkzeugs für die Gestaltung und Bewertung von Auftragsabwicklungsprozessen ist u.a. auch ein Softwareprojekt. Es ist nahe liegend, für ein solches Projekt auf bewährte Technologien zurückzugreifen, wie die Entwurfstechniken OOA/OOD und die objektorientierte Programmierung. Darüber hinaus lassen sich die Vorteile dieser Ansätze auch für die Modellierung von Auftragsabwicklungsprozessen nutzbar machen (siehe Kapitel 6).
 - ▷ [GOM: WIRTSCHAFTLICHKEIT] Die Bewertung auch sehr umfangreicher

¹⁷Im Folgenden auch *Basisframework* genannt.

¹⁸Als Methode zur Entscheidungsfindung in der Softwareauswahl kann z.B. die *Multi Attribute Utility Theory* (MAUT) eingesetzt werden (vgl. [BBH⁺01, S. 8]), die es erlaubt, unterschiedliche Kriterien auf einer Skala (z.B. 0 bis 1) quantitativ zu bewerten und zu einem Gesamtergebnis zu aggregieren. Eine Variante der MAUT ist die *Simple Multi Attribute Rating Technique* (SMART), die eine Gleichbewertung von Alternativen zulässt, wenn die Bewertungsgrundlage keine Vergabe unterschiedlicher Bewertungen zulässt (vgl. [BBH⁺01, S. 8]).

¹⁹Zwischen dieser »Basis« und der konkreten Ausgestaltung des im Rahmen dieser Arbeit entwickelten Ansatzes existieren zahlreiche Wechselwirkungen. Unterstützt z.B. die verwendete Basis (Framework, Entwicklungsumgebung) nicht direkt die Verwendung objektorientierter Programmierung, ist die Nutzung von OO-Techniken innerhalb des Ansatzes weitaus schwieriger umzusetzen.

²⁰Im Folgenden auch *Bewertungswerkzeug* oder *Bewertungstool* genannt.

Modelle muss in »angemessener« Zeit möglich sein²¹. Ein zentrales Ziel des hier vorgestellten Ansatzes ist es, die Bewertung des Entwurfes eines Auftragsabwicklungsprozesses nicht nur innerhalb »endlicher Zeit« zu ermöglichen, sondern innerhalb eines Zeitraumes, der so kurz ist, dass auch die Untersuchung zahlreicher Gestaltungsalternativen möglich wird.

Dieses Ziel soll als erreicht gelten, wenn die Bewertung eines Modells so möglich ist, dass eine Berechnung »über Nacht« am nächsten Arbeitstag Ergebnisse bringt (<12 Stunden). Diese Obergrenze soll für höchst umfangreiche Modelle gelten, in denen sehr viele Baureihen eines Herstellers in zahlreichen Standorten gefertigt werden, bei einem viele Jahre umfassenden Bewertungszeitraum (siehe Kapitel 3.1.2).

- ▷ [GOM: WIRTSCHAFTLICHKEIT] Die Bewertung auch sehr umfangreicher Modelle muss hauptspeicherbasiert erfolgen können. Die für die Berechnung erforderlichen Daten müssen vollständig im Hauptspeicher des verwendeten Computers gehalten werden können, um sehr zeitintensive Zugriffe auf sekundäre Speichermedien (z.B. Festplatten) zu vermeiden (vgl. [Zei76, S.3]).
- ▷ [GOM: WIRTSCHAFTLICHKEIT] Verfügbarkeit leistungsfähiger Entwicklungswerkzeuge.
 - Compiler/Interpreter. Die Übersetzung des Source Codes des Bewertungswerkzeugs in ein ausführbares Programm bzw. die Interpretation des Source-Codes während der Ausführung muss in einer Weise erfolgen, dass auch große Datenmengen (mehrere Gigabyte im Hauptspeicher) sicher beherrscht werden können.
 - Debugger. Der *interne Zustand* des Bewertungswerkzeugs muss während der Berechnung zugänglich sein, d.h. es muss möglich sein, die Berechnung anzuhalten, um interne Variablen des Werkzeugs zu untersuchen. Das Vorhandensein eines solchen Werkzeugs ist erforderlich, um Fehler in der Programmierung schnell aufspüren zu können.
 - Profiler. Durch statische Betrachtung der Implementierung eines Programms ist es außerordentlich schwer abzuschätzen, wie viel Zeit tatsächlich für die Abarbeitung von bestimmten Code-Stücken erforderlich ist. Dies gilt insbesondere, wenn Bibliotheken²² genutzt werden, deren Quellcode nicht zugänglich ist. Ein *Profiler* ist ein Werkzeug, welches die Zeit misst, die tatsächlich für die Abarbeitung einer Methode bzw. Funktion verstrichen ist, bzw. welchen Anteil diese Berechnung an der Gesamtrechnenzeit einnimmt.
- ▷ [GOM: WIRTSCHAFTLICHKEIT] Geringes Risiko von Problemen durch Implementierungsbeschränkungen. In vielen Fällen sind Softwareentwickler gezwungen, Entscheidungen zwischen Effizienz in der Berechnung und einer Beschränkung der max. Systemgröße zu treffen, die ein Simulationsmodell haben kann. Eine Datenstruktur beschränkter Größe kann z.B. im Hauptspei-

²¹Anforderungen 21, 22.

²²Diese Bibliotheken können z.B. Funktionen für die Verwaltung von Objekten in Listen oder Vektoren zur Verfügung stellen.

cher bearbeitet werden. Für größere Datenmengen ist eine Bearbeitung auf einem Massenspeichermedium erforderlich (z.B. Festplatte). Der Geschwindigkeitsunterschied zwischen diesen Varianten kann mehrere Dimensionen umfassen.

3. Wünschenswerte Eigenschaften.

- ▷ [GOM: WIRTSCHAFTLICHKEIT] Vorhandene Funktionalitäten für den Anwendungsbereich. NIKOUKARAN et al unterscheiden drei Kategorien von Simulationssoftware (vgl. [NP98, S. 405]): Allgemeine Programmiersprachen, Simulationssprachen und Simulatoren²³. WENZEL und NOCHE unterteilen die Kategorie der Simulatoren nochmals in fünf Unterkategorien und ordnen den Kategorien unterschiedliche Grade des Anwendungsbezuges bzw. der Allgemeingültigkeit zu (siehe Abbildung 5.4).

Eine Basis-Software, die möglichst viele der benötigten Funktionalitäten bereits bietet (z.B. leistungsfähige Produktmodellierung oder Auswertungsmöglichkeiten), erlaubt eine schnellere Umsetzung des gewünschten Werkzeugs.

- ▷ [GOM: WIRTSCHAFTLICHKEIT] Möglichkeit zur multidimensionalen Datenanalyse. Die Verfügbarkeit von OLAP-basierten Auswertungen (vgl. [RG01, S. 682]) erleichtert die Formulierung auch sehr komplexer Abfragen und senkt somit den Aufwand für die Bewertung von Simulationsmodellen.
- ▷ [GOM: WIRTSCHAFTLICHKEIT] Detaillierte Dokumentation. Die großen zu bewältigenden Datenmengen erfordern einen Einblick in die Realisierung der Funktionalitäten des Basiswerkzeugs, die zur Umsetzung des Bewertungstools benötigt werden, um das Verhalten bei großen Datenmengen zuverlässig beurteilen zu können.
- ▷ [GOM: KLARHEIT, WIRTSCHAFTLICHKEIT] Einfache Bedienung. Der Modellaufbau sollte in intuitiver Weise möglich sein, z.B. in graphischer Form.
- ▷ [GOM: WIRTSCHAFTLICHKEIT] Große Verbreitung. Eine große Anzahl von Anwendern führt zu einer entsprechenden Anzahl Rückmeldungen über etwaige Probleme an den Hersteller einer Software und bildet einen interessanten Markt für Verlage, so dass im Falle großer Verbreitung von einem reichhaltigen Literaturangebot ausgegangen werden kann.

4. **Lösungsalternativen.** Als Kandidaten für einen Einsatz als Basisframework kommt zunächst Software aus allen Kategorien aus Abbildung 5.4 in Frage. Spezialsimulatoren für den Bereich Auftragsabwicklungsprozesse sind bislang nicht verfügbar²⁴, daher kommen für die weitere Auswahl »Allgemeine Programmier-

²³In der VDI-Richtlinie 3633 [VDI96, S.15-16] wird »Simulator« definiert als:

Softwareprogramm, mit dem ein Modell zur Nachbildung des dynamischen Verhaltens eines Systems und seiner Prozesse ausführbar gemacht werden kann.

²⁴In zwei Artikeln aus dem Jahr 2002 berichten KUMARA und YEE (vgl. [KLT⁺02] und [Yee02]) u.a.

sprachen«, »Simulationssprachen«, »Simulatorentwicklungsumgebungen« und »Simulationsumgebungen« in Betracht (siehe oben).

Das Attribut »objektorientiert« wird unüberschaubar vielen Konzepten angeheftet. Daher ist es erforderlich zu definieren, was gemeint ist, wenn auch hier von einem objektorientierten Ansatz gesprochen wird. MEYER definiert²⁵ [Mey97, S. VI]:

Object technology is at its core the combination of four ideas: a structuring method, a reliability discipline, an epistemological principle and a classification technique.

1. **Structuring Method.** Strukturierung auf der Basis von Objekten anstelle von Aktionen.
2. **Reliability Discipline.** Auffassung eines Systems als eine Ansammlung von Komponenten, die auf genau definierte Weise interagieren (*by Construct*).
3. **Epistemological Principle.** Wie werden die Klassen eines Systems beschrieben? Die Objekte einer Klasse werden dadurch beschrieben, was mit diesen Objekten getan werden kann. Die Anwendung von Operationen dieser Klasse (*Features*) zusammen mit den formalen Eigenschaften dieser Operationen (*Contracts*).
4. **Classification Technique.** Ableitung von Taxonomien über den Anwendungsbereich, der Gegenstand der Untersuchung ist. Ein wichtiges Instrument hierbei ist die *Vererbung*.

Objektorientierte Programmierung ist eine Methode der Softwareentwicklung, die den gesamten »Lebenszyklus« von Software umfasst, von Analyse und Design, über die Implementierung bis hin zur Wartung. Das zentrale Konzept der Methode ist die *Klasse*, ein Softwareelement, welches einen abstrakten Datentyp und seine teilweise oder vollständige Implementierung beschreibt.

Ein *abstrakter Datentyp* ist eine Menge von Objekten, zusammen mit einer Reihe von Operationen, die darauf anwendbar sind sowie die Eigenschaften dieser Operationen (vgl. [Mey97, S.22-23]). STROUSTRUP sieht als Bedingungen für »Objektorientiertheit« [Str95, S. 2]:

A language or technique is objectoriented if and only if it directly supports:

1. *Abstraction – providing some form of classes and objects.*
2. *Inheritance – providing the ability to build new abstractions out of existing ones.*
3. *Runtime polymorphism – providing some form of runtime binding.*

Eines der Kernkonzepte objektorientierter Programmierung ist die *Vererbung* [Mey97, S. 459]. Hiermit wird die Möglichkeit beschrieben, durch *Ableitung* von ei-

über eine interne General Motors Entwicklung einer internetfähigen Benutzeroberfläche für einen Simulator, welcher zur Abbildung von Auftragsabwicklungsprozessen geeignet sein soll. Details über die Funktionsweise des Simulators lassen sich den Publikationen aber nicht entnehmen.

²⁵Im Rahmen dieser Arbeit werden nicht alle unten aufgeführten Elemente genutzt. Die unter dem Punkt *Reliability Discipline* genannten formal spezifizierten *Contracts* werden nicht explizit formuliert. Diese werden von STROUSTRUP allerdings auch nicht als Bedingung für »Objektorientiertheit« gesehen (vgl. [Str95, S. 2]).

ner Klasse sowohl die Attribute als auch die Methoden²⁶ zu erben, d.h. nutzen zu können²⁷.

Eine nahe liegende Kategorie von Simulationstools aus Abbildung 5.4 für die Modellierung und Bewertung von Auftragsabwicklungsprozessen sind die »Simulationsumgebungen«. Diese Werkzeuge vereinen häufig eine graphische Modellierungskomponente, eine Simulationssprache, eine Komponente, die die erzeugten Modelle mit Möglichkeiten zur Analyse der Ergebnisse bewertet.

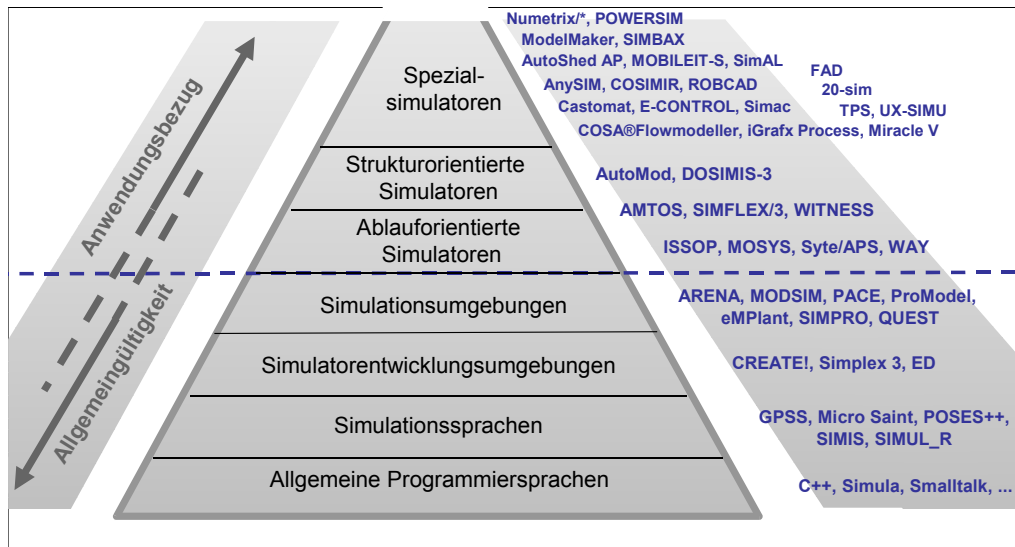


Abbildung 5.4.: Klassifizierung von Simulationstools (Quelle: [WN00, S. 426])

In einer Softwareauswahl für das Einsatzgebiet »Supply Chain Modeling«²⁸ haben KIM et al aus zahlreichen Simulationsumgebungen und Simulatorentwicklungsumgebungen²⁹ diejenigen Softwarepakete aufgeführt, die den formulierten Anforderungen am besten entsprechen (vgl. [KTB⁺04, S. 44-46]). Arena des Herstellers »Rockwell Automation« erhielt die beste Bewertung. Auf den »Plätzen« folgen Quest, Witness, Extend und ProModel, FlexSim, AutoMod, Enterprise Dynamics und Anylogic.

Die zwingenden Anforderungen waren (vgl. [KTB⁺04, S. 43]):

- ▷ *Vendor: includes vendor pedigree, maintenance support, and documentation*
- ▷ *Model development and input: includes the model building and coding aspect, batch processing, library of reusable modules, conditional routing, statistical*

²⁶In obiger Definition MEYERS *Features*.

²⁷Für eine ausführliche Beschreibung der Konzepte der objektorientierten Programmierung siehe z.B. [Mey97]; für eine Umsetzung dieser Konzepte in eine konkrete Programmiersprache (C++) siehe [Str97].

²⁸Zulieferernetze können auch Bestandteil von Modellen von Auftragsabwicklungsprozessen sein. Eine Überschneidung in der Aufgabenstellung ist hier also gegeben.

²⁹Untersucht wurden: Analytica, AnyLogic, Arena, AutoMod, Crystal Ball, DESMO, DecisionPro, eM-Plant, Enterprise Dynamics, Extend, Factory Explorer, Goldsim, Flexsim, GPSS/H, High-MAST, MAST, PCModel, MicroSaint, PIMSS, ProcessModel, ProModel, Quest, Resource Manager, ShowFlow, SIGMA, SimCAD Pro, SIMUL8, Slam, SLIM, SLX, Supply Chain Builder, TOMAS, VisSim, Visual Simulation Environment, Witness.

distributions, queue policy, input modes, automatic documentation, batch input modes, random number generator, standard commands, model packaging

- ▷ *Execution: includes multiple run, automatic batch run, reset capability, start in nonempty state, interaction with user and unit conversions*
- ▷ *Simulation and optimization engines*
- ▷ *Animation: includes icon and animation development, screen layout and animation running*
- ▷ *Testing and efficiency: includes validation and verification tools, display features, tracing, step functions, breakpoints, model size and model speed*
- ▷ *Output: includes report, integration with external packages and business graphics*
- ▷ *User: required experience and cost of package*
- ▷ *Experimental design use of tools that can help to model correctly.*

Den Vorteilen, die eine Verwendung bereits am Markt existierender Simulationssoftware mit sich bringt – insbesondere müssen bereits vorhandene Funktionen nicht mehr implementiert werden – stehen gewichtige Nachteile entgegen:

1. Es fehlen u.a. ausreichende Möglichkeiten zur Produktbeschreibung (Anforderungen 1, 5, 7, 8, 9, 10).
2. Die Auswahl an Entwicklungswerkzeugen ist im Vergleich zu allgemeinen Programmiersprachen sehr eingeschränkt.
3. Index und Speicherstrukturen zur effizienten Datenverwaltung (Hash-Tables, B-Trees, etc.) sind regelmässig nicht vorhanden.
4. Aussagen zur Bewertbarkeit sehr großer Simulationsmodelle sind kaum möglich.

Es verbleibt die Kategorie »Allgemeine Programmiersprachen« für die weitere Auswahl, aus folgenden Gründen:

- ▷ Die Umsetzbarkeit von, mittels OOA/OOD Methoden (vgl. [Boo94, S. 35-40]) entwickelten, Software Designs ist bei Programmiersprachen wie C++ oder Java sehr gut gegeben.
- ▷ Für allgemeine Programmiersprachen stehen hocheffiziente Entwicklungswerkzeuge zur Verfügung, die z.B. im Falle von Intel für ihre Prozessoren »maßgeschneidert« werden. Spezielle Funktionseinheiten von Prozessoren (z.B. SSE – Streaming SIMD Extensions – von Intel, vgl. [Pat05]) können direkt angesprochen werden³⁰.
- ▷ Die sehr gute Kontrolle über den Speicherverbrauch³¹ und die Möglichkeit, z.B. durch Bit-Vektoren, Platz sparend und effizient Informationen ablegen zu können. Diese Eigenschaft ist für die Möglichkeit auch große Modelle hauptspeicherbasiert bewerten zu können, von hoher Bedeutung.

³⁰Von dieser Möglichkeit wurde Gebrauch gemacht, um einige sehr häufig genutzte Operationen zu beschleunigen.

³¹Dies gilt nur eingeschränkt für Programmiersprachen, die einen sog. »Garbage Collector« einsetzen, um nicht mehr benötigte Speicherblöcke zu erkennen und freizugeben, wie z.B. Java und C#.

- ▷ Allgemeine Programmiersprachen werden genutzt, um auch größte Softwarepakete zu erstellen. Das Risiko, auf eine unüberwindbare Größenbeschränkung zu stoßen, ist minimal. Die vergleichsweise große Plattformunabhängigkeit erlaubt es, durch Ausweichen auf eine andere Rechnerplattform etwaige, plattformspezifische Engpässe zu umgehen.
- ▷ Für allgemeine Programmiersprachen stehen zahlreiche Werkzeuge zur Verfügung, um z.B. Interpreter für einfache Programmiersprachen umzusetzen (z.B. Lex und Yacc).

Die objektorientierten Programmiersprachen mit der größten Verbreitung sind zweifellos Java und C++. Von Microsoft wurden Elemente aus Java und C++ zu einer neuen Sprache C# zusammengeführt³². Alle drei Sprachen eignen sich sehr gut, um die Software zur Bewertung von Auftragsabwicklungsprozessen darin umzusetzen. Die bessere Kontrolle über den Speicherverbrauch spricht allerdings für C++ und gibt den Ausschlag für die Verwendung von C++ für die Implementierung diesbezüglich kritischer Teile des Bewertungswerkzeugs.

C++ kann auf einer Vielzahl von Rechnerplattformen zu ausführbaren Programmen übersetzt werden. Die mit Abstand am weitesten verbreitete Plattform im Bereich von »Personal Computern« ist »Microsoft Windows«³³ und soll auch hier zum Einsatz kommen. Ein wesentlicher Vorteil von C++ ist die gute Portabilität, d.h. Programme können mit geringem Aufwand so kodiert werden, dass sie auf mehreren Rechnerplattformen zu lauffähigen Programmen übersetzt und ausgeführt werden können (vgl. [Mey97, S. 11]).

Um zu vermeiden, dass die Nutzung des Bewertungswerkzeugs an plattformspezifischen Beschränkungen scheitert, soll eine leichte Übertragbarkeit des Werkzeugs auf die »Open-Source«-Plattform³⁴ »Linux«³⁵ ein Entwurfsziel sein.

Da die Verwendung von Programmiersprachen, wie Java und C# gegenüber C++, u.a. aufgrund der stark vereinfachten Speicherverwaltung, starke Erleichterungen in der Programmierung mit sich bringen (vgl. [Mey97, S. 30]), soll hier C# neben C++ verwendet werden. C# bietet bei Nutzung von Microsoft Windows als Betriebssystem den Vorteil, dass die Microsoft.NET-Infrastruktur genutzt werden kann, die zahlreiche Funktionen z.B. zur Gestaltung von Benutzungsoberflächen oder Anbindung von Datenbanken bietet. Zur Speicherung von Modellen und Berechnungsergebnissen wird ebenfalls ein Produkt von Microsoft verwendet – der Microsoft SQL-Server 2000.

Eine wahrscheinlich ebenso geeignete – alternative – Auswahl entsteht durch den Austausch von C# und Microsoft.NET gegen Java zusammen mit einer geeigneten Infrastrukturkomponente von Microsoft Windows gegen Linux und vom Microsoft SQL-Server gegen z.B. eine Oracle Datenbank. Die Auswahl von Microsoft-Produkten wurde an dieser Stelle aufgrund der besseren Verfügbarkeit getroffen und stellt keine Wertung gegenüber den alternativ genannten Komponenten dar.

³²Siehe <http://msdn.microsoft.com/vcsharp/programming/language/>.

³³Siehe <http://www.microsoft.com/windows>.

³⁴Unter »Open-Source« kann zunächst die freie Einsehbarkeit des Quellcodes einer Software verstanden werden (für eine präzisere Definition siehe <http://www.opensource.org/docs/definition.php>). Die Regelungen zur Verwendbarkeit werden häufig durch Lizenzmodelle festgelegt (siehe z.B. <http://www.opensource.org/licenses/>).

³⁵Siehe z.B. <http://www.linux.org>.

Die hohen Anforderungen in Bezug auf erlaubte Modellgröße und Effizienz führten schnell zur Auswahl allgemeiner Programmiersprachen (konkret C++) zur Bewertung der Modelle von Auftragsabwicklungsprozessen. Die durch die GoM-Kategorie WIRTSCHAFTLICHKEIT getriebene Entscheidung kollidiert jedoch mit einer weiteren Anforderung aus eben dieser GoM-Kategorie, nämlich der wirtschaftlichen Modellerstellung (z.B. fehlen alle in Anforderung 1 aufgeführten Modellelemente). Von Verständlichkeit innerhalb der Anwendungsdomäne (siehe Kapitel 5) kann ebenso wenig die Rede sein, wie von einem intuitiven Modellaufbau (GOM: KLARHEIT).

Eine Lösung dieses Dilemmas kann erreicht werden, wenn mehrere Sprachen in ihren jeweiligen Stärken kombiniert werden. Es soll also eine der Sprachen aus Kapitel 5.2 ausgewählt werden, so dass eine Geschäftsprozessmodellierungssprache für die Modellerstellung, jedoch für die Bewertung eine allgemeine Programmiersprache, genutzt werden kann.

Diese neue Anforderung führt unmittelbar zur Auswahl von UML als Modellierungssprache, da UML gerade im Bereich der Softwareerstellung große Stärken aufweist und sogar zur Erzeugung von C++-Quellcode eingesetzt werden kann³⁶. Auf eine Beschreibung der UML soll an dieser Stelle verzichtet werden, da es zu diesem Thema sehr viel Literatur gibt, die vom Softwareentwurf (vgl. z.B. [Boo94], [BRJ99]) über Fragen der Umsetzung in konkreten Code mit einer bestimmten Entwicklungsumgebung (vgl. z.B. [SS99]) bis hin zur Erzeugung von Simulations- aus UML-Modellen (vgl. [AS00]) reicht.

Im Folgenden wird ausschließlich der UML-Bestandteil des »Business Process Definition Meta-Model« (siehe Abbildung 5.3) genutzt, da hierin bereits alle an dieser Stelle erforderlichen Konzepte enthalten sind. Eine komplementäre Nutzung weiterer Bestandteile des »Business Process Definition Meta-Model« ist möglich.

5.4. Zusammenfassung

Um Auftragsabwicklungsprozesse modellieren und bewerten zu können, bietet die Simulation unterschiedliche Techniken zur Auswahl. Die Anforderungen im Hinblick auf Detaillierung und Bewertungseffizienz führen schnell auf die dynamische ereignisorientierte Simulation als Grundlage für das weitere Vorgehen.

Geschäftsprozessmodellierungssprachen jedoch als nahe liegende Repräsentation für Auftragsabwicklungsprozesse erweisen sich u.a. durch ihre Allgemeinheit als nicht ohne Weiteres geeignet, Auftragsabwicklungsprozesse in der gewünschten Form abzubilden. Bereits am Markt existierende Simulationssoftware als nahe liegende Wahl für die Umsetzung der Bewertungskomponente erweist sich durch mangelndes Vorhandensein erforderlicher Modellelemente (z.B. für die Produktbeschreibung) und Risiken bzgl. Laufzeit und etwaiger Implementierungsbeschränkungen ebenfalls als nicht überzeugend für das vorliegende Einsatzgebiet.

Die im Anschluss geprüften allgemeinen Programmiersprachen zeigen sich jedoch als gut geeignet, sehr komplexe Modelle abzubilden und zu bewerten, jedoch bleibt hier die Tauglichkeit in Bezug auf die Wirtschaftlichkeit der Modellierung zurück.

³⁶Die Klassenstruktur kann durch ein UML-Modellierungswerkzeug (z.B. Rational Rose) in einer Programmiersprache (z.B. C++ oder Java) erzeugt werden. In der *Object Constraint Language* können zusätzliche Hinweise für eine Implementierung notiert werden (vgl. [OMG05, S. 14], [OMG03b]).

Die Synthese aus dem Potenzial von Geschäftsprozessmodellierungssprachen zur Abbildung ebensolcher Prozesse und den Möglichkeiten, in allgemeinen Programmiersprachen sehr komplexe Modelle zu bewerten, führt auf den im folgenden Kapitel beschriebenen Lösungsweg.

6. Modellierungs- und Bewertungsansatz

Die in Kapitel 4 formulierten Anforderungen an einen Modellierungsansatz sowie die Ergebnisse aus Kapitel 5 führen, hier noch einmal zusammengefasst, auf die Notwendigkeit, einen Modellierungs- und Bewertungsansatz mit folgenden wesentlichen Eigenschaften zu entwerfen, um die gestellten Aufgaben lösen zu können:

- ▷ Möglichkeit der Abbildung der Prozesse zwischen Beauftragung durch den Kunden und Auslieferung des Produktes.
- ▷ Möglichkeit der Abbildung des Produktaufbaus, Erzeugung von (Kunden-) Aufträgen auf Basis komplexer Regelwerke (Lastgenerator).
- ▷ Möglichkeit zur Abbildung von Planungsprozessen (Kapazitätsplanung, Auftragsverwaltung etc.).
- ▷ Bewertung der Modelle mittels ereignisorientierter dynamischer Simulation.
- ▷ Möglichkeit zur Auswertung auf Basis von Informationen zu jedem Einzelauftrag.
- ▷ Bewältigung großer Datenmengen sowohl auf der Modellseite als auch auf Seiten der durch die Simulation erzeugten Daten.
- ▷ Variabler Abstraktionsgrad der Modellierung.

Aufgrund der großen Variabilität der Aufgabenstellungen, muss ein Ansatz zur Modellierung von Auftragsabwicklungsprozessen an neue Anforderungen angepasst werden können. Der Ansatz muss *erweiterbar*¹ sein.

In den folgenden Abschnitten dieses Kapitels sollen die grundlegenden Konzepte des hier entwickelten Ansatzes dargestellt werden. Die Umsetzung dieser Konzepte in Software wird in Kapitel 7 erläutert. Die in diesem Kapitel dargestellten Konzepte auf der Modellierungsseite werden vollständig von der in Kapitel 7 entwickelten Software unterstützt. Wünschenswerte Erweiterungen des Modellierungsschemas und der Bewertungssoftware werden in Kapitel 9 diskutiert.

6.1. Grundlagen

Die in Kapitel 5 untersuchte Eignung von Modellierungsansätzen bzw. Beschreibungssprachen führte zu dem Ergebnis, dass eine Umsetzung eines Werkzeugs zur Bewertung von Modellen von Auftragsabwicklungsprozessen in einer allgemeinen Programmiersprache erfolgen sollte, um den formulierten Anforderungen an solch eine Lösung

¹MEYER definiert *Erweiterbarkeit* [Mey97, S. 6]: »Extendibility is the ease of adapting software products to changes of specification.«

gerecht werden zu können. Ein Nachteil der Verwendung einer allgemeinen Programmiersprache ist naturgemäß das Fehlen von grundlegenden Elementen aus dem Anwendungsbereich², die für eine Lösung in einem Anwendungsbereich benötigt werden.

Der Modellbau unter direkter Verwendung einer allgemeinen Programmiersprache – das Modell des Auftragsabwicklungsprozesses ist dann z.B. ein Stück C++-Code bzw. ein daraus erzeugtes ausführbares Programm – erfordert zudem Kenntnisse dieser Programmiersprache sowie die Beherrschung von Entwicklungswerkzeugen, die für die Implementierung der zu erzeugenden Konstrukte (Modellbestandteile) erforderlich sind³.

Eine solche Herangehensweise kann zwar die grundsätzliche Eignung der verwendeten Lösungsbausteine, insbesondere die Verwendung einer allgemeinen Programmiersprache, zeigen. Für einen praktischen Einsatz ist sie nicht geeignet, da u.a. die benötigte Expertise im Bereich Programmierung bei den Experten für das Anwendungsgebiet kaum erwartet werden kann (vgl. [FP97, S. 13]). Zusätzliche Fachkräfte sind nötig, wodurch ein wirtschaftlicher Einsatz erschwert wird.

Ein wichtiger Schritt in Richtung eines Modellierungsansatzes, der auf einem solchen detaillierten Modell beruht, ist die Festlegung einer Menge von Parametern⁴, die es erlaubt, mit möglichst geringem Aufwand Modelle von Auftragsabwicklungsprozessen so zu erzeugen, dass die in Kapitel 4 formulierten Anforderungen erfüllt werden können. Um dies zu erreichen, soll in der gewählten Programmiersprache ein Bewertungswerkzeug bzw. Simulator implementiert werden (»OTD-NET«), der eine solche Menge von Parametern einlesen und durch Simulation eine Bewertung unterstützen kann. Die Aufgabe der Modellierung besteht dann nicht mehr in der Erzeugung von Code in einer allgemeinen Programmiersprache, sondern in der Setzung einer Reihe von Parametern.

Die Aufgabe, einen speziellen Simulator für die Bewertung von Auftragsabwicklungsprozessen umzusetzen, beinhaltet die Konstruktion eines Meta-Modells⁵ von diesen Prozessen, welches die oben genannten Parameter umfasst. Dieses Meta-Modell »innerhalb« des Simulationswerkzeugs soll als *abstraktes Meta-Modell* bezeichnet werden. Dieses Meta-Modell kann im Hinblick auf die Auswahl, Anordnung und Benennung der Parameter dem jeweiligen Einsatzzweck angepasst werden. Dieses angepasste Meta-Modell soll als *konkretes Meta-Modell* bezeichnet werden.

Eine oder mehrere Klassen des abstrakten Meta-Modells können den gewünschten Modellelementen (siehe oben), wie z.B. »Händler«, »Markt« oder »Produkt« zugeordnet werden und beschreiben die Eigenschaften dieser Modellelemente. Diese Modellelemente können dabei in einer festgelegten Weise miteinander in Verbindung gebracht werden. Zum Beispiel kann einem Markt eine beliebige Anzahl Händler zugeordnet werden, wobei ein Händler immer nur zu genau einem Markt gehören darf. Die Ab-

²Während für z.B. die Materialflusssimulation solche grundlegenden Elemente relativ leicht zu benennen sind (z.B. Fördermittel, Bearbeitungsstationen, etc.), ist eine solche Modellwelt für Auftragsabwicklungsprozesse erst zu entwickeln.

³Die Spannbreite reicht in diesem Bereich von einfachen Texteditoren für die Programmerstellung bis hin zu komplexen Entwicklungsumgebungen mit Möglichkeiten für graphisch unterstützten Entwurf von Komponenten, Hilfsmitteln für die Laufzeitoptimierung (Profiler), für die Fehlersuche (Debugger), etc..

⁴Dies kann z.B. durch Platzierung und Verknüpfung grafischer Objekte geschehen.

⁵siehe auch [FvL03, S. 39].

bildungen 6.1 und 6.2 zeigen einige wichtige Klassen des abstrakten Meta-Modells im Zusammenhang.

Um bei der Auswahl der Parameter, die für die Modellierung von Auftragsabwicklungsprozessen zur Verfügung gestellt werden, einige Freiheiten zu bieten, kann eine abgeleitete Klassenstruktur definiert werden – das konkrete Meta-Modell –, die für die Erzeugung des Simulationsmodells genutzt wird. So können z.B. Klassen oder Klassenattribute andere Bezeichnungen erhalten oder mehrere Klassen des abstrakten Meta-Modells zu einer Klasse des konkreten Meta-Modells zusammengefasst werden. Sollte z.B. die Unterscheidung in Märkte und Händler in einem Anwendungsbereich nicht erforderlich sein, können diese Klassen zusammengefasst werden⁶. Weiterhin können nicht benötigte Parameter »ausgeblendet« werden.

Das Werkzeug zur Bearbeitung der Klassen⁷ und auch der Objekte der konkreten Meta-Prozessbeschreibung trägt den Namen »Graphical Modeling Environment« (GME). Daher werden diese Klassen im Weiteren als *GME-Klassen* bezeichnet, die Klassen der abstrakten Meta-Prozessbeschreibung als *UML-Klassen*. Die Instanzen der GME-Klassen werden als GME-Objekte bezeichnet. Die UML-Beschreibung der Klassen der abstrakten Meta-Prozessbeschreibung wird direkt in die Programmiersprache C++ übertragen⁸.

Zur Abbildung eines Auftragsabbildungsprozesses wird ein Modell (kurz: *OTD-NET-Modell*) aus GME-Objekten aufgebaut. Jedes dieser Objekte ist eine Instanz einer GME-Klasse. Jeder GME-Klasse kann eine beliebige Anzahl Instanzen von GME-Objekten zugeordnet werden.

Jeder GME-Klasse kann genau eine UML-Klasse zugeordnet werden, einer UML-Klasse können eine oder mehrere GME-Klassen zugeordnet werden⁹.

Einem Objekt »Händler MAHAG« kann eine GME-Klasse »Dealer« zugeordnet werden¹⁰. Dieser GME-Klasse kann wiederum eine UML-Klasse »CDealer« zugeordnet werden, die eine Entsprechung in C++ findet. Das Bewertungswerkzeug – der *Simulator* – interpretiert das aus GME-Objekten bestehende Modell und erzeugt ein Abbild dieses Objektes als Grundlage für Berechnungen, in die Eigenschaften des Händlers einfließen. Nach der vollständigen Interpretation aller GME-Objekte liegt somit ein Modell im Simulator vor, welches die für die Bewertung relevanten Daten enthält¹¹. Für jedes Objekt im GME-Modell existiert nun ein entsprechendes C++-

⁶Jedem Objekt dieser Klasse des konkreten Meta-Modells würden exakt zwei Objekte des abstrakten Meta-Modells zugeordnet.

⁷MEYER definiert eine *Klasse* im Kontext der objektorientierten Programmierung als (vgl. [Mey97, S. 165]):

A class is an abstract data type equipped with a possibly partial implementation.

⁸BALZERT (vgl. [Bal98, S. 565-566]) unterscheidet zwischen *Exemplar-Ebene* (hier: GME-Objekte), *Typ-Ebene* (hier GME-Klassen, UML-Klassen) und *Meta-Ebene*. Die Erzeugung der *Meta-Prozessbeschreibung* wird also auf der *Typ-Ebene* aus softwaretechnischer Sicht durchgeführt.

⁹In einigen (wenigen) Fällen werden mehrere GME-Klassen einer UML-Klasse zugeordnet. Dies ist z.B. für Wahrscheinlichkeitsverteilungen der Fall. Hier werden alle verfügbaren Verteilungen von einer UML und damit C++-Klasse gehandhabt. In der Regel existiert eine Zuordnung von einer UML-Klasse zu einer GME-Klasse.

¹⁰Die Klasse eines GME-Objektes wird über einen eindeutigen Bezeichner bestimmt. Ein Objekt »Händler MAHAG« ist ein Objekt, welches einen Klassenbezeichner für die Klasse »Dealer« trägt und mit dem ein Attribut »Name« verknüpft ist, welches den Wert »Händler MAHAG« aufweist.

¹¹Das AAP-Modell kann Informationen enthalten, die z.B. für Auswertungszwecke benötigt werden,

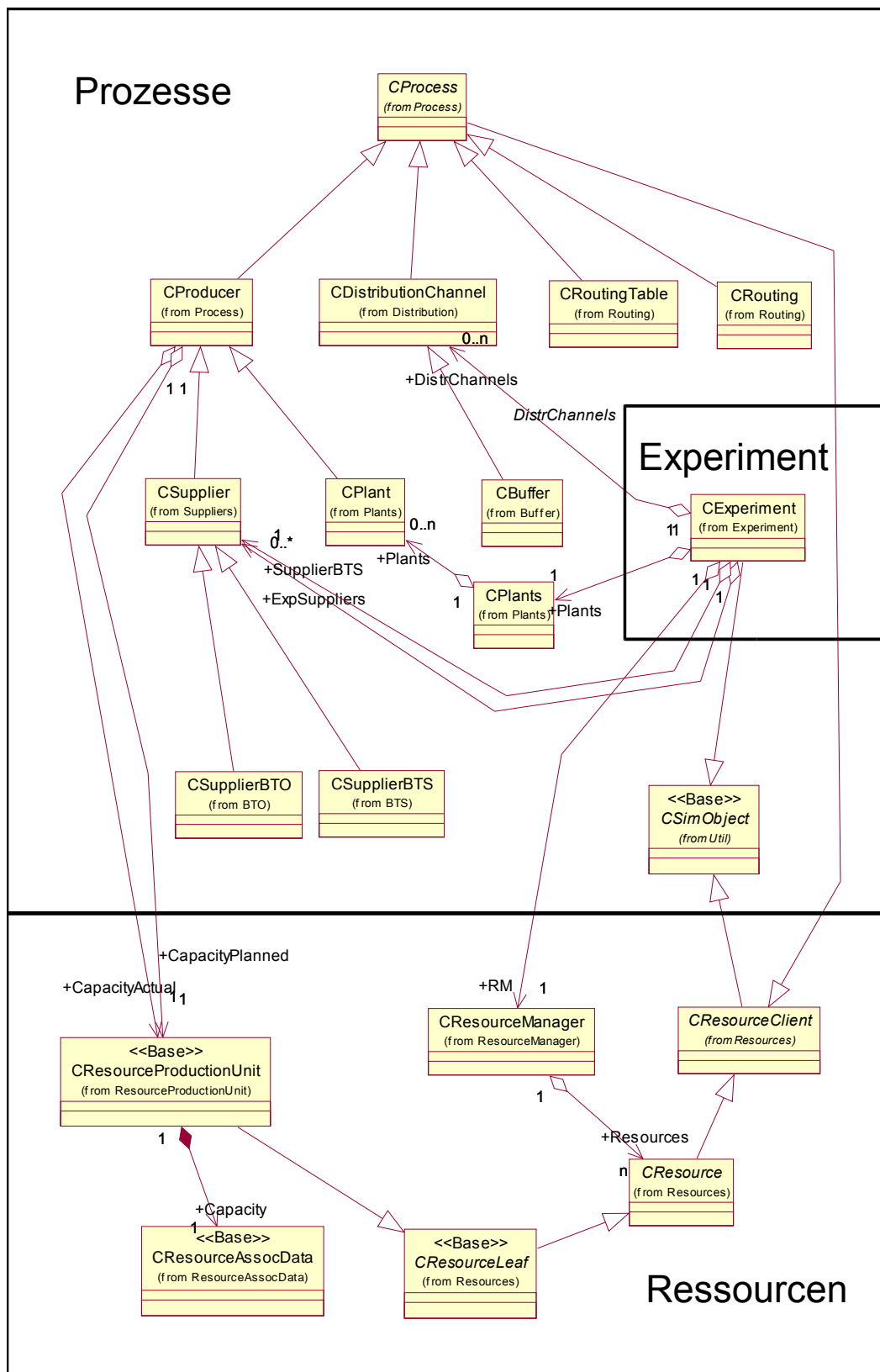


Abbildung 6.1.: Klassenstruktur Prozesse

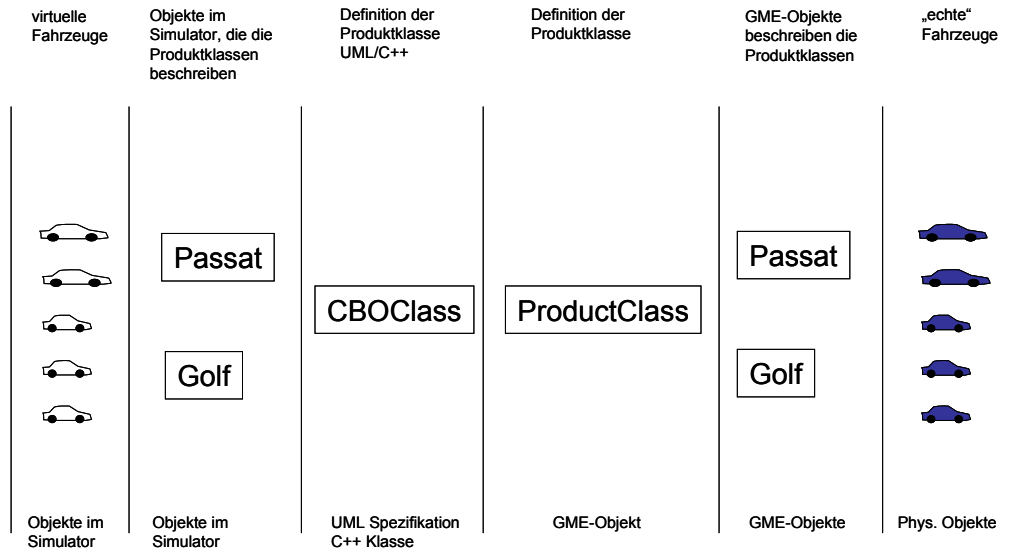


Abbildung 6.3.: UML-, GME-Klassen und Objekte im Zusammenhang

Objekt.

In den Abbildungen 6.1 und 6.2 wurde eine Reihe von Kategorien gebildet, die Klassen bündeln, die in einem besonderen Zusammenhang stehen. Abbildung 2.4 (siehe Seite 13) zeigt beispielhaft die Prozesse der Auftragsabwicklung sowie angrenzender Prozesse in der Dortmunder Prozesskettennotation (vgl. [KLS02]), die gegenüber der abstrakteren Darstellung in Abbildung 2.2 (siehe Seite 8) die Aktivitäten einzelner Akteure aufzeigt¹².

Die Abweichungen der Kategorien der Parameter des Dortmunder Prozesskettenparadigmas in Abbildung 5.2 (Seite 83) und denen der Abbildungen 6.1 und 6.2 entstehen durch die Notwendigkeit, hier Informationen explizit zu repräsentieren, die in der Prozesskettendarstellung nur implizit vorhanden sind. Zu diesen Informationen zählen Steuerungsregeln in einer »ausführbaren« Form und auch Informationen über Produkte und deren Zusammensetzung (Produktstruktur und Stückliste).

Die Prozesskettendarstellung fokussiert die intuitive Darstellung der Zusammenhänge zwischen den Prozesskettenelementen. Der hier vorgestellte Ansatz hingegen legt den Schwerpunkt auf die Bewertung der Prozesse und ergänzt in dieser Hinsicht das Prozesskettenparadigma.

aber keinen Eingang in die Bewertung finden. Diese Informationen werden vom Simulator ignoriert.

¹²Abbildung 2.4 kann auch als UML Activity-Diagramm (siehe [OMG05, S. 285]) nachgebildet werden. Allerdings ist diese Diagrammform nicht ausschließlich auf die Beschreibung von Prozessen abgestellt – viele Informationen aus Abbildung 2.4 (z.B. der Zeitverbrauch von Prozessen) müssen durch »Hilfskonstrukte« nachgebildet werden, worunter die Lesbarkeit der Diagramme leidet.

6.2. Lastgenerator

Die Aufgabe der Komponente *Lastgenerator*¹³ besteht in der Erzeugung von Aufträgen, die die *Systemlast* konstituieren. Diese Aufträge werden in der Simulation von Händlern, vom Hersteller selbst (z.B. Vertrieb) und von Kunden spezifiziert und bestimmen durch Anzahl und konkrete Gestaltung wesentlich die Ergebnisse der Simulation und somit die Aussagen, die über das abgebildete System getroffen werden.

Die vom Lastgenerator zu verarbeitenden Eingangsdaten (Parameter) sollen sein:

- ▷ **Produktbeschreibung:** Wie werden Produkte bzw. Produktklassen definiert, welche Eigenschaften machen die Produktbeschreibung aus?
- ▷ **Einbauraten:** Mit welcher Häufung treten die jeweiligen Eigenschaften in den Produkten auf?
- ▷ **Konstruktionsregeln:** Welche Eigenschaften erzwingen die Anwesenheit weiterer Eigenschaften? Welche Eigenschaften dürfen nicht mit anderen Eigenschaften kombiniert werden?

Im folgenden Kapitel wird die Rolle dieser Parameter beschrieben zusammen mit der Vorgehensweise zur Erzeugung von Aufträgen, die sowohl einzeln betrachtet – im Hinblick auf Konstruktionsregeln, als auch in ihrer Summe – in Bezug auf Einbauraten, den Vorgaben¹⁴ weitgehend entsprechen.

6.2.1. Produktbeschreibung

In der oben stehenden Aufzählung wurde bereits der Begriff *Produkt*¹⁵ verwendet. Für die Entwicklung eines Ansatzes zur *Produktbeschreibung* ist es offenbar erforderlich zu definieren, was unter einem Produkt verstanden werden soll. Ziel ist es dabei nicht, eine umfassende ontologische Herleitung der verwendeten Begriffe durchzuführen (siehe hierzu z.B. [PCDM⁺04]), sondern dies insoweit durchzuführen, dass eine sichere Identifikation der Entsprechungen der Begriffe im Anwendungsfall möglich ist.

Die von Unternehmen produzierten Güter können als *Produkte* bezeichnet werden (vgl. [Wöh93, S. 555]), wobei umgangssprachlich sowohl die konkrete Instanz als auch eine Klasse von Produkten gemeint sein kann. Im Weiteren wird zur besseren Unterscheidung von *Produkten* und *Produktklassen* gesprochen.

Über den gesamten »Lebenszyklus« (vgl. [Czi89, K1]) einer Produktklasse werden Informationen erzeugt, auf die im weiteren Verlauf zugegriffen werden soll. Begonnen mit Skizzen eines Designers über Entwürfe in CAD-Systemen bis hin zu den Informationen, die für Fertigung und Vertrieb der Fahrzeuge benötigt werden (vgl. [BG97a, Y33]). Mit diesen Aufgaben sind verschiedene Sichten auf die angesammelten Informationen verbunden. Die Gesamtheit der Informationen wird *Produktdokumentation*

¹³Die Komponente *Lastgenerator* wird durch die Klassen der Kategorie »Produktstruktur« in Abbildung 6.1 realisiert.

¹⁴Diese Vorgaben entsprechen den Parametern des Lastgenerators.

¹⁵Da der vorgestellte Ansatz nicht prinzipiell auf die Analyse von Auftragsabwicklungsprozessen von Automobilherstellern beschränkt ist, wird hier anstelle des Begriffes »Fahrzeug« der allgemeinere Begriff »Produkt« verwendet.

genannt (vgl. [Kof03a, S. 7]).

Die Anforderungen an die Produktdokumentation im Rahmen des in dieser Arbeit präsentierten Ansatzes sind weniger umfassend, als die Anforderungen, denen die Hersteller von Standardsoftware in diesem Bereich begegnen müssen:

- ▷ Die Repräsentation logistisch relevanter Informationen steht im Fokus. Die Integration mit CAD-Systemen ist nicht zwingend erforderlich¹⁶.
- ▷ Die Abbildung der Strukturen der Automobilindustrie steht im Vordergrund. Obwohl eine Eignung des hier präsentierten Schemas für andere Branchen als die Automobilindustrie vermutet werden kann, werden keine Anforderungen aus anderen Industriezweigen berücksichtigt.
- ▷ Es müssen nur die für den Gestaltungs- und Bewertungsansatz relevanten Informationen abgebildet werden. Es wird keine Verwaltung von Stammdaten umgesetzt.

Die Adaption von Beschreibungsschemata für die Produktstruktur eines Automobilherstellers für die Gestaltung und Bewertung von Auftragsabwicklungsprozessen bringt gewichtige Nachteile mit sich, da die Schemata der Automobilhersteller nicht notwendig untereinander kompatibel sind – der Ansatz wäre aufgrund proprietärer Anteile nur für einen Hersteller nutzbar und zudem nur schwer mit dem gewählten objektorientierten Ansatz vereinbar¹⁷.

Die Verwendung eines Schemas zur Produktdokumentation eines Softwareherstellers, wie es z.B. von SAP angeboten wird (PVS), ist vor allem aufgrund der enormen Komplexität dieser Repräsentationen kaum möglich. Die Beschränkung auf diejenigen Informationen der Produktstruktur, die für die Gestaltung und Bewertung von Auftragsabwicklungsprozessen relevant sind, ermöglicht es, eine Repräsentation zu erzeugen, die zwar einerseits Informationen aus vielen operativen Systemen der Automobilhersteller integriert, aber andererseits noch überschaubar bleibt. Diese (relative) Einfachheit der Darstellung als Ergebnis eines Abstraktionsprozesses bei gleichzeitig umfassender Abbildung aller relevanten Informationen für die vorliegenden Aufgabenstellungen ist ein wesentliches Ergebnis dieser Arbeit.

Es soll im Weiteren versucht werden, eine formale Repräsentation eines *Produktes* im Anwendungsfall »Auftragsabwicklungsprozess« zu entwickeln, die die hierfür relevanten Eigenschaften von Produkten einfängt, ohne dass der Anspruch erhoben wird, ein Produkt »an sich« abzubilden. Die hier definierte Produktbeschreibung ist also nicht allgemeingültig.

Aufgrund der Vielzahl von Bezeichnungen, die von den verschiedenen Automobilherstellern verwendet werden, um ähnliche Dinge zu beschreiben, wird hier eine »neutrale« Terminologie insoweit verwendet, als dass hier von »Produkten« die Rede ist und nicht von »Fahrzeugen«, obwohl der Fokus der Betrachtung auf der Auftragsabwicklung der Automobilindustrie liegt. Der wichtigste Grund hierfür ist der Wunsch,

¹⁶In einigen Bereichen der Logistik, wie z.B. der Versorgungsplanung, spielen geometrische Eigenschaften von Teilen eine Rolle, daher kann eine Integration von Informationen aus CAD-Systemen sinnvoll sein. Die Berücksichtigung solcher Informationen ist im hier vorgestellten Informationsmodell vorgesehen (siehe Abbildung E.13).

¹⁷In den Schemata werden z.B. oft die hierarchischen Beschreibungsebenen und die Zuordnung von Informationen zu diesen Ebenen festgeschrieben. Dies ist unter anderem erforderlich, um die Interoperabilität mit anderen Komponenten der jeweiligen IT-Systemlandschaften sicherzustellen.

die automobilindustriespezifischen Bestandteile des Modellierungs- und Bewertungsansatzes möglichst gering zu halten. Des Weiteren sollen irreführende Assoziationen innerhalb der Automobilindustrie vermieden werden.

Die Verwendung von innerhalb einer Anwendungsdomäne bekannten Termini in einem Modellierungsansatz führt schnell zur Vermutung, dass ein bestimmtes Konzept des Modellierungsansatzes ein ebenso benanntes Konzept des jeweiligen Unternehmens abbildet. Diese vermutete Identität kann sowohl eine inkorrekte Modellierung als auch Akzeptanzprobleme für die Ergebnisse einer Untersuchung nach sich ziehen.

Unter einem *Produkt* soll hier ein Leistungsobjekt verstanden werden, welches im hier diskutierten Zusammenhang als *Produktspezifikation* einen Kundenwunsch repräsentiert, der in einem *Auftrag* abgebildet, durch die Prozesse eines Unternehmens transformiert und letztlich nach Durchlaufen der Fertigungs- und Distributionsprozesse an den Kunden übergeben wird.

Übertragen in die Welt der Automobilherstellung ist ein konkretes Automobil ein *Produkt*, der Fahrzeugtyp – z.B. »Volkswagen Golf«, »Opel Astra« oder »Ford Focus« – die *Produktklasse*. In einem Modell sind nicht alle Unterschiede zwischen Produktklassen von Belang. Um die relevanten Unterschiede zwischen Aufträgen für ein Modell beschreiben zu können, wird hier ein zweigeteilter Ansatz vorgeschlagen: Aufträge werden genau einer *Produktklasse* zugeordnet und werden zudem durch eine Menge von *Eigenschaften* beschrieben. Im Extrem kann so allen möglichen Auftragspezifikationen eine eigene Produktklasse zugeordnet werden¹⁸, bzw. genau eine Produktklasse für alle Aufträge zusammen mit einer jeweils unterschiedlichen Menge von Eigenschaften.

Gegebene Produktbeschreibungen eines Automobilherstellers können also auf viele verschiedene Weisen modelliert werden (siehe Beispiel 6.1). Dieser Freiheitsgrad in der Abbildung kann z.B. genutzt werden, um die Struktur für eine automatische Modellerstellung auf Basis von Daten aus den operativen Systemen eines Herstellers zu optimieren.

Beispiel 6.1 *Die Produktklasse steht in eckigen Klammern, die Eigenschaften eines konkreten Fahrzeuges angehängt in geschweiften Klammern (sofern definiert).*

1. [*Golf_75KW_Motor_Automatikgetriebe_Navigation*]
2. [*Golf_75KW_Motor_Automatikgetriebe_Navigation_BiXenon*]
3. [*Golf*]{*75KW_Motor, Automatikgetriebe, Navigation, BiXenon*}
4. [*Golf*]{*75KW_Motor, Automatikgetriebe, Navigation*}
5. [*Golf*]{*141KW_Motor, Automatikgetriebe, Navigation*}

Wird die Beschreibung des vom Kunden gewählten Produktes in einem Auftrag ausschließlich durch die Zuordnung einer Produktklasse geleistet, ist die Anzahl unterschiedlicher spezifizierbarer Aufträge genau die Anzahl der Produktklassen in einem Modell. Die *Instanzen* der Produktklassen tragen keine weiteren Informationen als diejenigen, die der Klasse zugeordnet werden.

¹⁸Es resultieren viele Milliarden von Produktklassen bei einer für deutsche Hersteller typischen Produktkomplexität (siehe Kapitel 2.1).

Der offensichtliche Nachteil dieser Vorgehensweise ist die Notwendigkeit, alle Unterschiede zwischen Produkten, die in einem Modell abgebildet werden sollen, in jeweils eigene Produktklassen zu kodieren. Eine vollständige Beschreibung aller möglichen Produktklassen im Bereich der Automobilindustrie und deren Unterscheidung in einem Modell ist nur schwer durchführbar.

Eine starke Vereinfachung lässt sich erreichen, wenn Informationen zu den Merkmalen von Produkten in den Instanzen der Produktklassen kodiert werden, wie in den Zeilen drei und vier in Beispiel 6.1. Einem Auftrag wird hier eine Produktklasse zugeordnet – hier *Golf* – und zudem eine Menge von *Eigenschaften*, die im Modell eine Rolle spielen sollen. Diese Zweiteilung der Informationszuordnung in Klassen- und Instanzinformationen ermöglicht eine drastische Reduktion der Anzahl von Produktklassen, die definiert werden müssen, um typische Produktbeschreibungen der Automobilindustrie abzubilden.

Die Verteilung von Informationen auf diese Art ermöglicht ein hohes Maß an gestalterischem Spielraum in der Modellerstellung. In Beispiel 6.2 wird gegenüber Beispiel 6.1 die Information über die Motorisierung in der Produktklasse kodiert und resultiert in einer vergrößerten Anzahl von Produktklassen. Wie viele Produktklassen in einem Modell definiert werden, steht im Ermessen der Person, die das Modell erstellt. Für jede gewählte Verteilung von Informationen auf Produktklassen und Eigenschaften lassen sich Modelle mit genau einer Produktklasse und auf der anderen Seite ohne Eigenschaften (aber potenziell viele Produktklassen) angeben, die gleiche Informationen beinhalten.

Beispiel 6.2 *Erweiterung der Informationen, die als Bestandteil der Produktbeschreibung definiert werden.*

1. $[Golf_75KW_Motor]\{Automatikgetriebe, Navigation, BiXenon\}$
2. $[Golf_75KW_Motor]\{Automatikgetriebe, Navigation\}$
3. $[Golf_141KW_Motor]\{Automatikgetriebe, Navigation\}$

Eine in der Informatik, aber auch in anderen Disziplinen, sehr verbreitete Beschreibungsform für die Relationen von Klassen ist die *Baumstruktur* (vgl. z.B. [ea95, S. 200]). Diese Form der Darstellung hat neben der hohen intuitiven Zugänglichkeit für menschliche Betrachter den Vorteil, Informationen redundanzarm zu speichern und eignet sich weiterhin sehr gut für Rechenoperationen über Eigenschaften, wie sich im Weiteren zeigen wird.

Die Darstellung von Produktklassen in einer Baumstruktur eröffnet neben der Möglichkeit der schrittweisen Konkretisierung von Produkteigenschaften auch die Möglichkeit, unterschiedliche Definitionsschemata für die Erzeugnisse eines Unternehmens abzubilden. Um die Besonderheiten der Märkte, die von einem Automobilunternehmen beliefert werden, geeignet berücksichtigen zu können, passen die Unternehmen die Möglichkeiten zur Produktkonfiguration, die den Kunden angeboten werden, dem Markt an:

- ▷ Es wird die Möglichkeit zur individuellen Auftragskonfiguration angeboten (build-to-order) oder nicht.

- ▷ Die im Markt verfügbaren Karosserieformen, Motoren, Getriebe und sonstige Ausstattungen werden durch den Hersteller festgelegt.
- ▷ Produktnamen werden teilweise marktspezifisch festgelegt¹⁹.
- ▷ Ausstattungspakete werden den vermuteten marktspezifischen Präferenzen angepasst²⁰.

6.2.2. Objektorientierte Struktur

Eines der Kernkonzepte objektorientierter Programmierung, die Vererbung, soll auch für die Definition von Produktklassen zum Einsatz kommen und so die Möglichkeit einer effizienten Beschreibung der Erzeugnisse eines Unternehmens bieten. Der hier vorgestellte Ansatz ermöglicht die Nutzung des Vererbungskonzeptes auf zwei Ebenen: Erstens in der Klassenstruktur im Quellcode des Simulationswerkzeugs²¹ (siehe Abbildung 6.1, Seite 97) und zum zweiten durch Vererbung von Attributwerten zwischen Produktklassen²². Der in Abbildung 6.4 präsentierte kleine Ausschnitt des Beschreibungsschemas zeigt letztere Form der Vererbung am Beispiel.

Das Modell beschreibt einen Teil der (fiktiven) Produktstruktur des »Volkswagen Golf«, daher wird als Wurzel des Beschreibungsbaumes eine Produktklasse *Golf* erzeugt. Verbunden mit dem Wurzelobjekt der Produktstruktur ist die Information über den Absatz über alle beschriebenen Produkte. Im Beispiel ist dieser Absatz als 40.000 Einheiten definiert. Für die abgeleiteten Klassen werden Anteile am Absatz der Basisklasse spezifiziert, also z.B. 56 % von 40.000 Einheiten (= 22.400) im Falle *Golf Deutschland*; der Absatz von Fahrzeugen des Typs *Golf Basis* beträgt dann 44 % von 22.400 (= 9.856). Diese Notation bringt gegenüber der durchgängigen Verwendung von Absatzzahlen im Modell den Vorteil mit sich, dass eine Änderung der Absatzzahlen ohne Verschiebung zwischen den einzelnen Fahrzeugtypen leicht umsetzbar ist²³.

Abgeleitete Produktklassen können den gesamten Absatz einer übergeordneten Klasse auf sich vereinen, wie in obigem Beispiel. In einem solchen Fall kann die übergeordnete Klasse als *abstrakt*²⁴ gekennzeichnet werden mit der Folge, dass durch Normierung der Absatzanteile potenzielle Rundungsfehler vermieden werden.

¹⁹Manche Produktnamen wecken in einigen Sprachregionen unerwünschte Assoziationen (der Toyota Yaris heißt in Japan Vitz) oder entstammen der Vulgärsprache (Mitsubishi Pajero im spanischen Sprachraum).

²⁰Pakete haben also u.U. den gleichen Namen, aber anderen »Inhalt«, oder werden nur auf ausgewählten Märkten angeboten.

²¹Eine detaillierte Beschreibung dieser Klassenstrukturen erfolgt im Anschluss.

²²Die Terminologie ist an dieser Stelle etwas verwirrend – Attributwerte werden i.d.R. nicht vererbt, Attribute hingegen schon. Da die Definition der »Produktklasse« ein Objekt der konkreten Meta-Repräsentation ist, werden die Attribute der Produktklassenobjekte in der abstrakten Meta-Repräsentation (in UML/C++) festgelegt. Die Attributwerte hingegen werden an den Objekten vom Typ »Produktklasse« (PRODUCTCLASS) definiert.

Im Kontext der konkreten Modellbeschreibung sind also »Produktklassenobjekte« gemeint, wenn verkürzt von »Produktklassen« gesprochen wird. Um die Unterscheidung im Text zu vereinfachen, werden Klassen der abstrakten Meta-Repräsentation (z.B. CBOCLASS), der konkreten Meta-Repräsentation (z.B. PRODUCTCLASS) und Objekte der Meta-Repräsentation (z.B. *Golf*) typografisch unterschieden.

²³Eine Beschreibung der Absätze als ganze Zahlen, also nicht als Anteile von einem Gesamtabsatz, ist ebenfalls möglich. Beide Darstellungsformen können uneindeutig ineinander überführt werden.

²⁴Es können dann keine Instanzen dieser Klasse erzeugt werden.

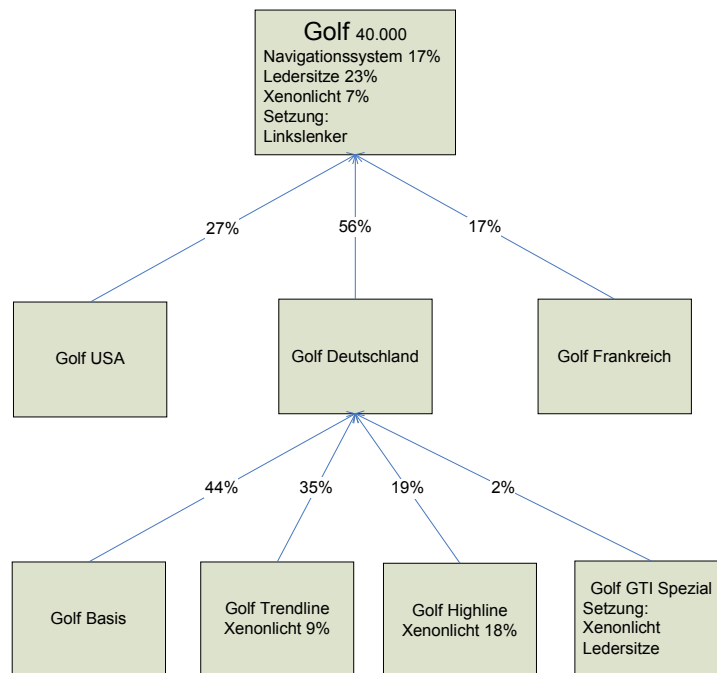


Abbildung 6.4.: Ableitung von Produktklassen

Das Konzept *Vererbung* kann an drei Stellen zum Einsatz kommen:

1. Die Ableitung von Produktklassen. Beispiel: Von der Klasse *Golf* in Abbildung 6.4 wird eine Produktklasse *Golf Spanien* abgeleitet.
2. Die Ableitung von GME-Klassen. Beispiel: Von der Klasse `PRODUCTCLASS` wird eine neue Klasse `PRODUCTCLASSVERSIONINFO` abgeleitet, die ein zusätzliches Attribut trägt, in dem Versionsinformationen abgelegt sind.
3. Die Ableitung von UML- bzw. C++-Klassen. Beispiel: Von der Klasse `CBOClass` wird eine Klasse `CBOClassVersionInfo` abgeleitet (analog der Ableitung von der Klasse `CSimObject` in Abbildung D.5), die die neuen Informationen der GME-Klasse `PRODUCTCLASSVERSIONINFO` aufnimmt und verarbeitet.

Während die letzte Form der Vererbung die »typischen« Möglichkeiten bietet, die in diesem Bereich objektorientierter Programmierung anzutreffen sind (siehe [Str97, S. 301]), werden diese Möglichkeiten im Falle der Ableitung von Produktklassen (Punkt eins obiger Aufzählung) eng begrenzt: In den abgeleiteten Klassen können lediglich Informationen überschrieben werden.

Ist z.B. die Einbaurrate für ein bestimmtes Navigationssystem an der Produktklasse *Golf* als konstant 7 % beschrieben, kann an der Produktklasse *Golf Spanien* ein anderer Wert spezifiziert werden, der die 7 % überschreibt. Diese strenge Beschränkung der Möglichkeiten der Vererbung an dieser Stelle trägt dazu bei, einen einheitlichen Lösungsweg für Erweiterungen des Konzeptes zu etablieren. Sollen also z.B. neue At-

tribute zur Steuerung von Planungsalgorithmen eingeführt werden, ist es erforderlich, eine Erweiterung der GME-Klassen und im Anschluss der UML- bzw. C++-Klassen vorzunehmen.

Ein Ausschnitt der insgesamt weit über einhundert Klassen des Simulationswerkzeugs ist in Form von UML-Diagrammen dem Anhang zu entnehmen. Im Folgenden werden die wichtigsten Attribute der beschriebenen Modellelemente in Tabellen zusammengefasst (siehe z.B. Tabelle 6.7 auf Seite 114). In der ersten Spalte wird der UML bzw. C++ Typ beschrieben. In Zeile eins werden die UML- bzw. GME-Klassen aufgeführt (CBOStructure und PRODUCTSTRUCTURE), zu denen die darunter stehenden Attribute gehören.

Zeile	Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
1		CBOStructure		PRODUCTSTRUCTURE	
2	COrderModifier 1..1 →	.OrderModifier	D.9	.ORDERMODIFIER	E.53
3		CProducer		SUPPLIER	
4	CBOM 1..n →	.BOM	D.8	.BILLOFMATERIALSUPPLIER	E.71

Tabelle 6.1.: Parametertabellen

Die Zeile zwei in Tabelle 6.1 bezeichnet eine eins zu eins Referenz (1..1 →) auf ein Objekt der Klasse COrderModifier. Beim zweiten Beispiel (Zeile vier) werden mehrere Objekte vom Typ CBOM referenziert (1..n). Die Spalten »Attribut« bezeichnen die Namen der Attribute der UML-Klassen und der GME-Klassen. Die Spalten »Abb.« geben an, in welchem Teil des Anhang die Klassen vollständig abgebildet (UML), bzw. beschrieben (GME) werden.

6.2.3. Einbauraten

Der Abbildung 6.4 sind neben den Absatzanteilen für die Produktklassen auch Einbauraten für einige Eigenschaften von Aufträgen zu entnehmen.

Definition 6.1 *Der Anteil von einer Anzahl Aufträgen mit einer bestimmten Eigenschaft wird im Folgenden als Einbaurrate dieser Eigenschaft bezeichnet.*

So ist für die Produktklasse *Golf* eine Einbaurrate für die Eigenschaft *Xenonlicht* angegeben. Unabhängig von den für abgeleitete Klassen angegebenen Einbauraten wird diese Zahl dahingehend interpretiert, dass die Einbaurrate über alle Produktinstanzen von einer von *Golf* abgeleiteten Produktklasse 7 % betragen muss. Wird keine weitere Einbaurrate spezifiziert, beträgt auch die Einbaurrate für die abgeleiteten Klassen 7 %. Im vorliegenden Fall werden aber für drei weitere Produktklassen Einbauraten spezifiziert:

Produktklasse	Anzahl	Einbaurrate	Anzahl Xenon
<i>Golf</i>	40000	7 %	2800
<i>Golf USA</i>	└ 10800	⇒ 3,2 %	⇒ 346
<i>Golf Frankreich</i>	└ 6800	⇒ 3,2 %	⇒ 218
<i>Golf Deutschland</i>	└ 22400	⇒ ?	⇒ ?
<i>Golf Basis</i>	└ 9856	⇒ 3,2 %	⇒ 315

<i>Golf Trendline</i>	⊥ 7840	9 %	706
<i>Golf Highline</i>	⊥ 4256	18 %	766
<i>Golf GTI Spezial</i>	⊥ 448	100 % ²⁵	448

Tabelle 6.2.: Berechnung von Einbauraten

Die durch einen vorangestellten Pfeil (\Rightarrow) gekennzeichneten Einbauraten müssen auf Basis der spezifizierten Parameter abgeleitet werden. Ausgehend von insgesamt 2.800 *Xenonlicht* Ausstattungen und

$$706 + 766 + 448 = 1920 \quad (6.1)$$

davon, die durch die spezifizierten Einbauraten bereits festgelegt sind, verbleiben

$$2800 - 1920 = 880 \quad (6.2)$$

Ausstattungen für die Produktklassen, für die keine Angaben gemacht wurden. Die Anzahl Aufträge beträgt für diese Produktklassen

$$10800 + 6800 + 9856 = 27456. \quad (6.3)$$

Die Einbaurate ergibt sich also als

$$\frac{880}{27456} = 3,2\% \quad (6.4)$$

Es ist nicht erforderlich, die Einbaurate für die Produktklasse *Golf Basis* zu berechnen, da die Absatzanteile dieser Produktklasse vollständig auf die abgeleiteten Klassen verteilt werden²⁶. Im vorliegenden Fall ist eine Lösung für die Berechnung aller erforderlichen Einbauraten für die jeweiligen Produktklassen ableitbar, dies ist jedoch nicht immer der Fall. Die Einbauraten, die als Eingangsgrößen in die oben beschriebene Berechnung eingehen, werden z.B. von Mitarbeitern einer Vertriebsseinheit eines Unternehmens prognostiziert. Aufgrund des hohen Aufwandes der »manuellen« Bestimmung von hunderten von gültigen Einbauraten²⁷, ist eine maschinelle Unterstützung wünschenswert.

Daher ist die Darstellung der Berechnungsergebnisse der Einbauratenberechnung vorgesehen, um noch vor Beginn umfangreicher Berechnungsvorgänge zur Bewertung ganzer Auftragsabwicklungsprozesse eine Korrektur bzw. Anpassung zu ermöglichen. Obgleich das Verfahren zur Berechnung von Einbauraten aus mathematischer Sicht sehr einfach ist und ohne Schwierigkeiten in einer Tabellenkalkulation anwendbar ist, liegt bereits in der Berechnung *effektiver* Einbauraten unter Beachtung aller Informationen zum Produktaufbau ein Mehrwert, da die Nutzung einer Tabellenkalkulation mit hohem manuellem Aufwand an die sich ändernde Produktstruktur angepasst werden muss.

²⁵Bei diesem Modell soll *Xenonlicht* grundsätzlich verbaut werden.

²⁶Die Produktklasse *Golf Deutschland* ist *abstrakt*.

²⁷Das heißt, Einbauraten, für die das oben im Beispiel beschriebene Berechnungsverfahren eine Lösung hat.

6.2.4. Wertverläufe

Sowohl die Einbauraten in Abbildung 6.4 als auch die Beispielrechnung basieren auf einem Wert für die Einbaurrate jeweils einer Eigenschaft. Es ist nahe liegend, dass die Einbauraten für viele, wenn nicht alle Eigenschaften von Aufträgen über die Zeit schwanken. Besonders stark ausgeprägt sind diese Schwankungen, wenn neue Eigenschaften für die Kunden zur Auswahl gestellt werden.

In einem solchen Fall ist die Einbaurrate bis zur Freigabe²⁸ der Eigenschaft Null. Um solche Szenarien, wie auch die Veränderung von Einbauraten, über die Zeit abbilden zu können, werden im Rahmen des vorgestellten Ansatzes *Wertverläufe* (GME-Klassen, die von ASSOCDATA abgeleitet sind) verwendet. Mit Hilfe von Wertverläufen lassen sich Schwankungen von Einbauraten über die Zeit modellieren, d.h. Daten innerhalb des Betrachtungszeitraumes können Werte zugeordnet werden.

Für diese Zuordnung können sowohl Tage als auch Wochen, Monate und Jahre verwendet werden, bzw. Mischformen davon. Eine Einbaurrate kann somit zunächst für Monatsintervalle, dann auch auf Basis von Wochen und folgend auf Tagen spezifiziert werden. Alle oben bzw. im Weiteren geschilderten Berechnungsverfahren können diese Wertverläufe als Eingangsdaten nutzen, auch wenn in diesen Daten verschiedene Bezugsintervalle verwendet werden (z.B. Daten auf Monats- und Wochenebene gemischt).

»Eigenschaften« repräsentieren Informationen über Produktklassen übergreifend; die »Eigenschaftenfamilien« (GME-Klasse PROPERTYGROUP) werden nur einmal für das gesamte Modell definiert. Es ist nahe liegend, dass z.B. die konkrete Eigenschaft *6-Gang-Getriebe* abhängig von der Produktklasse technisch unterschiedlich gestaltet sein kann. Ein Getriebe eines Oberklasse-Fahrzeugs unterscheidet sich stark von einem Getriebe für einen Kleinwagen. Weiterhin ist es wünschenswert, die Einbaurate für das Getriebe zwischen unterschiedlichen Modellen oder auch Absatzmärkten variieren zu lassen.

Um diese Ziele zu erreichen, wird zusätzlich zur »Eigenschaft« (GME-Klasse PROPERTY) das Modellelement »Eigenschaftenspezifikation« (GME-Klasse PROPERTYSPEC) eingeführt, welches Informationen über Eigenschaften spezifisch für eine Produktklasse bündelt. Eine spezifische Einbaurrate für ein *6-Gang-Getriebe* für die Produktklasse *Golf GTI Spezial* kann so in einem Objekt *Einbaurrate 6-Gang-Getriebe Golf GTI Spezial* an einer Eigenschaftenspezifikation *Spezifikation 6-Gang-Getriebe Golf GTI Spezial* definiert werden.

6.2.5. Erzeugung von Aufträgen

Für die Berechnung der Anzahl der Aufträge, die für einen Bezugszeitraum (z.B. eine Woche) zu erzeugen sind, ist es möglich, die Absatzanteile von einer Produktklasse mit den Absatzanteilen der übergeordneten Produktklasse (wiederum ein Objekt des Typs PRODUCTCLASS) rekursiv solange zu multiplizieren, bis die Wurzel der Struktur erreicht ist und somit eine Gesamtanzahl Aufträge mit dem errechneten Wert multipliziert werden kann (am Beispiel *Golf GTI Spezial*):

²⁸Unter »Freigabe« einer Eigenschaft soll verstanden werden, dass ein Kunde diese Eigenschaft für einen Auftrag auswählen kann.

$$2\% \times 56\% \times 40000 = 448 \quad (6.5)$$

Die hohe Anzahl verfügbarer Produktkonfigurationen bringt es mit sich, dass von bestimmten Fahrzeugmodellen in volumenschwachen Märkten unter Umständen nur wenige Fahrzeuge pro Wochen verkauft werden. Da für die Bewertung eines Modells zu bestimmen ist, wie viele Aufträge einer Produktklasse an einem Tag bearbeitet werden sollen, besteht die Aufgabe darin zu bestimmen, wie z.B. 6,5 Aufträge auf sechs Produktionstage zu verteilen sind. Diese Verteilung kann z.B. durch ein stochastisches Verfahren gelöst werden, so dass im Mittel 6,5 Aufträge auf eine Kalenderwoche entfallen – in einer Woche z.B. sechs Aufträge, in der nächsten Woche sieben Aufträge. Werden die Absätze mehrerer Produktklassen zum selben Zeitpunkt aufgerundet, entsteht in Summe eine nicht zu vernachlässigende Abweichung von den vorgegebenen Absatzzahlen.

Um diese Abweichungen zu verringern, wird im vorgestellten Verfahren ein Berechnungslauf durchgeführt²⁹, der durch Koordinierung der Verteilung von Aufträgen über alle Produktklassen garantiert, dass die Abweichung der Summe der Absätze über alle Produktklassen kleiner eins ist.

Die grundlegende Idee zur Erzeugung von Aufträgen, die einerseits keine der Regeln verletzt und andererseits aber in der Verteilung der Eigenschaften den vorgegebenen Einbauraten entsprechen, besteht darin, aus einer Menge von »zufällig« erzeugten regelkonformen Aufträgen diejenigen auszuwählen, die in Summe den Einbauraten entsprechen.

Für die Erzeugung der Aufträge auf Basis der berechneten Absatzanzahlen wird folgende Vorgehensweise vorgeschlagen: Beginne mit Aufträgen ohne Eigenschaften und füge unter Nutzung von Einbauraten und einer Menge von Regeln, die unzulässige Eigenschaftskombinationen ausschließen, Eigenschaft um Eigenschaft hinzu (siehe Abbildung 6.5 auf Seite 110).

Durch die Anwendung der Regeln (siehe unten) werden Aufträge verworfen, die nicht diesen Regeln entsprechen. Die Entfernung dieser Aufträge führt zu, gegenüber den Vorgaben, veränderten Einbauraten. Überschreitet diese Abweichung einen bestimmten Grenzwert, wird diejenige Eigenschaft bestimmt, die die größte Abweichung ausweist. Im nächsten Schritt wird ein nach einer Heuristik³⁰ ausgewählter Auftrag, der diese Eigenschaft besitzt, aus der Menge der bereits erzeugten Aufträge entfernt. Mit einer von den konkreten Einbauraten abhängigen Wahrscheinlichkeit enthält der im Folgeschritt erzeugte »Ersatzauftrag« die fragliche Eigenschaft nicht und eine Verringerung der Abweichung gegenüber der Vorgabe ist erreicht.

Das Ergebnis dieser Vorgehensweise sind Aufträge, die aus jeder Eigenschaftsfamilie genau eine Eigenschaft enthalten und mit steigender Anzahl die spezifizierten Einbauraten weiter annähern³¹.

²⁹Dieser Berechnungslauf kann unabhängig von weiteren Berechnungen im Vorfeld der eigentlichen Bewertung des Auftragsabwicklungsprozesses durchgeführt werden.

³⁰Für eine solche Heuristik können Informationen aus den Konstruktionsregeln herangezogen werden. In dem Beispiel in Tabelle 6.3 (das Beispiel wird in Kapitel 6.2.6 erläutert) sind Aufträge mit 75KW-Motor und 6-Gang-Getriebe »attraktiv« für eine Entfernung, weil das Getriebe zum 75KW-Motor ohne Beachtung von Konstruktionsregeln hinzugefügt werden kann und auf der anderen Seite das 6-Gang-Getriebe unter bestimmten Bedingungen (siehe Tabelle 6.4) bevorzugt ausgewählt wird.

³¹Eine Abweichung von den Einbauraten entsteht z.B. durch die Notwendigkeit, immer genau eine Ei-

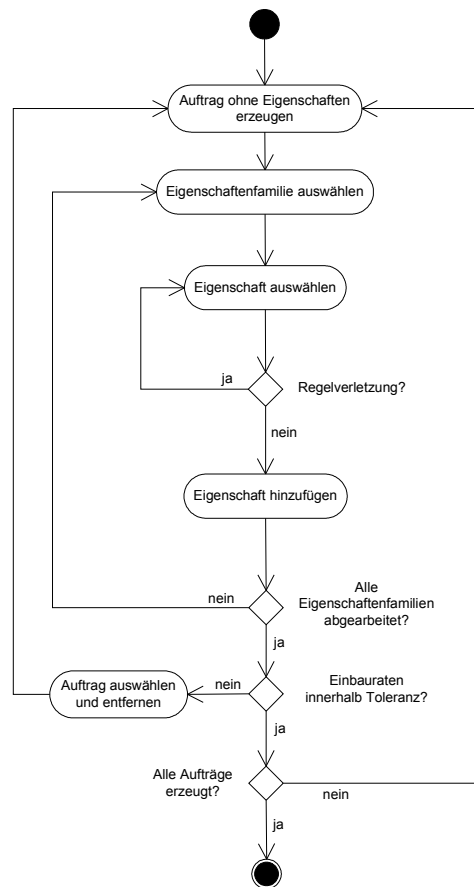


Abbildung 6.5.: Aktivitätsdiagramm Auftragserzeugung (vereinfachte Darstellung)

6.2.6. Konstruktionsregeln

Vor der Bestätigung eines Fertigungstermins für einen Auftrag durch einen Automobilhersteller wird dieser einer Prüfung auf Abweichungen gegenüber einem Regelwerk unterzogen. Dieses Regelwerk beinhaltet in einigen Fällen über zehntausend Regeln³², die sowohl technische Zusammenhänge als auch Marketingentscheidungen reflektieren. So wird z.B. die Verfügbarkeit von bestimmten Eigenschaften in einem Markt für einen Fahrzeugtyp über solche Regeln definiert – eine Entscheidung, die z.B. der

genschaft aus einer Eigenschaftenfamilie für einen Auftrag auszuwählen. Wird z.B. ein Auftrag auf Basis der Einbauraten in obiger Tabelle 6.3 erzeugt, existiert keine Zuordnung von Eigenschaften, dass die spezifizierten Einbauraten exakt in der Menge der Aufträge (hier ein Auftrag) wiedergegeben werden. Für zwei Aufträge existiert eine solche Zuordnung, für drei wiederum nicht. Die mögliche Annäherung an die gewünschten Einbauraten wird sowohl von den konkreten Konstruktionsregeln als auch der Güte der Heuristik zum Entfernen von Aufträgen (siehe oben) beeinflusst.

³²GEBHARDT et al sprechen von 10.000 Regeln für den VW Golf (vgl. [GDM03, S. 1]), die die technische Zusammenhänge repräsentieren. Noch weitere kommen hinzu, die aus dem Bereich Vertrieb vorgegeben werden.

Vertrieb des Unternehmens treffen kann³³. Weiterhin wird durch Regeln formuliert, welche Eigenschaften mit anderen kombiniert werden dürfen. Auch hier können wiederum auch nicht-technische Beweggründe eine Rolle spielen.

Verläuft eine solche Prüfung erfolgreich, entspricht der Auftrag den Regeln des Landesverkaufsprogramms³⁴, kombiniert keine Eigenschaften, die zwar gemäß Landesverkaufsprogramm zusammen bestellt werden können, aber aus technischen (oder sonstigen) Gründen nicht zusammen bestellt werden können. Es ist also durchaus möglich, dass ein gemäß Katalog korrekt spezifizierter Auftrag vom Automobilhersteller abgelehnt wird. In den Katalogen wird also nur eine Untermenge der Konfigurationsregeln für einen Auftrag angegeben.

Die Konstruktionsregeln wirken dabei nicht nur auf Eigenschaften, die von den Kunden spezifiziert werden können, sondern auch auf Eigenschaften, die auf Basis der kundenwählbaren und anderweitig gesetzten Eigenschaften vom Hersteller in den Auftrag spezifiziert werden. So kann z.B. eine Regel lauten: »Wenn im Auftrag ein *Navigationssystem* spezifiziert ist, muss eine *große Batterie* verwendet werden«. Im Rahmen dieser Arbeit werden die Konstruktionsregeln, als bestehend aus drei Teilen, definiert. Die Regel beginnt mit einem booleschen Ausdruck über der Menge der Eigenschaften. Der zweite Teil ist ein Operator *nur mit* oder *nicht mit* und der dritte Teil wiederum ein boolescher Ausdruck über der Menge der Eigenschaften³⁵. Obiges Beispiel in dieser Form: *Navigationssystem nur mit Große Batterie*.

Während aus Sicht des Automobilherstellers die Prüfung dieser Regeln keine Schwierigkeiten bereitet³⁶, wird die Erzeugung von Aufträgen, unter der Maßgabe bestimmte Einbauraten einzuhalten, erheblich erschwert.

Sollen Aufträge unter Beachtung der Konstruktionsregeln generiert werden, ergeben sich zahlreiche Herausforderungen. Die größte dieser Herausforderungen entsteht dadurch, dass die Konstruktionsregeln Abhängigkeiten zwischen den betroffenen Eigenschaften erzeugen.

Produktklasse	Eigenschaftenfamilie	Eigenschaft	Einbauraten
Golf	Motoren	75KW-Motor	50 %
		96KW-Motor	50 %
	Getriebe	6-Gang-Getriebe	50 %
		5-Gang Getriebe	50 %

Tabelle 6.3.: Beispiel Einbauraten

Kommt zu den Einbauraten aus Tabelle 6.3 die Konstruktionsregel in Tabelle 6.4 hinzu, führt eine einfache Umsetzung der Aktivitäten in Abbildung 6.5 bereits zu

³³Eine solche Regel kann auch technische Hintergründe haben (z.B. zu hohe Luftfeuchtigkeit für eine Elektronikkomponente im Verkaufsland) oder durch den Gesetzgeber festgelegt worden sein (Navigationssysteme werden nicht restriktionsfrei vertrieben).

³⁴Unter einem »Landesverkaufsprogramm« sollen alle von einem Hersteller für einen Markt vorgesehenen Fahrzeugmodelle und wählbaren Eigenschaften verstanden werden.

³⁵In den booleschen Ausdrücken können die Operatoren AND, OR und NOT verwendet werden (Klammerung möglich).

³⁶Ein konkreter Auftrag ist gegen alle Regeln zu prüfen. Tritt eine Regelverletzung auf, ist der Auftrag ungültig.

falschen Einbauraten in den erzeugten Aufträgen:

Beispiel 6.3 *Sollen 100 Aufträge gemäß obiger Einbauraten und Regeln erzeugt werden, sind jeweils 50 Eigenschaften 75KW-Motor und 96KW-Motor und 50 Eigenschaften 5-Gang-Getriebe und 6-Gang-Getriebe den 100 Aufträgen zuzuordnen, so dass jeder Auftrag genau einen Motor und genau ein Getriebe erhält³⁷. Im ersten Schritt werden die Motoren den Aufträgen zugeordnet.*

Im zweiten Schritt werden den Aufträgen Getriebe zugeordnet³⁸. Soll einem Auftrag mit einem 75-KW-Motor ein Getriebe zugeordnet werden, kann aufgrund der Einbauraten ein 6-Gang-Getriebe gewählt werden, ohne dass eine Konstruktionsregel durch diese Kombination verletzt wird. Es wird angenommen, dass noch nicht alle 6-Gang-Getriebe zugeordnet sind.

Wird nun zu einem Auftrag mit 75-KW-Motor ein 6-Gang-Getriebe gewählt, kann im weiteren Verlauf des Algorithmus für mind. einen Auftrag mit einem 96KW-Motor kein Getriebe zugeordnet werden, so dass keine Konstruktionsregel verletzt wird – die »fehlenden« 6-Gang-Getriebe sind »fälschlicherweise« mit 75-KW-Motoren kombiniert worden.

Nr	Produktklasse	Konstruktionsregel
1	Golf	96KW-Motor nur mit 6-Gang-Getriebe

Tabelle 6.4.: Erweiterung von Beispiel in Tabelle 6.3 um eine Konstruktionsregel

Nr	Produktklasse	Konstruktionsregel
2	Golf	75KW-Motor zwingt 5-Gang-Getriebe

Tabelle 6.5.: Aus Einbauraten abgeleitete Konstruktionsregel

In diesem sehr einfach konstruierten Beispiel kann das Problem leicht durch Hinzufügen einer zusätzlichen Regel (siehe Tabelle 6.5) gelöst werden. Jeder Eigenschaft aus der Eigenschaftenfamilie Motoren wird nun eine Eigenschaft der Familie Getriebe zugeordnet. Die Ableitung einer zusätzlichen Regel ist in diesem Fall noch leicht möglich. Bei komplexeren Modellen jedoch verunmöglicht die große Anzahl der benötigten Regeln eine Nutzung dieses einfachen Lösungsansatzes.

Ein wichtiger Baustein für die Lösung der Aufgabe regelkonforme Aufträge zu erzeugen, besteht in der Nutzung von Informationen, die über die Produktstruktur direkt vorhanden sind bzw. daraus abgeleitet werden können, um die relevante Regelmenge auf ein handhabbares Maß zu verkleinern. Hierzu soll eine Kategorisierung der Regeln durchgeführt werden in:

³⁷Diese Anforderung folgt aus der Zuordnung der Eigenschaften zu Eigenschaftenfamilien (hier: *Motoren* und *Getriebe*). Jeder Auftrag muss genau eine Eigenschaft aus jeder definierten Eigenschaftenfamilie aufweisen.

³⁸Die Eigenschaftenfamilien werden derart angeordnet, dass Familien, die in vielen Konstruktionsregeln eine Rolle spielen, zuerst abgearbeitet werden. Im konkreten Fall würde die Reihenfolge auf Basis der Reihenfolge im Modell (Listenposition) gesetzt, weil aus der Konstruktionsregel in Tabelle 6.4 keine eindeutige Reihenfolge abgeleitet wird.

1. Regeln, die die Verfügbarkeit von Ausstattungen für bestimmte Produktklassen festlegen. Regeln dieser Kategorie entsprechen dem Muster: .EIGENSCHAFTEN X *nur mit* .EIGENSCHAFTEN Y. Wie oben bereits definiert, können in beiden Teilen der Regel boolesche Verknüpfungen enthalten sein. Die Prämisse einer solchen Regel kann keine Eigenschaften enthalten. Diese Regel gilt dann für alle Instanzen einer Produktklasse (siehe Tabelle 6.6).
2. Regeln, die die Ausstattung eines Fahrzeugs festlegen, wenn keine Ausstattung durch einen Kundenwunsch geändert wird.
3. Regeln, die Inhalte von Ausstattungspakten bestimmen.
4. Regeln, die Kombinationen von Eigenschaften aus technischen oder sonstigen Gründen ausschließen.

Nr	Produktklasse	Konstruktionsregel
3	<i>Golf GTI Spezial</i>	<i>zwingt Xenonlicht</i>

Tabelle 6.6.: Konstruktionsregel ohne Eigenschaften in der Prämisse

Die Regeln der Kategorien 1, 2 und 3 können durch Integration dieser Regeln in die Produktbeschreibung so vorverarbeitet werden, dass durch das Erzeugungsverfahren bereits sichergestellt werden kann, dass keine Regelverletzung auftritt. Die Regeln der Kategorie 4 hingegen erfordern die Umsetzung eines Verfahrens, welches Regelverletzungen erkennt und in einem solchen Auftrag durch Austausch der »problematischen Eigenschaft(en)« eine regelkonforme Eigenschaftenkombination sucht (siehe oben).

6.2.7. Stücklisten

Stücklisten sind für die Abbildung von Auftragsabwicklungsprozessen insofern ein wichtiges Element, als dass diese die Grundlage für die Berechnung der tatsächlichen Materialbedarfe bilden, die aus der Marktnachfrage resultiert. Aus Sicht des Automobilherstellers sind Stücklisten ein Teil der Produktdokumentation und sind daher an dieser Stelle der Beschreibung des Modellierungsansatzes aufgeführt.

Bill of material (BOM)³⁹ A list of all of the items that make up a product or assembly. The list is complete for its given purpose, has a formal structure, and states the name, quantity, and unit of measure for each component.

In der Literatur werden zahlreiche Schemata beschrieben, Stücklisten aufzubauen. Beispiele für solche Schemata sind z.B. Mengen-, Struktur- und Baukastenstücklisten⁴⁰.

Während die Produkteigenschaften, die den Kunden zur Auswahl gestellt werden, eher nach Vertriebsanforderungen kategorisiert werden und die Funktion nahe legen (z.B. *Navigationssystem, Xenonlicht*), kann einer Stückliste entnommen werden, wel-

³⁹Quelle: Glossar auf »SAP INFO« Web-Site <http://www.sap.info>.

⁴⁰Für eine Darstellung dieser Schemata siehe z.B. [BG97a, F 33].

che Komponenten⁴¹ in ein Produkt verbaut werden müssen, um diese Funktion zu realisieren (z.B. Kompressor, Klima-Steuergerät, diverse Halter, Kabel, etc.).

Aufgrund der vielfältigen Möglichkeiten zur Produktkonfiguration durch den Kunden, ist die Anzahl der möglichen Fahrzeugvarianten, wie schon mehrfach dargestellt, unüberschaubar groß. Das heißt, die Speicherung jeweils einer kompletten Stückliste für alle möglicherweise vorkommenden Produktvarianten ist nicht praktikabel. Stattdessen können aber Stücklisten temporär für einen Auftrag, also eine bestimmte Produktvariante, zusammengestellt und wieder gelöscht werden, wenn diese Auftragsstückliste nicht länger benötigt wird.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CBOStructure			PRODUCTSTRUCTURE	
COrderModifier 1..1 →	.OrderModifier	D.9	.ORDERMODIFIER	E.53
CProducer ⁴²			SUPPLIER	
^{CBOM} 1..n →	.BOM	D.8	.BILLOFMATERIALSUPPLIER	E.71
CBOMEntry			BILLOFMATERIALRECORD	
CProductPattern 1..1 →	.Pattern	D.8	.PATTERN	E.9
CComponentReference 1..n →	.Components	D.8	.PARTS	E.9
CComponentReference			PARTREFERENCE	
CComponent 1..1 →	.Component	D.8	.PART	E.40
<i>real</i>	.Weight	D.8	.WEIGHT	E.40
CComponent			PART	
CComponentReference 1..n →	.SubComponents	D.8	.SUBPARTS	E.39
<i>string</i>	.ItemNumber	D.8	.ITEMNUMBER	E.39
CSupplier 1..1 →	.Supplier	D.8	.PARTSUPPLIER	E.39
CBOClass			PRODUCTCLASS	
CProperty 1..n →	.EnforcedProperties	D.8	.ENFORCEDPROPERTIES	E.48

Tabelle 6.7.: Zentrale Parameter Stücklisten

Im vorgestellten Ansatz finden sog. *Baukastenstücklisten* Verwendung (vgl. [WYS02, B3-53], [Ble79, S. 53], [BG97a, F 33]). Die Baukastenstückliste enthält Informationen über eine Stufe der Zerlegung einer Komponente in seine Bestandteile. Die Zerlegung dieser (Sub-)Komponenten wiederum wird in einer weiteren Baukastenstückliste aufgeführt, sofern für den jeweiligen Einsatzzweck erforderlich. Da wie gesagt nicht für jede denkbare Produktvariante eine Baukastenstückliste vorgehalten werden kann, muss diese dynamisch bestimmt werden, wenn alle erforderlichen Informationen vorliegen. Neben der Konfiguration des Auftrags in Bezug auf kundenwählbare Eigenschaften, ist auch das Datum, an dem das Fahrzeug gebaut werden soll, eine Information, die maßgeblich über die zu verbauenden Teile bestimmt. Anhand

⁴¹Eine *Komponente* soll als Teil eines Produktes aufgefasst werden. Der Begriff *Teil* wird als Synonym hierzu verwendet. Während die vom Kunden wählbaren Eigenschaften lediglich die gewünschten Funktionen eines Produktes festlegen, bestimmt die Zerlegung in Teile bzw. Komponenten die physikalische Ausführung des Produktes.

⁴²Im UML-Diagramm D.28 ist die Stückliste in der Basisklasse von CSupplier und CPlant definiert – CProducer. Die Stückliste am Werk ist bislang aber nur zur Nutzung vorgesehen, daher wird im Text von einer Definition der Stücklisten an der Klasse SUPPLIER gesprochen und damit eine Definition an der Klasse CSupplier impliziert.

des Datums wird bestimmt, welche Version eines Teils aus einer (etwaigen) Liste von verschiedenen Möglichkeiten gewählt wird.

Die Stücklistenauflösung kommt in Simulationsmodellen von Auftragsabwicklungsprozessen sinnvollerweise dann zum Einsatz, wenn Abhängigkeiten zwischen Auftragseinplanung und Kapazitätssituation untersucht werden sollen. In vielen Fällen ist dabei die Verbindung zwischen Eigenschaften und Engpassressourcen so einfach, dass eine 1:1-Verbindung diese Relation gut abbildet. Ist z.B. die Kapazität für ein bestimmtes Aggregat in besonderer Weise limitiert, kann dieser Zusammenhang dadurch abgebildet werden, dass immer dann, wenn ein solches Aggregat verbaut werden soll, die verfügbare Kapazität zum fraglichen Zeitpunkt mittels eines »Zählers« verwaltet wird. Das heißt, der Weg über eine dann triviale Stücklistenauflösung sollte vermieden werden können.

Um in einem solchen Fall ohne Parametrisierung von Stücklisten auskommen zu können, verwalten Zulieferer vom Typ SUPPLIERBTO Kapazitäten auf Basis von Eigenschaften. Ist eine Stücklistenauflösung gewünscht bzw. erforderlich, können (Sub-) Zulieferer an einem Zulieferer vom Typ SUPPLIER spezifiziert werden, die Kapazitäten auf der Basis von Komponenten/Teilen verwalten. Dazu ist an diesem Zulieferer eine Stückliste anzugeben. Um eine Gruppe von Eigenschaften mit einer Kapazität zu verknüpfen, kann eine (Hilfs-)Eigenschaft eingeführt werden, die genau dann zu einem Auftrag hinzugefügt wird⁴³, wenn der Auftrag einer definierten Menge von Bedingungen genügt.

Am Attribut .BILLOFMATERIALSUPPLIER der Klasse SUPPLIER⁴⁴ werden die relevanten Stücklisteneinträge für diesen Zulieferer vermerkt, die in einem Simulationsmodell genutzt werden sollen. Die hier definierten Objekte vom Typ BILLOFMATERIALRECORD repräsentieren eine Stufe der Zerlegung eines Produktes in seine Komponenten. Durch eine Differenzierung in der an einen Abnehmer gelieferten Produktvariante können auch abnehmerspezifische Stücklisten zum Einsatz kommen. Das heißt, liefert ein Zulieferer an verschiedene Werke eines oder mehrerer Automobilhersteller, können unterschiedliche Stücklisten zum Einsatz kommen.

Um für einen konkreten Auftrag alle benötigten Objekte dieses Typs zu identifizieren, wird das durch das Attribut .PATTERN referenzierte Objekt des Typs PRODUCTPATTERN genutzt. Wenn alle hier spezifizierten Bedingungen erfüllt sind⁴⁵, werden alle Objekte vom Typ PARTREFERENCE, welche am Attribut .PARTS referenziert werden, in eine temporäre Liste aufgenommen. Durch weitere Berechnungsschritte kann nun abgeleitet werden, welche Teile benötigt werden, um den Auftrag zu fertigen.

Hierzu werden alle Objekte vom Typ PARTREFERENCE der temporären Liste durchlaufen. Das Attribut .PART dieser Objekte vom Typ PART verweist auf eine Subkomponente, also wiederum eine Stufe der Zerlegung. Das Attribut .WEIGHT bestimmt die benötigte Anzahl Teile. In weiteren Stücklisten anderer Zulieferer können die so ermittelten Teile weiter zerlegt werden.

⁴³Parametrisiert durch Objekte vom Typ ORDERMODIFIER am Attribut .ORDERMODIFIER der Produktstruktur.

⁴⁴Und damit in den von SUPPLIER abgeleiteten Klassen SUPPLIERBTO und SUPPLIERBTS.

⁴⁵Das kann z.B. ein Teil (Attribut .PARTS) sein (welches vom Zulieferer produziert werden soll), zusammen mit einem Intervall, in dem dieser Stücklisteneintrag gültig ist (Attribute .VALIDFROM, .VALIDTO).

Die Anzahl der Stücklisteneinträge, aus denen die Auftragsstückliste zusammengestellt wird, kann mehrere 1.000 betragen. Um dennoch eine zeiteffiziente Stücklistenauflösung zu gewährleisten, wird ein Teil dieser Baukastenstücklisten bereits einer Produktbeschreibung zugeordnet, soweit möglich. Das Attribut `.ENFORCEDPROPERTIES` der Klasse `PRODUCTCLASS` erlaubt es, Eigenschaften zu spezifizieren, die durch die Produktklasse bereits festgelegt werden und somit vom Kunden nicht mehr beeinflusst werden können, wenn die Produktklasse einmal bestimmt ist. Liegen also z.B. Motor und Getriebe neben der Produktklasse bereits fest und alle Eigenschaften, die durch Konstruktionsregeln in den Auftrag »gezwungen« werden, liegen ebenfalls vor, können die Stücklisteneinträge, die bereits auf Basis dieses »Rumpfauftrages« ausgewählt werden können, bereits gespeichert werden⁴⁶. Da diese Bearbeitungsschritte statisch, d.h. ohne dass konkrete Aufträge vorliegen, durchgeführt werden können, ist eine u.U. starke Reduktion⁴⁷ der Anzahl der Bearbeitungsschritte zur Bestimmung der Auftragsstückliste möglich.

Sowohl die Abbildung des Prozesses der Bestimmung der Auftragsstückliste bei sehr variantenreichen Produkten als auch die Möglichkeit zur Definition und Anwendung komplexer Baukastenstücklisten, erschließt zum einen die Fähigkeit der Bestimmung der Teile-Bedarfe eines Automobilherstellers und zum anderen aber auch die Modellierung von Teilen der Systemlast des Lieferantennetzwerkes und bildet so eine Grundlage für die Beantwortung von Fragestellungen, die eine detaillierte Abbildung des Netzwerkes von Kapazitäten eines Automobilherstellers erfordern.

6.2.8. Zusammenfassung

Die Beschreibung von *Produkten* im oben dargestellten Schema ordnet diesen *Eigenschaften* zu, die in den Produkten in unterschiedlichen Anteilen zu finden sein können. Diese Eigenschaften können nicht beliebig kombiniert werden, sondern unterliegen in ihrer Kombinierbarkeit einer Reihe von Konfigurationsregeln. Eigenschaften sind Bestandteile der Sicht, die Kunden auf ein Fahrzeug haben. Typische Eigenschaften sind z.B. *Navigationssystem*, *Abstandsradar* und *Xenonlicht*.

Zusätzlich zu den Eigenschaften, die ein Fahrzeug haben kann, können *Teile* definiert werden. Teile repräsentieren diejenigen Komponenten eines Fahrzeugs, die für eine Fertigung zur Verfügung stehen müssen und z.B. von einem Lieferanten angeliefert werden. Das Modul »Lastgenerator« erzeugt Produkte inklusive seiner Eigenschaften so, dass diese in Anzahl und Eigenschaftenverteilung den Vorgaben entsprechen, die in einem Modell gemacht werden.

6.3. Märkte und Händler

Die Kunden auf den jeweiligen Märkten bilden in gewisser Hinsicht die »Systemlast« für den Auftragsabwicklungsprozess. Die Anzahl und Konfiguration der tatsächlich nachgefragten Fahrzeugmodelle beeinflusst nachhaltig die Erlössituation eines Automobilherstellers. Die Eigenschaften des Auftragsabwicklungsprozesses bestimmen

⁴⁶Diese Vorverarbeitung der Daten ist allerdings nur dann möglich, wenn nicht mehrere Zulieferer und damit mehrere verschiedene Stücklisten für eine Berechnung betrachtet werden müssen.

⁴⁷Die tatsächliche Reduktion hängt von der konkreten Definition der Produktklassen ab.

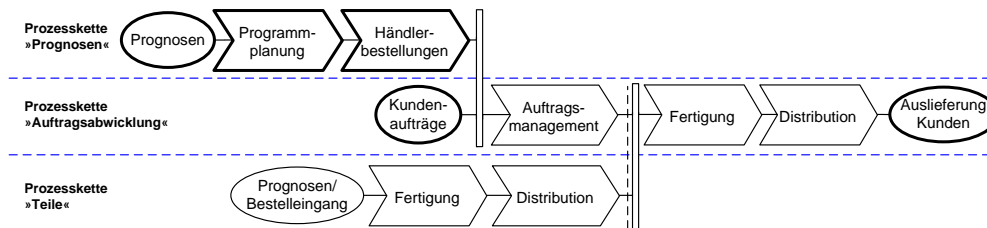


Abbildung 6.6.: Modellelemente Märkte und Händler

dabei entscheidend, inwieweit und zu welchen Kosten der Hersteller auf Abweichungen zwischen prognostizierter und tatsächlicher Nachfrage reagieren kann.

Aufgrund der hohen Bedeutung des Zusammenspiels zwischen Prognosen und tatsächlicher Nachfrage für die Bewertung von Auftragsabwicklungsprozessen ist es nicht nur erforderlich, Aufträge in realistischen Anzahlen und Konfigurationen zu erzeugen (siehe oben), sondern auch die Bestellprozesse der Händler, die Auftragsverbuchung und weitere Bearbeitung der Aufträge durch den Hersteller abzubilden. Im folgenden Abschnitt wird eine über die Modellelemente »Märkte« und »Händler« hinaus greifende Darstellung der wichtigsten Modellparameter in diesem Kontext versucht.

6.3.1. Einplanung von Aufträgen

Für die Bestimmung eines geeigneten Fertigungszeitpunktes für einen gegebenen Auftrag können zahlreiche Faktoren in Betracht gezogen werden. Aufträge können so eingeplant werden, dass die Kundenanforderungen möglichst gut befriedigt werden (Liefertermin), es können aber auch Auslastungskriterien der Werke oder das Profitstreben eines Händlers eine Rolle spielen. Diese unterschiedlichen Zielsysteme werden häufig auf die Terminfindung sequentiell Einfluss nehmen: Die Verhandlung zwischen Kunde und Händler beeinflusst die Nennung eines gewünschten Liefertermins in Richtung Hersteller, d.h. dieser genannte Wunschtermin ist bereits ein Kompromiss zwischen Kunden- und Händlerinteresse. Im nächsten Schritt werden Kriterien des Herstellers angelegt, um einen Fertigungstermin festzulegen.

Eine Alternative zu dieser sequentiellen Vorgehensweise könnte es sein, einen für den Hersteller günstigen Fertigungstermin durch teilweise Weitergabe der Kostenvorteile an den Händler bzw. den Kunden, eine für alle Beteiligten vorteilhafte Situation zu schaffen. Der im Rahmen dieser Arbeit vorgestellte Ansatz soll beide Ansätze der Terminbestimmung unterstützen, um sowohl etablierte Prozesse als auch zukünftige Strukturen abbilden zu können.

Um dies zu erreichen, wurde eine Reihe von Parametern bestimmt (siehe Abbildung 6.7 und Tabelle 6.8), die wichtige Eigenschaften der Auftragseinplanung beeinflussen und es somit durch Variation dieser Parameter erlauben, unterschiedliche Abläufe in der Planung abzubilden.

- ▷ **Bestellverhalten Kunde.** Kunden bestellen Fahrzeuge mit einem gewissen zeitlichen Vorlauf – einige wünschen ein Fahrzeug sofort zu bekommen, andere äußern einen in der Zukunft liegenden Wunschliefertermin. Dieses Kundenverhalten wird

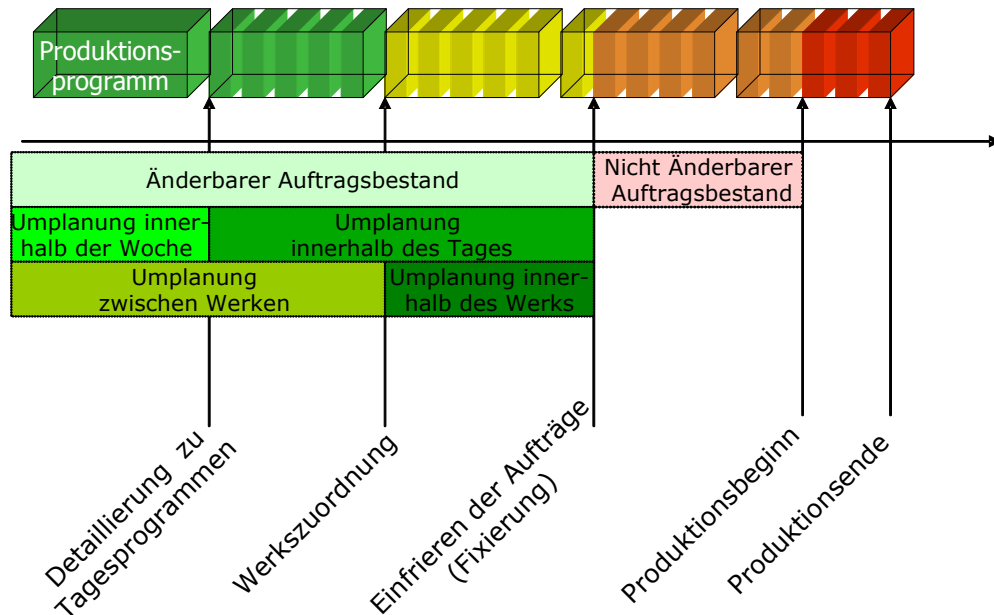


Abbildung 6.7.: Statuspunkte Auftragseinplanung

durch eine Wahrscheinlichkeitsverteilung⁴⁸ modelliert (Attribut `.CUSTOMERORDERBANDWIDTH`), die abbildet, mit welcher Wahrscheinlichkeit ein Kunde ein Fahrzeug n -Tage in der Zukunft wünscht (siehe Kapitel 6.3.2).

- ▷ **Bestellverhalten Händler.** In welchen Abständen spezifiziert der Händler Bestellungen (optional), um einer etwaigen Abnahmeverpflichtung nachzukommen (Attribut `.ORDERS`).
- ▷ **Tagesprogrammabbildung.** Aufträge können zunächst so in einen »Pool« eingeplant werden, dass Kapazitäten zunächst nur auf (z.B.) Wochenbasis überprüft werden. Eine Verteilung der Aufträge auf Tage wird unter Beachtung der Restriktionen⁴⁹ für diese Tage durchgeführt und kann zu einer Veränderung der Auftragssequenz führen⁵⁰.
- ▷ **Werkzuordnung.** Verbunden mit der Übergabe der Aufträge an das Werk⁵¹ ist ei-

⁴⁸Eine Wahrscheinlichkeitsverteilung gibt an, mit welcher Wahrscheinlichkeit eine Zufallsgröße X bestimmte Werte annimmt (vgl. [Czi89, A 128]).

⁴⁹Verfügbare Kapazitäten für Fahrzeuge und Eigenschaften. Die Reihenfolge innerhalb eines Tages wird hier nicht betrachtet.

⁵⁰Abbildung durch geeignete (wöchentlich/täglich) Kapazitätskonten bei Lieferanten; siehe Tabelle E.71.

⁵¹Wenn keine größere Störung eintritt (vgl. Kapitel 6.5.3), werden die Aufträge nicht mehr zwischen den Werken verschoben.

ne Beschränkung der Auftragsanpassbarkeit auf die Eigenschaften, die in diesem Werk verbaut werden können, sowie eine Fixierung der Volumina. Das heißt, die Aufträge können zwar noch (begrenzt) verändert, aber neue Aufträge können in diesem Bereich nicht mehr platziert werden (Attribute `.PLANTASSIGNMENTDATES` und `.DELAYPLANNING`).

- ▷ **Einfrierzeitpunkt Werk.** Vollständige Fixierung der Aufträge. Veränderung der Aufträge sind nicht mehr zulässig (Attribut `.HANDOVERDATES`).

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CCustomerType			CUSTOMERTYPE	
CProbabilityDistribution 1..1 →	.CustOrderDistribution	D.22	.CUSTOMERORDERBANDWIDTH	E.16
CPlanning			PLANNING	
CSchedule 1..1 →	.SeqF1	D.34	.FIRSTFORECAST	E.41
CSchedule 1..1 →	.SeqF2	D.34	.SECONDFORECAST	E.41
CSchedule 1..1 →	.PlantAssignmentDates	D.34	.PLANTASSIGNMENTDATES	E.41
CSchedule 1..1 →	.HandOverDates	D.34	.HANDOVERDATES	E.41
CPlant			PLANT	
COrderBank 1..1 →	.OBK	D.29	—	—
COrderBank			PLANT	
CResourceProductionUnit 1..1 →	.CapacityActual	D.33	.CAPACITYACTUAL	E.42
CResourceProductionUnit 1..1 →	.CapacityPlanned	D.33	.CAPACITYPLANNED	E.42
CSupplier			SUPPLIER	
CResource 1..1 →	.Capacity	D.35	Resource	E.71
CPlanningDealer			DEALERPLANNING	
CSchedule 1..1 →	.Seq0	D.23	.ORDERS	E.19
CProbabilityDistribution 1..1 →	.DelayPlanning	D.23	.DELAYPLANNING	E.19
CProbabilityDistribution 1..1 →	.DelayPlant	D.23	.DELAYPLANT	E.19

Tabelle 6.8.: Zentrale Parameter Auftragsplanung

Neben den oben genannten Parametern spielen weiterhin die kapazitiven Rahmenbedingungen für die Auftragsplanung eine Rolle, die wiederum über Parameter eingestellt werden können⁵².

Die Kapazität der Werke kann in nahezu beliebiger Granularität eingestellt werden und gibt an, wie viele Aufträge in einem Zeitraum eingeplant werden können⁵³. Ist für die Einbuchung eines Auftrages Kapazität verfügbar, wird zudem eine Überprüfung ausreichender Zuliefererkapazität vorgenommen. Dies wird für alle Ausstattungsmerkmale durchgeführt, für die Lieferantenkapazitäten parametrisiert sind.

In der Regel wird eine Einplanung eines Auftrages an mehr als einem Tag möglich sein, ohne Kapazitätsrestriktionen zu verletzen; unter Beachtung verschiedener

⁵²Die Kapazitätswerte der UML-Klasse `CPlant` werden in der Klasse `COrderBank` gebündelt. In der Tabelle wird die entsprechende Aggregationsstruktur durch einen vorangestellten Pfeil (→) gekennzeichnet (siehe Diagramme D.29 und D.33).

⁵³Die für die Fertigung tatsächlich zur Verfügung stehende Kapazität kann von der Plankapazität abweichend spezifiziert werden.

Auswahlkriterien ist nun ein Zeitpunkt für die Einplanung zu bestimmen. Zwei wesentliche Kriterien für die Terminfindung sind zum einen das Interesse des Automobilherstellers, den Auftrag möglichst kostengünstig zu fertigen und zum anderen das Interesse des Kunden, das Fahrzeug zum zugesagten Termin zu bekommen. Es ist nahe liegend, dass die Gewichtung dieser Kriterien zwischen den Automobilherstellern und sogar zwischen den verschiedenen Baureihen eines Herstellers schwanken kann⁵⁴. Um die Abbildung verschiedener Strategien in diesem Bereich zu ermöglichen, wird im vorgestellten Ansatz eine *Gütefunktion* verwendet, die aus einer Reihe von möglichen Fertigungszeitpunkten und Fertigungsstandorten (Werken) eine Auswahl trifft (siehe Kapitel 6.5.2).

Ist ein Fertigungszeitpunkt festgelegt, kann hieraus ein Anliefertermin für die Übergabe des Fahrzeugs an den Händler und potenziell den Kunden berechnet werden. Die berechneten Termine für die Händler- bzw. Kundenbelieferung bilden die Bezugsgrößen für die Messung einer Reihe von Kennzahlen (z.B. »Liefertreue (Kunde)«). Detailinformationen zur Berechnung dieser Termine werden in Kapitel 6.7.3 gegeben.

Bis zum Fertigungsbeginn können sich Verschiebungen der Auftragsreihenfolge ergeben, die durch Veränderungen der Kapazitätssituation gegenüber einer ursprünglichen Planung verursacht werden. Diese Einflüsse im Realsystem sind sehr vielfältig und können mit stochastischen Mitteln abgebildet werden. Hierzu kann die Auftragsreihenfolge an zwei Punkten verändert werden: Zum einen bei der »Werkszuordnung« und zum anderen bei der »Tagesprogrammabildung«.

Nach der »Verwirbelung« der Aufträge wird vor der Festlegung eines neuen Fertigungstermins nochmals die etwaige Verletzung von Kapazitäts- oder Fertigungsrestriktionen geprüft – die neue Fertigungsreihenfolge verletzt also, wie auch die vorherige Sequenz, keine der parametrisierten Obergrenzen. Zur Bildung der Sequenz am letztmalig geplanten Fertigungsbeginn werden die verfügbaren Kapazitäten der Lieferanten und des Werkes nochmals geprüft. Die am Tag der Tagesprogrammabildung verfügbaren Kapazitäten können abweichend von den Kapazitäten zum Zeitpunkt der Einplanung eingestellt werden, um ebensolche Abweichungen im Realsystem abbilden zu können. Neben den im Vorfeld eingestellten Kapazitätsabweichungen können die Kapazitäten auch durch stochastische Einflüsse verändert werden (siehe Kapitel 6.4).

6.3.2. Kundenverhalten

In den von den Automobilherstellern belieferten Märkten werden teilweise sehr unterschiedliche Vertriebsstrategien genutzt. Dies reflektiert zum einen geographische Aspekte – die Entfernung zwischen den Fahrzeugen, bauenden Werken und dem Markt, aber auch das Verhalten der Kunden. Während im deutschen Markt Fahrzeuge zumindest der inländischen Hersteller zum größten Teil Build-To-Order gefertigt werden, wählen amerikanische Kunden ihre Fahrzeuge in der Mehrzahl aus bereits gefertigten Fahrzeugen aus (siehe Kapitel 2.2.2).

Die Neigung der Kunden, ein Lagerfahrzeug zu akzeptieren, beeinflusst in erheblicher Weise einige der Kennzahlen, die benötigt werden, um einen Auftragsabwicklungsprozess zu bewerten. Eine dieser Kennzahlen ist der Lagerbestand gefertigter

⁵⁴Für ein Oberklassefahrzeug könnte z.B. der Kundenwunsch höher als für ein »Volumenmodell« gewichtet werden.

Fahrzeuge. Ein Auftragsabwicklungsprozess, der nur restriktiv Anpassungen der Aufträge an Kundenwünsche erlaubt, wird in der Tendenz dazu führen, dass relativ wenige händlerspezifizierte Aufträge an einen Kundenwunsch angepasst werden und somit in der vom Händler gewählten Konfiguration in das Lager eingehen. Im deutschen Markt sind solche Fahrzeuge nur mit erheblichen Rabatten an Kunden vermittelbar; die Auswirkungen auf den Deckungsbeitrag sind offensichtlich.

Um Bestandskennzahlen realistisch abbilden zu können, ist es also erforderlich, die »Neigung« der Kunden abbilden zu können, Fahrzeuge zu akzeptieren, die ihren Wünschen nicht oder zumindest nicht vollständig entsprechen. Um dies zu erreichen, wurden folgende Parameter zur Abbildung des Kundenverhaltens entwickelt:

▷ **Wunschterminverteilung.** Die Wunschterminverteilung spezifiziert die Anzahl der Tage, die nach dem Kundenwunsch zwischen der Beauftragung und der Auslieferung eines Fahrzeuges liegen sollen. Ein »eiliger« Kunden wünscht sofort ein Fahrzeug (Null Tage zwischen Auftrag und Lieferung bzw. Übergabe), ein anderer Kunde erwartet ein neues Fahrzeug nach Ablauf eines Leasingvertrages in drei Monaten. Durch eine Wahrscheinlichkeitsverteilung kann eine beliebige Verteilung dieser Wunschlieferzeiten der Kunden parametrisiert werden.

▷ **Akzeptanz Abweichung.** Die Neigung eines Kunden, eine Abweichung von der Wunschkonfiguration zu akzeptieren, ist stark von der konkreten Ausprägung dieser Abweichung abhängig. Ist beispielsweise ein Lagerfahrzeug zusätzlich zu den gewünschten Eigenschaften mit einem Regensensor ausgestattet, ist die Wahrscheinlichkeit vermutlich hoch, dass der Kunde das Fahrzeug akzeptiert, insbesondere, wenn ein Rabatt auf dieses Ausstattungsmerkmal gewährt wird.

Ist das Fahrzeug hingegen nicht mit dem gewünschten kleinsten Dieselmotor, sondern mit dem größten lieferbaren Benzinmotor ausgestattet, wird der Kunde kaum geneigt sein, das angebotene Fahrzeug zu erwerben. Der Parameter `.AKZEPTANZ ABWEICHUNG` erlaubt es, für alle Eigenschaftenfamilien für Paare von Eigenschaften Werte zu spezifizieren, die die Abweichung vom Kundenwunsch angeben. Obiges Beispiel könnte wie folgt abgebildet werden: Abweichung [*kein_Regensensor*, *Regensensor*] \rightarrow 0,05, Abweichung [*Diesel_75KW*, *Otto_220KW*] \rightarrow 1,0. Erreicht die Summe der Abweichungen 1,0, lehnt der Kunde das Fahrzeug ab.

▷ **Akzeptanz Lieferzeit.** Neben der Abweichung der Ausstattung geht die prognostizierte Lieferzeit eines Fahrzeugs in die Entscheidung eines Kunden ein. Ein Kunde, der dringend ein Fahrzeug zu dem Zeitpunkt benötigt, an dem dieser beim Händler erscheint, wird eher eine abweichende Ausstattung akzeptieren, als ein Kunde, der bereit ist, fünf Monate zu warten. Der Parameter `.AKZEPTANZ LIEFERZEIT` erlaubt es, die Abweichung der Ausstattung zur erwarteten Lieferzeit ins Verhältnis zu setzen.

▷ **Akzeptanz Lagerfahrzeuge.** Je länger ein Fahrzeug in einem Lager steht, desto größer wird die Neigung eines Händlers sein, einem Kunden einen Rabatt einzuräumen. Gelagerte Neufahrzeuge sind nach einer gewissen Standzeit im Lager Wartungsmaßnahmen zu unterziehen, wie z.B. das Laden der Batterie und Bewegungsfahrten, die Kosten verursachen.

Übersteigt die Standzeit einen von der jeweiligen Gesetzgebung abhängigen Wert, kann das Fahrzeug nicht mehr als »Neu« verkauft werden. Um den mit der Standzeit steigenden Rabatt und die aufgrund des Preises erhöhte Attraktivität eines solchen Fahrzeugs für einen Kunden abzubilden, dient der Parameter .AKZEPTANZ LAGERFAHRZEUGE. Der eingeräumte Rabatt wird der Ergebnis-Kategorie »Vertriebskosten« zugeordnet (siehe Tabelle F.1).

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CCustomerType		CUSTOMERTYPE	
CProbabilityDistribution 1..1 →	.CustOrderDistribution	D.22	.CUSTOMERORDERBANDWIDTH	E.16
CPropertyDeviation 1..1 →	.PropertyDeviation	D.22	.DEVIATIONACCEPTANCE	E.16
CAssocDataIVInt2Float 1..1 →	.DeliveryTimeAcceptance	D.22	.DELIVERYTIMEACCEPTANCE	E.16
CAssocDataIVInt2Float 1..1 →	.StockTimeAcceptance	D.22	.STOCKTIMEACCEPTANCE	E.16

Tabelle 6.9.: Zentrale Parameter Kundenverhalten

6.3.3. Zusammenfassung

Objekte der Klassen »Märkte«, »Händler« und »Kunden« bilden die Nachfragesituation bzw. die Systemlast für einen Auftragsabwicklungsprozess ab. Märkte bilden eine »Klammer« um Händler, die einer geographischen Region zugeordnet werden können bzw. in anderen für die Beantwortung einer Fragestellung relevanten Eigenschaften gruppiert werden sollen. Händler sind sowohl Quellen für Prognosen, die zu definierten Zeitpunkten abgegeben werden, wie auch für Aufträge, die gemäß einer parametrisierten Verteilung an den Automobilhersteller übergeben werden. Neben der Rolle als Quelle von Aufträgen dienen Händler auch als Senken für gefertigte Fahrzeuge, die nach Ankunft beim Händler einem Kunden (sofern vorhanden) übergeben werden.

6.4. Ressourcen

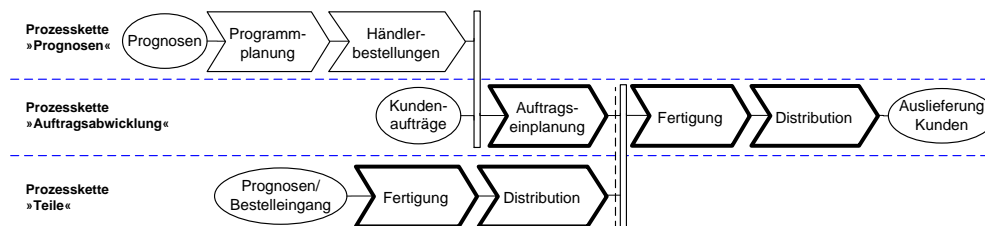


Abbildung 6.8.: Einsatz von Ressourcenobjekten

Die Modellierung von Kapazitäten spielt für die Beantwortung von vielen Fragestellungen eine Schlüsselrolle. Die Möglichkeit zur dynamischen Berechnung der Belastung von Kapazitäten ist einer der wesentlichen Vorteile des vorgeschlagenen Ansatzes; daher kommt der Abbildung dieser Kapazitäten eine Schlüsselrolle zu⁵⁵.

⁵⁵Der Prozess »Programmplanung« verwendet Informationen über Ressourcen, da aber an der Klasse

Aus den Anforderungen an die Modellelemente, die als »Nutzer« eines Ressourcenkonzeptes in Frage kommen (z.B. Werke, Zulieferer, Distributionskanäle), und den »direkten« Anforderungen an Ressourcen aus Kapitel 4 ergeben sich zusammengefasst die folgenden fünf Punkte:

1. Die Zuteilung von Kapazitäten an Abnehmer auf der Basis von Prognosen ist erforderlich.
2. Die Anpassung der Kapazitäten auf Basis von Prognosen sollte steuerbar sein.
3. Es muss möglich sein, eine Kapazität zu »buchen«, d.h. für eine spätere Verwendung zu reservieren.
4. Es ist erforderlich, ein Konzept zur Abbildung von ungeplanten Kapazitätsschwankungen (Störungen) umzusetzen.
5. Verschiedene Konstellationen der Ressourcennutzung durch die Abnehmer müssen abgebildet werden können (Teilung einer Ressource durch n Abnehmer, mehrere Ressourcen sind an einen Abnehmer gebunden etc.).

Die Anforderungen eins und zwei können beide als »Buchung« aufgefasst werden. Im ersten Fall wird auf Basis einer langfristigen Prognose⁵⁶ eine Reservierung von Kapazitäten angefordert. Ist der Nutzer einer Ressource z.B. ein Werk, liegen für diese langfristige Anfrage noch keine Aufträge vor, aber durch Addition aller von den Abnehmern einer Ressource prognostizierten Werte kann eine Prognose für die Ressource erstellt werden, auf deren Basis eine Kapazitätsanpassung durchgeführt werden kann. Diese Prognose kann zudem genutzt werden, um die Kapazitäten zwischen den Abnehmern zu verteilen, um eine »faire« Aufteilung der Kapazität zwischen den Abnehmern zu erreichen.

Die Möglichkeiten eines Werkes oder eines Zulieferers zur kurzfristigen starken Erhöhung der Kapazität auf Basis einer eintreffenden Prognose sind wahrscheinlich im Realsystem eher gering. Eine kurzfristige Erhöhung der LKW-Transportkapazität eines Distributionskanals erscheint dagegen eher umsetzbar. Um diese Fähigkeit zur Anpassung der Kapazität einer Ressource an Prognosen abzubilden, dienen die Parameter `.FLEXBAND_P` und `.FLEXBAND_M`. Hier kann sowohl für eine Anpassung nach oben als auch für eine Anpassung nach unten eine Bandbreite spezifiziert werden, innerhalb der eine Anpassung möglich ist. Diese Bandbreite kann abhängig von der zeitlichen Entfernung der gewünschten Kapazitätsanpassung gestaffelt werden. Eine mögliche Parametrisierung ist z.B. eine Fähigkeit zur Kapazitätserhöhung um 20 % bei 3 Monaten zeitlichem Vorlauf, um 7 % bei zwei Monaten und um 2 % bei kurzfristigeren Anfragen.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CResourceAssocData			ASSOCDATRESOURCE	
CAssocDataFloat 1..1 →	.Capacity	D.26	.CAPACITY	E.42

PLANNING keine Ressourcen definiert sind, wird dieser Prozess in Abbildung 6.8 nicht hervorgehoben.

⁵⁶Eine langfristige Prognose kann z.B. zwei Jahre vor dem Bezugszeitraum der Prognose abgegeben werden. Für die Gesamtfahrzeugabsätze in einem Modell (Klasse SALES) kann diese am Attribut `.LONGTERMEST` parametrisiert werden.

CAssocDataIVInt2Float 1..1 →	.FlexBand_P	D.25	.FLEXBAND_P	E.6
CAssocDataIVInt2Float 1..1 →	.FlexBand_M	D.25	.FLEXBAND_M	E.6
CFailure 1..n →	.Failures	D.24	.FAULTS	E.6
bool	.SimpleResource	D.24	.SIMPLERESOURCE	E.6

Tabelle 6.10.: Zentrale Parameter Ressourcen

Basis für die Anpassung von Kapazitäten ist die Sammlung von Prognosen aller Abnehmer. Die nach Anpassung der Grundkapazität, die durch das Attribut `.CAPACITY` spezifiziert wird, durchgeführte Zuweisung von Kapazitäten an die einzelnen Abnehmer erfolgt proportional zur Höhe der Prognosen, d.h. eine hohe Prognose führt zu einer hohen Zuteilung. Wird eine Zuteilung von Kapazitäten auf Basis von Prognosen nicht gewünscht, die Kapazität im Modell ist dann für alle Abnehmer genau die am Attribut `.CAPACITY` spezifizierte, kann dazu das Attribut `.SIMPLERESOURCE` auf den Wert »Ja« gesetzt werden.

Um die geforderten Verknüpfungen von Ressourcen abbilden zu können, werden zwei Klassen definiert: Summen- (`SUMRESOURCE`) und Aggregatressource (`AGGREGATERESOURCE`). Die Summenressource »bündelt« (addiert) die Kapazitäten der Ressourcen, die am Attribut »SubResources« definiert werden, die Aggregatressource »überwacht« die Verfügbarkeit von allen Ressourcen, die am Attribut »SubResources« definiert sind und liefert stets die kleinste der verfügbaren Kapazitäten der angehängten Ressourcen. Die nahezu beliebige Kombinierbarkeit der Ressourcen zu Ressourcennetzwerken erlaubt somit die Abbildung auch komplexer Zusammenhänge.

Zu einem späteren Zeitpunkt, also nach der Zuteilung von Kapazitäten auf Basis der Prognosen, können Kapazitäten für eine spätere Verwendung gebucht werden. Wird z.B. ein Auftrag in ein Werk eingeplant, reduziert sich damit die verfügbare Kapazität für weitere Aufträge. Diese Verwaltung bereits vergebener Kapazitäten wird durch die Ressourcen geleistet, wobei die im Vorfeld geleistete Zuordnung von Kontingenten an die Abnehmer der Ressource stets beachtet wird.

Da die prognostizierte Kapazität, z.B. im Fall von Störungen, von der tatsächlichen Kapazität abweichen kann, ist es möglich, am Attribut `.FAULTS` eine Liste von solchen Störungen zu spezifizieren. Die Parameter dieser Störungen erlauben eine Einschränkung der verfügbaren Kapazität um einen Faktor, sowie die Spezifizierung einer »Vorwarnzeit«. So kann z.B. eine Kapazitätsreduktion um 50 % mehrere Tage im Voraus angekündigt werden, was den Abnehmern der Ressource unter Umständen andere Reaktionsformen erlaubt, als eine Störungsmeldung ohne Vorlauf. Eine Störung wird zum Zeitpunkt der Vorwarnung, dem eigentlichen Eintreffen und dem Ende der Störung allen Abnehmern »gemeldet«. Die Reduktion der verfügbaren Kapazität wird durch die Implementierung der Ressource berechnet, darüber hinausgehende Reaktionen auf die Störungen sind durch die Implementierung der Abnehmer zu leisten (siehe z.B. Störungsregeln im Werk).

Ressourcen bilden grundlegende Funktionen zur Verwaltung von Kapazitäten ab, die an verschiedenen Stellen in einem Modell eingesetzt werden können (siehe Abbildung 6.8). Darüber hinaus gehende Funktionalitäten, wie z.B. die platzorientierte Verwaltung von Kapazitäten oder Funktionen zur vorausschauenden Produktion von Gütern, werden in den Klassen implementiert, die diese Funktionalitäten exklusiv nutzen (hier Werke und Zulieferer). Ressourcen können miteinander kombiniert werden

(Aggregat- und Summenressource) und damit auch komplexe Kapazitätsnetzwerke abbilden.

6.5. Werke

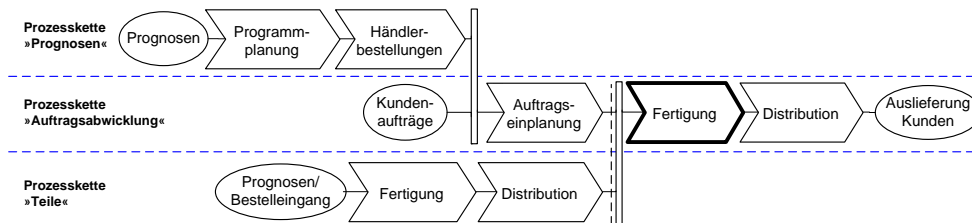


Abbildung 6.9.: Modellelement »Werk«

Nach den Kosten für den Einkauf der für die Fertigung eines Fahrzeuges benötigten Materialien haben die Produktionskosten den größten Anteil an den Gesamtkosten eines Fahrzeuges⁵⁷. Eine hohe Auslastung der Produktionseinrichtungen ist daher ein vordringliches Ziel bei der Gestaltung von Auftragsabwicklungsprozessen und führt zu den in Kapitel 4.6 genannten Anforderungen an die Abbildung von Werken.

6.5.1. Kapazitäten

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CPlant			PLANT	
CResourceProductionUnit 1..1 →	.CapacityActual	D.33	.PRODUCTIONUNITRESOURCE	E.42
CResourceProductionUnit 1..1 →	.CapacityPlanned	D.33	.CAPACITYPLANNED	E.42

Tabelle 6.11.: Zentrale Parameter Werkskapazität

Die Kapazitäten eines Werkes werden zum einen durch die Kapazitäten der Fertigungseinrichtungen und des Personals beschränkt und zum anderen durch etwaige knappe Ressourcen eines oder mehrerer Lieferanten. Bei der Einplanung von Aufträgen werden all diese Faktoren berücksichtigt und überprüft. Um eine an Schichten ausgerichtete Personalplanung mit geringem Aufwand abbilden zu können, werden die Kapazitäten von Werken durch Ressourcen vom Typ PRODUCTIONUNITRESOURCE an einem gleichnamigen Attribut abgebildet. Diese Ressourcen erlauben die Spezifikation von Arbeitszeiten je Wochentag und somit eine direkte »Übersetzung« von Schichtplanungen (»Werkskalender«).

Um Abweichungen der tatsächlichen Kapazität von der geplanten Kapazität abbilden zu können, steht das Attribut .CAPACITYPLANNED zur Verfügung, welches ebenfalls eine Referenz auf ein Objekt vom Typ PRODUCTIONUNITRESOURCE ist.

⁵⁷Nach in der Frankfurter Allgemeinen Zeitung zitierten Angaben der Volkswagen AG (FAZ vom 13.1.2001, S. 60) liegt der Produktionskostenanteil für einen Golf 1,4 bei 24,3 %.

6.5.2. Einplanung von Aufträgen

Für die Einplanung eines Auftrages kann unter Umständen mehr als ein Werk in Frage kommen bzw. mehr als ein Zeitpunkt, an dem Fertigung des Auftrages begonnen werden kann. Um ein Werk bzw. einen Fertigungszeitpunkt so auszuwählen, dass die gewählte Strategie eines Automobilherstellers zur Auftragseinplanung realistisch abgebildet werden kann, wird im Rahmen des vorgestellten Ansatzes eine *Gütefunktion* verwendet (vgl. [Rit04, S. 40]), die die möglichen Alternativen bewertbar macht.

Der Termin zur Einplanung eines Kundenauftrags ist im ersten Schritt auf Basis des gewünschten Liefertermins abzuleiten. Der Termin zum Beginn der Fertigung in den jeweiligen Werken kann durch retrograde Terminierung (siehe [WLB⁺02, S. B3-34]) bestimmt werden, indem vom Kundenwunschtermin die prognostizierte Durchlaufzeit in der Distribution und die prognostizierte Durchlaufzeit im Werk abgezogen werden.

Die Durchlaufzeit durch das Distributionsnetzwerk wird auf Basis der Relation Werk/Händler festgelegt. Die Durchlaufzeit durch das Werk wird einer Durchlaufzeitverteilung entnommen, die als passend für den Auftrag⁵⁸ ermittelt wurde.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CProbabilityDistribution			PROBABILITYDISTRIBUTION	
<i>int</i>	.PlanValue	D.32	.PLANVALUE	E.46
CProductPattern 1..1 →	.ProductPattern	D.32, D.31,	.PRODUCTPATTERN	E.46, E.50
CDistributionSystem			DISTRIBUTION	
CRouteInformation 1..n →	.DistanceMatrix	D.10	.DISTANCEMATRIX	E.20
CRouteInformation			ROUTEINFORMATION	
<i>int</i>	.Time	D.17	.TIME	E.63

Tabelle 6.12.: Zentrale Parameter retrograde Terminierung

Obige Berechnung resultiert in einem Fertigungszeitpunkt, der bei exakter Einhaltung der prognostizierten Durchlaufzeiten für Fertigung und Transport des Fahrzeugs zum Händler bzw. Kunden genau zum gewünschten Lieferzeitpunkt führt. Äußert der Kunde einen Wunschtag, ergibt die Berechnung einen Fertigungstag.

Die jeweiligen Hersteller berechnen unter Umständen keinen voraussichtlichen Liefertermin, so ermittelt z.B. Volkswagen lediglich einen Termin (eine Woche) für die Fertigstellung eines Fahrzeugs und teilt diesen Termin dem bestellenden Händler mit und »überlässt« es dem Händler, daraus einen Liefertermin zu berechnen (vgl. [Gra01, S. 136]).

Im hier vorgestellten Ansatz wird unabhängig von der konkreten Vorgehensweise des Automobilherstellers ein voraussichtlicher Liefertermin für jedes Fahrzeug unter Nutzung aller zur Verfügung stehenden Informationen berechnet⁵⁹, um die Leistung des Auftragsabwicklungsprozesses bis hin zur Auslieferung des Fahrzeugs bewerten zu können.

⁵⁸An einem Objekt vom Typ PROBABILITYDISTRIBUTION kann ein Objekt vom Typ PRODUCTPATTERN definiert werden, wodurch erreicht werden kann, dass eine Durchlaufzeitverteilung nur für bestimmte Aufträge genutzt wird. So kann z.B. eine längere Durchlaufzeit für Fahrzeuge mit Sonderaufbauten (z.B. Krankenwagen) vorgesehen werden.

⁵⁹Es ist nahe liegend, dass die Informationen über Kapazitäten und Fahrpläne des Distributionsnetzwerkes den Händlern nicht regelmäßig vorliegen, so dass die Händler den Ankunftszeitpunkt eines Fahrzeugs unter Nutzung von Erfahrungswerten bestimmen müssen.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CDealer		BASEDEALER	
^{CPlant} 1..n →	.Plants	D.20	.PLANTS	E.18
	CPlanningDealer		DEALERPLANNING	
real	.FetchAheadFidelity	D.23	.FETCHAHEADFIDELITY	E.19
real	.BacklogFidelity	D.23	.BACKLOGFIDELITY	E.19
real	.WeightPrio	D.23	.WEIGHTPRIO	E.19
real	.WeightPrioPre	D.23	.WEIGHTPRIOPRE	E.19
real	.WeightPrioPost	D.23	.WEIGHTPRIOPOST	E.19
int	.StartPlanInterval	D.23	.STARTPLANINTERVAL	E.19
int	.EndPlanInterval	D.23	.ENDPLANINTERVAL	E.19
	CPlanning		PLANNING	
real	.LoadDiffPrio	D.34	.LOADDIFFPRIO	E.41
	CBOClass		PRODUCTCLASS	
real	.Weight	D.5	.WEIGHT	E.48

Tabelle 6.13.: Parameter zur Beeinflussung der Gütefunktion zur Auftragseinplanung.

Die Berechnung eines Plan-Fertigungstermins ist für jedes Werk durchzuführen, welches das beauftragte Fahrzeug bauen kann, um im Nachgang die Ergebnisse vergleichen zu können. Wird vom Automobilhersteller für den Zeitpunkt des Abschlusses der Bearbeitung eines Auftrages lediglich eine Woche⁶⁰ genannt, steht für die Fertigung eines Auftrages mehr als ein Tag zur Verfügung, so dass das prognostizierte Fertigungsende in der zugesagten Woche liegt.

In einem solchen Fall wird eine Einplanung an all diesen Tagen versucht und in den Auswahlprozess eingebracht. Ist aus Kapazitäts- oder Auslastungsgründen eine Einplanung, unter Einhaltung des gewünschten Zeitfensters für die Fahrzeugauslieferung, an keinem der Tage möglich oder wünschenswert, ist abzuwägen⁶¹, ob eine frühere oder eine spätere Fertigung des Auftrages in den ersten Schritt des Auswahlprozesses aufgenommen werden soll.

In Abbildung 6.10 wird die Situation exemplarisch für eine Planwoche dargestellt. Mittels der Parameter `.STARTPLANINTERVAL` und `.ENDPLANINTERVAL` kann das Einplanungsintervall jeweils um n -Tage in beide Richtungen verlängert werden.

Um die Eignung der jeweiligen Tage in den Werken für die Platzierung eines Auftrages zu ermitteln, werden drei Summanden herangezogen, die in die Berechnung der Eignung eines Tages für die Auftragseinplanung eingehen (mit H Händler, Werk W , Auftrag A , Datum D):

1. Das bevorzugte Lieferwerk eines Händlers als $E_W(H, W)$.
2. Die Erreichung des gewünschten Zieltermins als $E_T(D, A, W)$.
3. Die Auslastung des Werkes als $E_A(W, D)$.

Der erste Summand E_W wird durch Nutzung des Positionsparameters (interpretiert als *Prioritätswert*(W, H)) in der Liste der Vorzugswerke eines Händlers berechnet⁶²

⁶⁰Im Gegensatz zu einer tagesgenauen oder noch präziseren Terminzusage.

⁶¹Konkret: Im Einplanungsalgorithmus zu prüfen.

⁶²Diese »Doppelnutzung« der Listenposition eines Werkes erfolgt aus pragmatischen Gründen, um die Implementierung einer weiteren Klasse für diesen Modellbereich zu vermeiden.

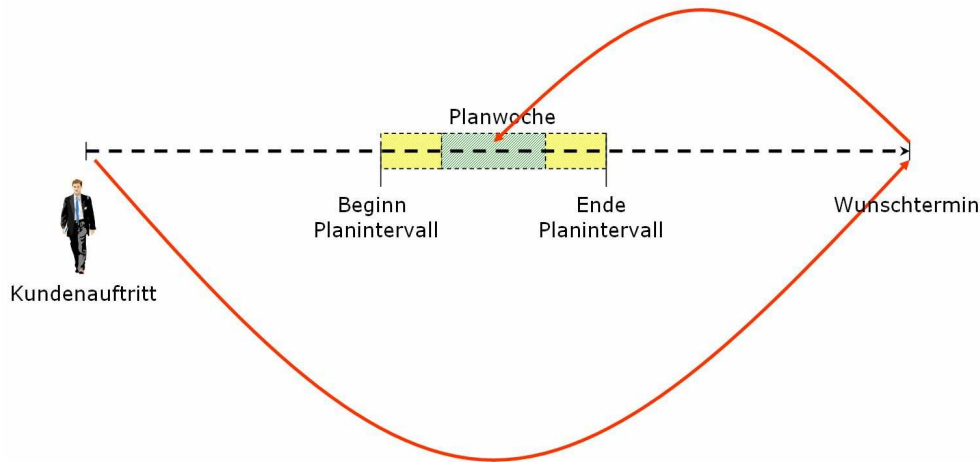


Abbildung 6.10.: Bestimmung des Einplanungsintervalls

$$E_W = \frac{1}{\text{Prioritätswert}(W, H)} \quad (6.6)$$

Der zweite Summand der Einplanungsfunktion dient der Priorisierung der Termin-einhaltung durch Berechnung der voraussichtlichen Abweichung vom Kundenwunschtermin (Δtd , positive Werte stehen für eine verspätete voraussichtliche Lieferung). Es werden unterschiedliche Gewichtungsfaktoren genutzt, abhängig davon, ob der Auftrag termingerecht, zu früh, oder zu spät ausgeliefert würde.

Diese Gewichtungsfaktoren können händlerspezifisch durch die Parameter `.WEIGHTPRIOPRE`, `.WEIGHTPRIO` und `.WEIGHTPRIOPOST` gesetzt werden. Wird das Fahrzeug voraussichtlich termintreu geliefert, ist der Gewichtungsfaktor 1. Der Summand $E_T(D, A, W)$ wird wie folgt berechnet:

$$E_T(D, A, W) = \begin{cases} \text{FetchAheadFidelity} & \Delta td < 0, \\ 1 & \Delta td = 0, \\ \text{BacklogFidelity} & \Delta td > 0. \end{cases} \quad (6.7)$$

Der dritte Summand der Einplanungsformel dient der Priorisierung gleichmäßiger Werksauslastung und wird aus folgenden drei Faktoren berechnet:

1. Dem »globalen« Werksauslastungsgewichtungsfaktor F_G ,
2. dem händlerspezifischen Auslastungsfaktor F_H und
3. dem produktklassenspezifischen Gewichtungsfaktor F_F .

Analog zur Berechnung des Summanden zur Bewertung der termingerechten Einplanung kann auch die Gewichtung der Werksauslastung für die Fälle termingerechte

Einplanung, Vorgriff⁶³ oder Rückstand⁶⁴ parametrisiert werden. Der händlerspezifische Auslastungsfaktor F_H wird analog zur obigen Gleichung berechnet:

$$F_H(D, A) = \begin{cases} WeightPrioPre & \Delta td < 0, \\ WeightPrio & \Delta td = 0, \\ WeightPrioPost & \Delta td > 0. \end{cases} \quad (6.8)$$

Der produktklassenspezifische Gewichtungsfaktor dient als Basis für die Berechnung der Auslastung eines Tages in einem Werk. Um reflektieren zu können, dass ein Werk durch Fahrzeuge einer Produktklasse stärker belastet werden kann als durch Fahrzeuge einer anderen Klasse, dient der Parameter `.WEIGHT` des Typs `PRODUCT-CLASS`. Durch Summieren aller »Gewichte« der Produktklassen der Fahrzeuge in einem Tagesprogramm wird F_F berechnet:

$$F_F = 1 - \frac{\sum_{i=0}^{Card_{FP}} Gewichtung(Produktklasse(Fahrzeug_i))}{MaxWeight}, \quad (6.9)$$

mit $Card_{FP}$ Anzahl Fahrzeuge im Tagesprogramm, $Fahrzeug_i$ das i -te Fahrzeug in einem Tagesprogramm und $MaxWeight$ das größte mögliche »Gewicht« eines Tagesprogrammes⁶⁵. Der Summand für eine gleichmäßige Werkauslastung ergibt sich als Produkt der drei genannten Faktoren:

$$E_W = F_G \times F_H \times F_F \quad (6.10)$$

Für die Gütefunktion G ergibt sich mit allen beschriebenen Summanden:

$$G(A, W, D) = E_W + E_T + E_A \quad (6.11)$$

Durch geeignete Einstellung der Parameter in 6.13 kann festgelegt werden, ob eher die Termineinhaltung oder die Werksauslastung bei der Auswahl von Zeitpunkten für die Einplanung von Aufträgen bevorzugt werden. Dabei sind sowohl beide Extreme – maximale Priorisierung gleichmäßiger Werksauslastung und maximale Priorisierung hoher Termintreue – möglich, wie auch Kompromisse zwischen diesen Zielgrößen.

Ist aufgrund der berechneten Gütewerte für die untersuchten Tage in den jeweiligen Werken ein Tag mit bester Eignung bestimmt worden, muss ein geeigneter Zeitpunkt für die Fertigung bestimmt werden. Die retrograde Terminierung auf Basis des gewünschten Zieltermins erfolgt in sekundengenauer⁶⁶ Auflösung, so dass nicht nur ein Tag für den Fertigungsbeginn berechnet wird, sondern ein Zeitpunkt an einem Tag, bis zu welchem der Fertigungsbeginn stattfinden muss, um den Zieltermin zu erreichen, wenn als Durchlaufzeiten die angesprochenen Planwerte angenommen werden.

Um sicherzustellen, dass nicht nur generell die Fertigungskapazität im Werk an dem betreffenden Tag verfügbar ist, sondern dass die Fertigung des Auftrages bis zu einem

⁶³ Auslieferung *vor* dem vom Kunden bzw. Händler gewünschten Liefertermin.

⁶⁴ Auslieferung *nach* dem vom Kunden bzw. Händler gewünschten Liefertermin.

⁶⁵ Maximal mögliche Anzahl von Aufträgen in einem Tagesprogramm multipliziert mit dem größten Wert des Attributes `.WEIGHT` über alle Produktklassen.

⁶⁶ Die Verwendung der Zeiteinheit »Sekunde« als Basis für jedwede Berechnung von Daten wurde gewählt, um Rundungsfehler auf ein vertretbares Maß zu reduzieren.

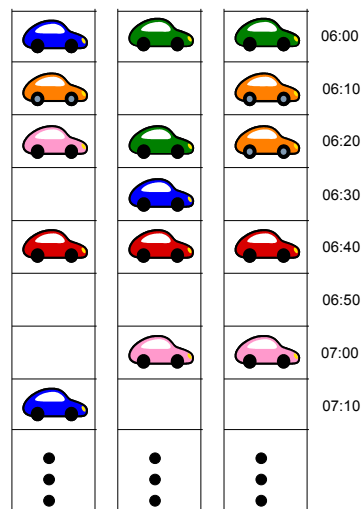


Abbildung 6.11.: Platzorientierte Ressourcenverwaltung

bestimmten Zeitpunkt begonnen werden kann, verwalten Objekte der Klasse PLANT einzelne Plätze, die für die Fertigung belegt werden können (siehe Abbildung 6.11).

6.5.3. Störungsmanagement

Die Zuordnung eines voraussichtlichen Fertigungstages zu einem Auftrag hat nicht notwendigerweise Bestand. Zahlreiche Einflüsse verändern die Reihenfolge der Aufträge. Neben den vom Automobilhersteller selbst verursachten Verschiebungen von Aufträgen, z.B. durch eine Sperrung aufgrund technischer Probleme, spielen Lieferengpässe der Lieferanten eine wichtige Rolle.

Solche Kapazitätsveränderungen können dazu führen, dass weniger Aufträge als geplant in einem Zeitabschnitt neu eingeplant werden können, aber auch dass die Kapazität für bereits eingeplante Fahrzeuge nicht mehr ausreicht. Ist der Engpass vor der Einplanung neuer Aufträge bekannt, die Kapazität reicht also zu jedem Zeitpunkt für die bereits eingeplanten Aufträge aus⁶⁷, müssen keine bereits genannten Liefertermine verschoben werden. Für neu eingeplante Aufträge kann, verursacht durch die Kapazitätsreduktion, eine verhältnismäßig lange Lieferzeit erwartet werden⁶⁸.

Tritt der Engpass hingegen zu einem Zeitpunkt auf, an dem bereits viele Aufträge in dem betroffenen Zeitabschnitt eingeplant sind, müssen unter Umständen viele betroffene Aufträge neu eingeplant werden. Sind Aufträge von dem Engpass nicht betroffen, können diese an die bisherige Stelle der neu eingeplanten Aufträge treten. Es wird also die Auftragsreihenfolge wieder an die verfügbaren Kapazitäten angepasst.

Für die Erstellung einer neuen Auftragsreihenfolge können zahlreiche Strategien zum Einsatz kommen:

⁶⁷Unter der Prämisse, dass ein Auftrag nur dann eingeplant wird, wenn ausreichend Kapazität vorhanden ist.

⁶⁸Dies gilt nur dann, wenn für die Fertigung des Auftrages die fragliche Engpasskapazität benötigt wird.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CDealer		BASEDEALER		
CPlant 1..n →	.Plants	D.20	.PLANTS	E.18
CPlant		PLANT		
CFailureRule 1..n →	.FailureStrategies	D.29	.FAILURERULES	E.42
int	.ReserveDistance	D.29	.RESERVEDISTANCE	E.42
bool	.RescheduleOrders	D.29	.RESCHEDULEORDERS	E.42
CPlanning		PLANNING		
bool	.PrefetchOrders	D.34	.PREFETCHORDERS	E.41
CPlanningDealer		DEALERPLANNING		
bool	.PrefetchOrders	D.23	.PREFETCHORDERS	E.19

Tabelle 6.14.: Zentrale Parameter Störungsmanagement

1. Es werden stets so viele Aufträge wie möglich gebaut. Hierzu werden aus dem Auftragsbestand Aufträge »vorgezogen«, also vor dem ursprünglich geplanten Termin gefertigt (Attribute .PREFETCHORDERS für globale Aktivierung oder Aktivierung für Aufträge einzelner Händler). Aufgrund des Engpasses werden zeitweilig nicht baubare Aufträge frühestmöglich gefertigt.
2. Die Aufträge, die aufgrund eines Engpasses nicht gefertigt werden können, werden neu eingeplant. Alle Aufträge, die nicht von dem Engpass betroffen sind, bleiben an der ursprünglichen Position (Attribut .RESCHEDULEORDERS).
3. Die Aufträge werden in ihrer Reihenfolge belassen, aber es wird zum nächsten möglichen Zeitpunkt Kapazität im erforderlichen Umfang reserviert.

Eine Auswahl zwischen diesen Varianten wird im Modell dadurch getroffen, dass ein Objekt einer von der Klasse FAILURERULE abgeleiteten Klasse (z.B. FAILURERULERESERVE) in die Objektliste .FAILURERULES an einem ein Werk repräsentierenden Objekt eingefügt wird. Das Attribut .PATTERN an diesen Objekten erlaubt es, genau festzulegen, welche Aufträge in welcher Weise zu verarbeiten sind, wenn es zu einem Engpass kommt.

So können z.B. Aufträge für Fahrzeuge einer Baureihe vorgezogen werden, während Aufträge für Fahrzeuge einer anderen Baureihe den zugeordneten geplanten Fertigungstermin behalten. Ebenso kann für eine bestimmte Eigenschaftskombination in einem Auftrag eine besondere Regel für die etwaige Umplanung spezifiziert werden.

Soll die Fertigungsreihenfolge der Aufträge weitergehend beeinflusst werden, kann eine Funktion angegeben werden, die eine »Kleiner-als«-Relation über den Aufträgen spezifiziert⁶⁹. Die Aufträge werden dann bezüglich dieser Relation aufsteigend sortiert. So lässt sich z.B. festlegen, dass (soweit möglich) Aufträge für Lagerfahrzeuge⁷⁰ neu eingeplant werden, während Aufträge von Endkunden den geplanten Fertigungstermin behalten (siehe Beispielskript in Anhang G.1). Wird kein Skript angegeben, werden die Fahrzeuge in der ursprünglich geplanten Fertigungsreihenfolge abgearbeitet.

⁶⁹Die Regeln zur Formulierung einer solchen Funktion sind in Anhang G in Form einer Grammatik abgebildet, wie sie vom Parser-Generator Bison (siehe <http://www.gnu.org/software/bison/bison.html>) verarbeitet werden kann.

⁷⁰Gemeint sind hier Aufträge, für die noch kein Endkunde gefunden wurde, d.h. die nach der Fertigung in ein Lager gestellt werden.

Es ist nahe liegend, dass die Festlegung der Regeln zum Störungsmanagement manifesten Einfluss auf die Kennzahlen eines Auftragsabwicklungsprozesses haben. Eine Priorisierung von Kundenaufträgen kann z.B. Schwächen im Prozess in der Auftragsabwicklung verdecken, solange ausreichend Aufträge für Lagerfahrzeuge als »Spielmasse« zur Verfügung stehen. Eine Erhöhung des Build-to-Order Anteils und somit eine Vergrößerung des Anteils von Aufträgen mit einem an den Endkunden kommunizierten Liefertermin, kann eine starke Verschlechterung der Liefertreue für die Endkunden mit sich bringen – ein Bewertungsergebnis, das nur dann errechnet werden kann, wenn das Modell des Auftragsabwicklungsprozesses auch Detailfragen des Störungsmanagements abbildet.

6.5.4. Fertigung

Die oben in ihrer Interpretation geschilderten Modellparameter beziehen sich ausschließlich auf die Einplanung von Aufträgen und die Kapazitätsverwaltung im Werk. Ergebnis der oben geschilderten Aktivitäten ist eine in definierter Reihenfolge vorliegende Menge von Aufträgen. Für jeden Tag des betrachteten Zeitraums kann auf Basis der geplanten Produktionsvolumina und der hierfür verfügbaren Kapazität bestimmt werden, wie viele Aufträge an einem Tag gebaut werden sollen. Für den »aktuellen« Tag wird zunächst ermittelt, wie viele Aufträge für diesen Tag in der Planung vorgesehen wurden.

Ist diese Anzahl bekannt, werden Aufträge in der Reihenfolge, in der die Fertigung geplant ist, neu am »aktuellen Tag« eingeplant. Diese Neueinplanung soll die gegenwärtige Verfügbarkeit der benötigten Kapazität sicherstellen und nutzt hier die parametrisierte tatsächliche Kapazität im Gegensatz zur Plankapazität, die genutzt wurde, um die Aufträge erstmalig einzuplanen. Sollte ein Auftrag aufgrund eines Engpasses nicht eingeplant werden können, wird der Auftrag übersprungen⁷¹ und der nächste Auftrag in der Reihe auf Baubarkeit am aktuellen Tag untersucht (siehe Kapitel 6.5.3).

Die so festgelegten Aufträge werden nun zu der mit der Position an diesem Tag assoziierten Startzeit gefertigt. Der Weg des Auftrags durch das Werk hängt im Folgenden davon ab, ob der Fertigungsprozess in Teilprozesse unterteilt wurde. Diese am Attribut .SUBPLANTS spezifizierte Liste enthält wiederum Objekte vom Typ PLANT. An allen diesen Werksobjekten kann eine Liste von Durchlaufzeitverteilungen angegeben werden, die wiederum über ein Produktsuchmuster (optional) bestimmten Produktklassen zugeordnet werden kann.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CPlant		PLANT	
CProcess 1..n →	.SubProcesses	D.28	.SUBPLANTS	E.42

Tabelle 6.15.: Parameter .SUBPLANTS in der Klasse PLANT

6.5.5. Zusammenfassung

Objekte der Klasse PLANT bilden die Produktionswerke von Automobilherstellern sowohl in Hinsicht auf Informationsflüsse als auch auf Materialflüsse ab. Für die Ein-

⁷¹Am nächsten Simulationstag wird eine erneute Einplanung versucht.

planung von Aufträgen werden Ressourcen sowohl auf Seiten der Zulieferer (Lieferantenkapazitäten) als auch auf Seiten der Werke selbst verwaltet (technische bzw. Personalkapazitäten). Im Unterschied zu Lieferanten werden von Werken Aufträge in einem platzorientierten Schema (tageweise) verwaltet. Ein weiterer wesentlicher Unterschied zu Lieferanten ist, dass diese Mengen anonym verwalten, während Werke die jeweiligen Plätze durch identifizierbare Aufträge belegen.

6.6. Zulieferer

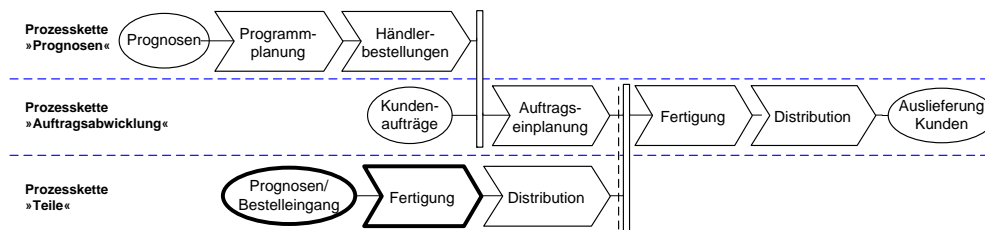


Abbildung 6.12.: Modellelement Zulieferer

Eine komplexe Aufgabe im Bereich der Fahrzeugproduktion besteht in der Koordination der Lieferanten in der Weise, dass einerseits jedes der vielen tausend Teile, die für die Fertigung eines Fahrzeugs benötigt werden, zum richtigen Zeitpunkt am Verbauort zur Verfügung steht, aber dass andererseits Kosten für die Bevorratung dieser Teile⁷² vermieden werden (siehe Kapitel 3.2.1).

Die Abbildung der Lieferanten im Modell erlaubt zum einen eine Parametrisierung der Kapazitäten, die für die Einplanung von Aufträgen zur Verfügung steht und zum anderen stochastisch oder zu fixierten Zeitpunkten auftretende Engpässe, die Abweichungen gegenüber den geplanten Kapazitäten mit sich bringen.

Neben den kurzfristigen (im Bereich von Stunden) auftretenden Versorgungsengpässen, wie sie sich z.B. durch den Ausfall einer LKW-Lieferung ergeben können, spielen weiterhin Engpässe eine Rolle, die sich durch eine Abweichung der tatsächlich abgerufenen Menge von den prognostizierten Mengen ergeben⁷³. Solche Abweichungen können z.B. dann auftreten, wenn der Hersteller die Reihenfolge der Aufträge stark abändert⁷⁴ und sich somit neue Liefermengen für den Zulieferer ergeben. Auch wenn sich Lieferanten nicht vertraglich verpflichten, sehr große kurzfristige Schwankungen in der Nachfrage befriedigen zu können, wird dieser jedoch bestrebt sein, dem Kundenwunsch nachzukommen. Erreicht der Lieferant die neue Mengenvorgabe nicht, entsteht ein weiterer Kapazitätsengpass, der wiederum eine Änderung der geplanten Auftragsreihenfolge nach sich ziehen kann.

⁷²Diese Kosten entstehen unter anderem durch Kapitalbindung, Flächenverbrauch, Verschrottungskosten für Teile, die nach einer technischen Änderung nicht mehr verbaut werden dürfen usw..

⁷³Lieferanten erhalten von vielen Herstellern periodisch eine Vorschau auf die vom Hersteller erwarteten Abnahmemengen (vgl. [VA02, S. 15]).

⁷⁴Solche Änderungen können durch das Störungsmanagement des Herstellers hervorgerufen werden (siehe 6.5.3).

Neben den oben genannten kurz- und mittelfristigen Kapazitätsengpässen können langfristige Abweichungen zwischen Nachfrage durch den Automobilhersteller und Lieferfähigkeit des Zulieferers durch eine zu pessimistische Auslegung der Fertigungskapazitäten entstehen. Eine solche Situation kann entstehen, weil der Hersteller die Nachfrage nach einem Produkt unterschätzt hat, oder weil der Lieferant im Rahmen seiner »internen Optimierung« nicht die Kapazitäten aufgebaut hat, die der Automobilhersteller verlangt hat⁷⁵.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CPlant		PLANT	
<i>int</i>	.SuppliersDB	D.29	.SUPPLIERS	E.42
	CSupplier		SUPPLIER	
CResourceProductionUnit 1..1 →	.CapacityActual	D.28	.CAPACITYACTUAL	E.71
CResourceProductionUnit 1..1 →	.CapacityPlanned	D.28	.CAPACITYPLANNED	E.71
<i>bool</i>	.DisablePrefetch	D.35	.DISABLEPREFETCH	E.71
<i>bool</i>	.IncludePrefetchSource	D.35	.INCLUDEPREFETCHSOURCE	E.71
<i>bool</i>	.IncludePrefetchTarget	D.35	.INCLUDEPREFETCHTARGET	E.71
CAssocDataIVInt2Float 1..n →	.MaxOutput	D.35	.MAXOUTPUT	E.71
CAssocDataIVInt2Float 1..n →	.UpperBandwidthDay	D.35	.UPPERBANDWIDTHDAY	E.71
CAssocDataIVInt2Float 1..n →	.LowerBandwidthday	D.35	.LOWERBANDWIDTHDAY	E.71
CAssocDataIVInt2Float 1..n →	.UpperBandwidthWeek	D.35	.UPPERBANDWIDTHWEEK	E.71
CAssocDataIVInt2Float 1..n →	.LowerbandwithWeek	D.35	.LOWERBANDWITHWEEK	E.71
<i>int</i>	.BandwidthRefDay	D.35	.BANDWIDTHREFDAY	E.71

Tabelle 6.16.: Zentrale Parameter Zulieferer

6.6.1. Verwaltung von Kapazitätsinformationen

Kapazitäten werden von den Objekten des Typs SUPPLIER⁷⁶ auf zwei Ebenen verwaltet: Zum einen durch zwei referenzierte Objekte vom Typ PRODUCTIONUNITRESOURCE⁷⁷ (siehe Kapitel 6.4) und zum anderen durch Methoden zur Verwaltung der Kapazitätsinformationen durch den Zulieferer. Durch diese Methoden werden sowohl Möglichkeiten zur »Vorproduktion« von Teilen als auch das Störungsmanagement des Zulieferers abgebildet (siehe Kapitel 6.6.5). Die Abbildung der internen Prozesse der Lieferanten ist abstrakter, als dies bei Werken (Typ PLANT) der Fall ist, um den Ressourcenbedarf (Arbeitsspeicher und Laufzeit) für die Simulation dieses Modellteiles zu begrenzen.

⁷⁵Die Beweggründe für einen Lieferanten, die Vorgaben des Automobilherstellers nicht »blind« zu übernehmen, sind der Automobilpresse häufig zu entnehmen. So werden von den Herstellern regelmäßig Wachstumszahlen für eine Baureihe angegeben, die nur zu errechnen sind, wenn der Hersteller gegenüber der direkten Konkurrenz überproportional wächst. Wenn also jeder Hersteller mit seinem Modell ein Wachstum von z.B. 10 % erreichen will, aber für das Modellsegment insgesamt nur 3 % angenommen werden, ist es möglicherweise unvernünftig, aus Sicht des Lieferanten Kapazitäten für ein Wachstum von 10 % vorzuhalten.

⁷⁶Und damit von Objekten des Typs SUPPLIERBTO und SUPPLIERBTS.

⁷⁷Zur Modellierung der Plan-Kapazität und der tatsächlichen Kapazität. Die Abbildung ist analog zu der in der Klasse PLANT. In der UML-Modellierung sind die beiden Attribute .CapacityPlanned und .CapacityActual in der Klasse CProducer definiert und stehen somit durch Vererbung in den Klassen CSupplier und CPlant zur Verfügung.

Die von Zulieferern produzierten Teile werden lediglich auf der Ebene »Anzahl pro Variante« verwaltet. Die Teile können auf Seiten des Zulieferers nicht einem einzelnen Auftrag zugeordnet werden. Die Zuordnung eines Teils zu einem Auftrag erfolgt erst im Werk. Ist allerdings ein Auftrag in einem Werk eingebucht, sind die hierfür erforderlichen Teile beim Zulieferer reserviert. Das heißt, die Implementierung der Klasse SUPPLIER verwaltet für jeden Abnehmer und für jede Teilevariante »Konten«, denen zu entnehmen ist, wie viele Teile für einen Tag von dem jeweiligen Abnehmer gebucht wurden. Die Zulieferer übermitteln den Werken im Modell Informationen, wie viele Teile an einem Tag zur Verfügung stehen, aber nicht, zu welchem Zeitpunkt individuelle Teile fertig gestellt werden⁷⁸.

6.6.2. Vorproduktion von Teilen

Ebenso wie die Automobilhersteller suchen auch die Lieferanten nach einer möglichst gleichmäßigen Auslastung ihrer Produktion. Eine gleichmäßige Auslastung der Fertigungseinrichtungen des Automobilherstellers bringt aber nicht notwendigerweise eine ebensolche des Lieferanten mit sich. Stellt der Lieferant ein Teil her, welches in jedem Fahrzeug einer Baureihe Eingang findet, profitiert der Lieferant direkt von einer gleichmäßigen Auslastung des Herstellers.

Werden die Produkte des Lieferanten z.B. nur in solche Fahrzeuge verbaut, in denen eine bestimmte Eigenschaftskombination anzutreffen ist, kann der Teilebedarf stark schwanken. Aus fertigungstechnischen Gründen strebt der Hersteller für einige Eigenschaften eine Gleichverteilung an, da z.B. mit der Montage der Teile für diese Eigenschaften ein zusätzlicher Personalbedarf verknüpft ist⁷⁹, aber aus nahe liegenden Gründen kann nicht über eine beliebige Anzahl Eigenschaften eine Gleichverteilung erreicht werden, wenn viele Einbauraten dieser Eigenschaften schwanken.

Ein weiterer Grund für Schwankungen der Teilebedarfe kann die Distribution der Fahrzeuge in die Zielmärkte sein. Für Regionen mit sehr geringer Marktnachfrage kann es sinnvoll sein, Fahrzeuge hierfür gebündelt zu fertigen, um einen kostengünstigen Transport organisieren zu können. Wird z.B. in einem nordeuropäischen Markt ein besonders verstärkter Stoßfänger stark nachgefragt, der zudem in anderen Märkten nicht geordert werden kann, ist eine sehr hohe Bedarfsschwankung für diese Stoßfänger nahe liegend. Um trotz dieser Bedarfsschwankungen eine gleichmäßige Auslastung seiner Produktion zu erreichen, kann der Lieferant Bedarfsspitzen durch Vorproduktion befriedigen.

Diese Vorproduktion kann in zwei Kategorien unterteilt werden: Zum einen die »anonyme« Vorproduktion, die auf die Erhaltung eines bestimmten Lagerbestandes abzielt⁸⁰ und zum anderen die Vorproduktion bei bekannter Bedarfssituation. Letzterer

⁷⁸Wird in einem Modell ein Distributionsweg vom Zulieferer zu einem Werk abgebildet, werden die Zeitpunkte der Fertigstellung von Teilen berechnet, um ermitteln zu können, wann Behälter (Typ CONTAINER) befüllt und bereit zum Transport sind. Hierfür werden aber keine Teile individuell betrachtet, sondern es wird die Ausbringung des Zulieferers gleichförmig über die Arbeitszeit des Zulieferers (entnommen aus dem jeweils »zuständigen« Objekt vom Typ PRODUCTIONUNITRESOURCE) verteilt.

⁷⁹Ein Fahrzeug mit einem Schiebedach verursacht einen höheren Montageaufwand, als ein Fahrzeug ohne Schiebedach.

⁸⁰Dies ist dann möglich, wenn die Produkte des Lieferanten so wenig variantenreich sind, dass eine Vorproduktion sinnvoll möglich ist. Bei sehr variantenreichen Teilen bzw. Komponenten wie z.B.

Fall kann eintreten, wenn der konkrete Bedarf einem Zulieferer, z.B. zehn Tage vor dem letztmöglichen Liefertermin gemeldet wird. Der Zulieferer kann so eine etwaige Bedarfsspitze auf mehrere Tage verteilen.

Zur Steuerung dieser Form der Vorproduktion werden im Modell lediglich drei Parameter benötigt: `.DISABLEPREFETCH` deaktiviert die Vorproduktion, die beiden Parameter `.INCLUDEPREFETCHSOURCE` und `.INCLUDEPREFETCHTARGET` steuern das Zusammenspiel zwischen Vorproduktion und der Begrenzung der maximalen Ausbringung des Zulieferers (Parameter `.MAXOUTPUT`). Für jeden Tag der Simulation werden Informationen verwaltet⁸¹, ob an einem Tag Teile vorproduziert und an welchem Tag diese Teile benötigt werden. Dieser Tag der Simulation ist wiederum mit einem Zähler versehen, der Informationen verwaltet, wie viele Teile insgesamt vorproduziert wurden, die an diesem Tag geliefert werden sollen.

Es ist also stets nachvollziehbar, wie viele Teile am Tag des Bedarfs vom Zulieferer produziert wurden und an welchen Tagen zuvor für diesen Tag Teile vorproduziert wurden und wie viele davon⁸².

Die exakte »Buchführung« über die Vorproduktion bei einem Zulieferer ist deswegen erforderlich, weil die Menge der vorzuproduzierenden Teile im Verlauf der Simulation schwankt. Durch die Anpassung von durch einen Händler gebuchten Aufträgen an einen Kundenwunsch kann nicht nur ein zusätzlicher Bedarf entstehen, sondern es kann auch ein Minderbedarf die Folge sein. Im Falle eines Minderbedarfes wird die Anzahl der vorproduzierten Teile an dem Tag reduziert, der am weitesten vom Tag des Bedarfes entfernt ist. Das Verfahren stellt so sicher, dass eine Buchung von n Teilen und eine darauf folgende Freigabe dieser Teile zur ursprünglichen Situation beim Lieferanten führt.

6.6.3. Begrenzung der Ausbringung

Die Möglichkeit der Vorproduktion erhöht unter Umständen die Flexibilität des Zulieferers auf Schwankungen zu reagieren stärker, als es vom Modellierer gewünscht ist⁸³. Daher wurde zu den oben genannten Parametern ein weiteres Attribut hinzugefügt (`.MAXOUTPUT`), welches Obergrenzen für die Leistungsfähigkeit des Zulieferers abbildet.

Mit Hilfe dieses Parameters lassen sich zwei Arten von Restriktionen abbilden: Zum einen kann die Anzahl der tatsächlich gelieferten Teile nach oben beschränkt werden. Hier wird die Summe aus den an diesem Tag fertig gestellten Teilen und solchen aus der Vorproduktion gebildet und nach oben begrenzt⁸⁴. Zum anderen kann die in einem

dem Kabelbaum steht die Konfiguration und der gewünschte Liefertermin erst wenige Tage vor der Fertigung des Fahrzeugs fest (frühestens zum Einfrierzeitpunkt des Auftrags); hier können allerdings Segmente des Kabelbaumes vormontiert werden.

⁸¹Jeweils pro Teilevariante und Abnehmer.

⁸²Konkretes Beispiel: Am 55. Simulationstag sei der Bedarf des Werkes 100, die max. Ausbringung des Lieferanten sei 50. Es müssen also 50 Teile vorproduziert werden, um den Bedarf zu befriedigen. Beträgt die freie Kapazität an den Tagen davor (54. und 53. Simulationstag) jeweils 25, dann wird in einer Datenstruktur jeweils für beide Tage (53 und 54) vermerkt, dass 25 Teile für den 55. Simulationstag vorproduziert wurden.

⁸³Die Menge der an einem Tag verbauten Teile übersteigt die im Realsystem vorhandene Obergrenze.

⁸⁴Der Schalter `.INCLUDEPREFETCHTARGET` (siehe Tabelle 6.16) ist auf den Wert »wahr« zu setzen. Durch diese Parametrisierung wird die Anzahl der von einem Lieferanten pro Tag gelieferten Teile nach oben begrenzt.

Zeitraum beim Lieferanten produzierte Menge Teile (inkl. Anzahl vorproduzierter Teile) nach oben begrenzt werden⁸⁵.

Letzterer Parameter kann z.B. eingesetzt werden, um die Verteilung der Kapazität einer von zwei Lieferanten gemeinsam genutzten Ressource nach oben zu steuern⁸⁶.

6.6.4. Flexibilitätsbandbreiten

Ein weiteres wichtiges Merkmal eines Lieferanten ist neben der absoluten Kapazität die Flexibilität, mit der dieser auf veränderte Anforderungen eines Automobilherstellers reagieren kann. Diese Flexibilität spielt an zwei Stellen eine Rolle: Zum einen als »echte« Flexibilität eines Lieferanten – kann der Lieferant tatsächlich die geforderte, abweichende Menge liefern – und zum anderen als angenommene Flexibilität, also den Annahmen über den Zulieferer, mit denen der Automobilhersteller in seinem Auftragsabwicklungsprozess plant. Es kann vermutet werden, dass in der Praxis sowohl zu beobachten ist, dass die tatsächliche Flexibilität des Lieferanten höher als die angenommene ist, als auch der umgekehrte Fall.

Beide Fälle können durch die letzten fünf Parameter der Klasse SUPPLIER aus Tabelle 6.16 abgebildet werden. Die Repräsentation dieser Flexibilitäten als Wertverläufe (siehe Diagramm E.71) erlaubt eine Gestaltung als »Trichter«, so dass eine hohe Flexibilität bei langfristig angekündigten Änderungswünschen zusammen mit einer niedrigen Flexibilität bei kurzfristigen Änderungen vorgesehen werden kann. Als Bezugszeitraum für die eingestellten Flexibilitäten können dabei sowohl Tage als auch Wochen dienen⁸⁷.

Es stehen somit fünf Gruppen von Parametern zur Verfügung, um die Kapazität von Zulieferern abzubilden:

1. **Zulieferer.** Ein Zulieferer kann mehrere Abnehmer beliefern. Mehrere Zulieferer können dasselbe Teil an einen Abnehmer liefern. Die Zulieferer identischer Teile können jeweils pro Abnehmer priorisiert werden.
2. **Stücklisten.** Durch Stücklisten kann der Ressourcenbedarf für ein Teil auch in »entfernten« Teilen des Lieferantennetzwerkes beeinflusst werden.
3. **Steuerung der Vorproduktion.** Durch Vorproduktion von Teilen können Zulieferer Schwankungen in der Nachfrage der Automobilhersteller ausgleichen.
4. **Beschränkung der Ausbringung.** Die Schwankungen in der tatsächlich in einem Zeitraum dem Werk zur Verfügung gestellten Kapazität lassen sich durch diese Parameter beschränken.
5. **Flexibilitätsbandbreiten.** Die Fähigkeit des Lieferanten auf kurz-, mittel- und langfristig geäußerte Anforderungen eines Herstellers zu reagieren, kann mit die-

⁸⁵Der Schalter »IncludePrefetchSource« ist auf den Wert »wahr« zu setzen.

⁸⁶Nutzen zwei Lieferanten eine gemeinsame Ressource, wird dadurch zunächst keine Verteilung dieser Ressource auf die Lieferanten impliziert. Durch Nutzung des zuletzt beschriebenen Modellparameters kann erreicht werden, dass jeder der Lieferanten zu keinem Zeitpunkt mehr als z.B. 70 % der Kapazität der Ressource nutzt.

⁸⁷Ein Beispiel für einen solchen Wertverlauf: Die Intervalle 0 – 2, 2 – 4, 4 – 8, 8 – 30 zusammen mit den Werten 0, 0,05, 0,15 werden interpretiert als: Keine Flexibilität null bis zwei Wochen vor Fertigungsbeginn, 5 % Flexibilität zwei bis vier Wochen vor Fertigungsbeginn usw..

sen Parametern eingestellt werden.

6.6.5. Störungsmanagement

Durch die oben beschriebenen Möglichkeiten zur Modellierung von Kapazitäten lassen sich bereits Aussagen darüber gewinnen, wie sich ein System im Falle einer »Störung« einer unvorhergesehenen Kapazitätsreduktion eines Zulieferers verhält. So kann in einem Zeitraum die Ausbringung des Zulieferers gezielt gesenkt werden, um eine solche Störung abzubilden. Vorteil einer solchen Herangehensweise vor stochastisch erzeugten Störungen (siehe unten) ist, dass Umfang und Dauer der Störung im Vorhinein festgelegt werden können und somit durch die Experimentgestaltung genau festgelegt wird, welche Kapazitätssituation untersucht wird.

Auf der anderen Seite lassen sich Aussagen über die »Robustheit« eines Auftragsabwicklungsprozesses gegenüber Störungen in der Teileversorgung nur schwer anhand weniger Testfälle belegen, da der Einfluss einer veränderten Kapazitätssituation stets im Zusammenhang mit der zu diesem Zeitpunkt vorliegenden Auftragssequenz bzw. den Aufträgen, die während des Bestehens der Störung hinzukommen, gesehen werden muss. So kann eine Absenkung der verfügbaren Kapazität für das Teil »Halter XY« ohne Auswirkungen auf die Auftragssequenz bleiben, weil zufällig zu diesem Zeitpunkt nur sehr wenige Aufträge eingeplant wurden, für deren Fertigung dieser Halter benötigt wird.

Kommt eine zweite Störung beim Teil »Halter YZ« hinzu, können die Auswirkungen additiv sein – für keinen Auftrag werden die betroffenen Teile zusammen benötigt – oder aber im anderen Extrem wird die zweite Störung nur wirksam, wenn die Kapazitätseinbuße größer als bei der ersten Störung ist – die Teile werden stets zusammen verbaut. Die stochastische Erzeugung von Störungen erlaubt es, eine große Anzahl von Experimenten durchzuführen, um die Bandbreite der Ergebnisse auszuloten, mit denen zu rechnen ist. Aufgrund der enormen Vielfalt möglicher Ausgangssituationen zum Zeitpunkt des Auftretens einer Störung, sind die Aussagen auch sehr umfangreicher Experimentserien allerdings mit Bedacht zu interpretieren (siehe Kapitel 8.3).

Die stochastische Veränderung der Kapazität eines Lieferanten durch eine Störung wird an der zugeordneten Ressource definiert (siehe Kapitel 6.4). Das Ressourcenobjekt in der Simulation »benachrichtigt« über eine hierfür bestimmte Schnittstelle⁸⁸ alle Instanzen, die sich für solche Nachrichten registriert haben. Zulieferer-Objekte führen eine solche Registrierung stets durch und werden daher über das Auftreten einer Störung benachrichtigt. Der Zulieferer verschickt diese Nachricht seinerseits an das zugeordnete Werk.

In dem von der Störung betroffenen Zeitraum wird durch die Implementierung des Lieferanten zunächst geprüft, inwieweit die Beeinträchtigung in Richtung seiner Kunden wirksam ist. Dies ist genau dann der Fall, wenn die veränderte Kapazität kleiner als die von den Kunden angeforderte Kapazität ist. Wird eine Unterdeckung festgestellt, kann versucht werden, durch Vorproduktion die Auswirkungen der Störung zu minimieren, falls die Parametrisierung eine solche Vorproduktion zulässt. Wird eine Mengenerhöhung durch ein Werk angefragt, findet die Prüfung auf Basis der neu be-

⁸⁸Alle Klassen, in denen eine Reaktion auf Störungen implementiert ist, sind von der Klasse `CFailureClient` abgeleitet (siehe Abbildung D.30).

rechneten reduzierten Kapazität⁸⁹ statt. Eine Reduktion der angefragten Menge durch einen Abnehmer wird zunächst zur Reduktion der Kapazitätsunterdeckung in einem Zeitraum genutzt, also der Differenz zwischen ursprünglich zugesagter Liefermenge und der nach Eintreten der Störung absehbaren Liefermenge⁹⁰.

6.6.6. Zusammenfassung

Die Parametrisierung einer Störung an einer Ressource, die von einem Zulieferer referenziert wird, erlaubt es, stochastische Einflüsse auf die Kapazität der Lieferanten abzubilden. Nach Eintreten einer Störung und der Benachrichtigung aller Abnehmer, berechnet die Implementierung der Klasse SUPPLIER die kapazitative Unterdeckung und erlaubt es, der nutzenden Klasse, z.B. durch Umbuchung von Aufträgen, die Unterdeckung zu reduzieren.

6.7. Distribution

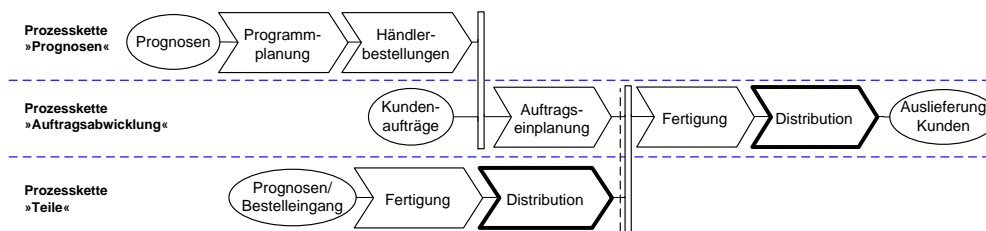


Abbildung 6.13.: Modellelemente Distribution

Die Distribution eines Automobils vom Werk zu dem bestellenden Händler und weiter zum Endkunden ist aus mindestens zwei Gründen von großer Bedeutung für die Gestaltung von Auftragsabwicklungsprozessen: Zum einen nimmt der Fahrzeugtransport verglichen mit der eigentlichen Fertigung viel Zeit in Anspruch und zum anderen ist das gebundene Kapital bezogen auf den Auftrag nach Beginn der Fertigung stark gestiegen⁹¹. Daher ist die Distribution ein Ansatzpunkt sowohl für Strategien zur Reduktion der Lieferzeit als auch zur Reduktion der Kapitalbindungskosten.

Zur effektiven Beurteilung von Veränderungen in diesem Bereich ist eine detaillierte Betrachtung der Transportprozesse erforderlich (siehe Kapitel 4.8). Dies gilt insbesondere dann, wenn die Datengrundlage für eine abstrakte Betrachtung z.B. auf Basis durchschnittlicher Transportzeiten pro Relation für neu gestaltete Prozesse nicht ohne Weiteres aus Vergangenheitsdaten ableitbar ist: Wenn also z.B. die Verwendung kleinerer aber häufiger fahrender Schiffe für die Belieferung des US-Marktes untersucht werden soll, ist unter der Randbedingung, dass diese kleinen Schiffe weniger

⁸⁹Es ist auch möglich, durch eine Störung eine Kapazitätserhöhung zu modellieren; die Implementierung lässt auch eine solche Parametrisierung zu. Hier wird allerdings von einer Reduktion ausgegangen.

⁹⁰Die tatsächliche Liefermenge kann durch geeignete Nutzung des Parameters .CAPACITYACTUAL nochmals verändert sein.

⁹¹Die Automobilhersteller übernehmen die Teile der Zulieferer teilweise erst bei Anlieferung am Band in Besitz, bis dahin hat der Lieferant die Kapitalbindungskosten zu tragen.

Verladehäfen in Nordamerika anfahren, die durchschnittliche Transportzeit nicht mehr leicht ableitbar, da die in den Häfen angelandeten Volumina auch in der prozentualen Verteilung schwanken.

Um die Untersuchung der Prozesse der Distribution in oben angedeuteter Weise durchführen zu können, wird im vorgestellten Ansatz der Aufbau von nahezu beliebig komplexen Transportnetzwerken⁹² ermöglicht.

Da innerhalb eines Simulationsexperimentes dynamisch bestimmt werden kann, wie viele Fahrzeuge in den jeweiligen Ziel-Häfen eines Schiffes entladen werden⁹³, ist eine Untersuchung der oben geschilderten Fragestellung ohne Weiteres und bei hoher Ergebnisqualität möglich.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CDealer			BASEDEALER	
CTransportRelation 0..n →	.TransportRelations	D.12	.TRANSPORTRELATIONS	E.81
CTransportRelation			TRANSPORTRELATION	
int	.MaxWait	D.12	.MAXWAIT	E.81
CDistributionChannel 0..n →	.DistributionChannels	D.11	.DISTRIBUTIONCHANNELS	E.21
CDistributionChannel			DISTRIBUTIONCHANNEL	
CRoutingTable 1..1 →	.RoutingTable	D.13	.ROUTINGTABLE	E.65
CDistributionChannel 0..n →	.SubChannels	D.11	.SUBCHANNELS	E.21
CDestination 1..1 →	.Source	D.11	.SOURCE	E.21
CDestination 1..1 →	.Destination	D.11	.DESTINATION	E.21
CProbabilityDistribution 1..1 →	.TransportTime	D.32	.TRANSPORTTIME	E.46
CAssocDataFloat 1..1 →	.ResCapacity	D.39	.CAPACITY	E.5
String	.SortCondition	D.11	.SORTCONDITION	E.21
CRouting			ROUTING	
CAssocDataFloat 1..1 →	.Probability	D.13	.PROBABILITY	E.64
int	.BaseGoodness	D.13	.BASEGOODNESS	E.64
int	.Reference	D.13	.REFERENCE	E.64
CRoutingTable			ROUTINGTABLE	
bool	.PreferScheduled	D.13	.PREFERSCHEMULED	E.65
int	.PreferScheduledRating	D.13	.PREFERSCHEMULEDRATING	E.65
bool	.CapacityLimit	D.13	.CAPACITYLIMIT	E.65
int	.CapacityLimitRating	D.13	.CAPACITYLIMITRATING	E.65

Tabelle 6.17.: Zentrale Parameter Distribution.

Das grundlegende Konzept für die Beschreibung von Distributionsnetzwerken ist der *Distributionskanal*, der den Transportweg zwischen zwei Punkten (Attribute .SOURCE und .DESTINATION) beschreibt. Die diese Punkte abbildenden Objekte enthalten als Parameter die geographische Breite und Länge des Start- und des Zielpunktes des Distributionskanals, um einen Zeitverbrauch für die Be- und Entladung abbilden zu können. Die Zeit, die für den Transport benötigt wird, kann durch den Parameter .TRANSPORTTIME festgelegt werden (als Wahrscheinlichkeitsverteilung).

Die Kapazität der Transportmittel kann durch das Attribut .CAPACITY beeinflusst werden. Die Möglichkeit der Verwendung eines Wertverlaufes an dieser Stelle erlaubt

⁹²Größenbeschränkungen werden durch die verwendete Hard- und Software gesetzt und liegen weit jenseits praktisch erreichter Modellgrößen. Die Abbildung des gesamten Distributionsnetzes eines Automobilherstellers mit allen Transportrelationen ist also realisierbar.

⁹³Die Häfen sind dann als Teil der Transportwege der Fahrzeuge von den Werken zu den ausliefernden Händlern definiert.

die Veränderung der Transportkapazität innerhalb des Simulationszeitraumes. Das Attribut .SORTCONDITION ermöglicht es, die Fahrzeuge im Eingangspuffer des Distributionskanals zu sortieren, so dass z.B. die Fahrzeuge in der Reihenfolge abgefahren werden, in der sie den Kunden versprochen wurden (d.h. sortiert nach dem den Kunden genannten Liefertermin). So lassen sich z.B. Veränderungen der Auftragsreihenfolge durch die Fertigung teilweise kompensieren. Parametrisiert wird hier eine Methode, die zwei Fahrzeuge als Parameter erhält und eine »kleiner als« Relation implementiert. Ein Beispiel für ein solches Skript ist:

```
if(@[0].PlannedEOP < @[1].PlannedEOP ; 1; 0);
```

Die Grammatik der Skriptsprache ist in Anhang G.2 (Seite 321) aufgeführt. Er gibt die Auswertung des »kleiner« Operators den Wert »Wahr«, evaluiert der gesamte Ausdruck zu 1, zu 0 sonst. Die Sortierung wird hier nach dem geplanten Ende der Fertigung durchgeführt, also in der Reihenfolge, die durch eine Produktion ohne »Verwirbelung« entstanden wäre⁹⁴.

Neben der Nutzung der oben dargestellten Modellelemente zur Abbildung des Transportes von gefertigten Fahrzeugen können analog auch die Transportwege zwischen Lieferanten (Klassen SUPPLIERBTS und SUPPLIERBTO) oder von Lieferanten zu Werken (Klasse PLANT) abgebildet werden. Anstatt Fahrzeuge werden hier *Behälter* transportiert, die wiederum Teile bzw. Komponenten enthalten. Eine nähere Beschreibung der Modellierungsmöglichkeiten erfolgt in Kapitel 6.7.4.

6.7.1. Transportrelationen

Zur Beschreibung der Wege, die die Fahrzeuge von den Werken zu den Händlern bzw. Kunden nehmen, können zwei unterschiedliche Verfahren genutzt werden: Zum einen hierarchisch strukturierte Transportrelationen und zum anderen die implizite Beschreibung der Transportwege durch »Routing-Tabellen«.

Für den ersten der beiden Beschreibungswege werden für alle Relationen $\{Werk, H\ddot{a}ndler\}$ Wege angegeben (Attribut .TRANSPORTRELATIONS, siehe Tabelle 6.17), auf denen Transportmittel verkehren, die Fahrzeuge transportieren. Verbunden mit einem Objekt des Typs TRANSPORTRELATION ist ein Distributionskanal⁹⁵, der den Transportweg zwischen Werk und Händler abbildet. Eine Detaillierung dieses Transportweges kann durch die Aufzählung von Teilstücken erreicht werden, die als Parameter .SUBCHANNELS eines Distributionskanals angegeben werden. Diese können wiederum rekursiv unterteilt werden. Die Transportwege zwischen Händlern und Werken müssen dabei nicht disjunkt sein, eine gemeinsame Nutzung von Teilstücken ist möglich.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CPlant		WERK	
CProcess 0..n →	DistributionStart	D.29	DistributionStart	E.42

⁹⁴Genauer: Die Fahrzeuge werden exakt mit der im Vorhinein geplanten Durchlaufzeit gefertigt. Die geplante Durchlaufzeit kann für die jeweiligen Aufträge, z.B. abhängig von der Ausstattung, unterschiedlich sein.

⁹⁵Das Attribut .DISTRIBUTIONCHANNELS erlaubt die Spezifikation weiterer Distributionskanäle, die im Fall eines Engpasses ausgewählt werden können (siehe unten).

CRouting			ROUTING	
CProcess 0..n →	.OUT	D.13	.OUT	E.64
CAssocDataFloat 0..n →	.Probability	D.40	.PROBABILITY	E.3
int	.Reference	D.13	.REFERENCE	E.64

Tabelle 6.18.: Zentrale Parameter Bestimmung Transportwege.

Bei der obigen Form der Abbildung von Distributionsnetzwerken steht der Transport des Fahrzeugs vom Werk zum Händler im Vordergrund. Der Weg eines Fahrzeugs wird durch Aufzählung von Teilstücken festgelegt, Entscheidungen über alternative Routen spielen eine untergeordnete Rolle⁹⁶. Daher eignet sich diese Form der Modellierung in erster Linie für Modelle, in denen keine Auswahl über Transportwege zu treffen ist.

Ist hingegen der Weg eines Fahrzeugs nicht bereits durch die Relation $\{Werk, Händler\}$ festgelegt, bzw. soll eine detailliertere Abbildung der Distributionsprozesse vorgenommen werden, kann eine alternative Formulierung des Distributionsnetzes genutzt werden. Hier wird zunächst ein Graph festgelegt, der alle Punkte miteinander verbindet, zwischen denen Transporte durchgeführt werden sollen⁹⁷.

Die Entscheidung, welche der von einem Knoten ausgehenden Kanten – also welches Transportmittel im Anschluss an das vorherige – zu nutzen ist, wird dynamisch auf Basis eines festzulegenden Kriteriensatzes getroffen. In die Auswahlentscheidung fließen (optional) die Auslastung von Transportmitteln und die Transportdisposition (siehe Kapitel 6.7.2) ein. Das Attribut `.ROUTINGTABLE` eines Distributionskanals (siehe Tabelle 6.17) erlaubt die Parametrisierung von diesen alternativen Routen.

Für die konkrete Bestimmung eines Transportweges wird eine Gütefunktion verwendet, in die folgende Summanden eingehen:

1. Verkehrt das Transportmittel gemäß Fahrplan (Attribut `.PREFERSCHEDULED`)? Ist dies der Fall, wird das Attribut `.PREFERSCHEDULEDRATING` als Summand herangezogen. Mittels dieses Parameters lässt sich z.B. erreichen, dass ein ohnehin (gemäß Fahrplan) verkehrendes Transportmittel zunächst ausgelastet wird, bevor auf LKW als Transportmittel zurückgegriffen wird.
2. Ist eine Auslastungsgrenze bestimmt (Attribut `.CAPACITYLIMIT`)? Es ist unter Umständen nicht sinnvoll, Fahrzeuge auf einem Weg zu transportieren, von dem bekannt ist, dass Zwischenstationen vollständig belegt sind. Ist an einem (festzulegenden) Punkt des Distributionskanals die Auslastung größer oder gleich dem Parameter `.CAPACITYLIMIT`, fließt der Wert des Attributes `.CAPACITYLIMITRATING` in die Gütefunktion als Summand ein.
3. Welche »Basisgüte« ist für einen Transportweg vorgesehen (Attribut `.BASEGOODNESS`)? Um grundlegende Präferenzen für Transportwege auszudrücken, kann der Parameter `.BASEGOODNESS` genutzt werden.

⁹⁶Durch Festlegung des Parameters `.MAXWAIT` an der Klasse `TRANSPORTRELATION` (siehe Abbildung E.81) kann ein alternativer Distributionskanal ausgewählt werden, wenn die Wartezeit auf dem zunächst vorgesehenen Weg ein festzulegendes Maß überschreitet.

⁹⁷Geographische Informationen werden im vorgestellten Ansatz in erster Linie zu Visualisierungszwecken genutzt; es ist also nicht zwingend erforderlich, dass die oben angesprochenen »Punkte« auch in geographischer Hinsicht sinnvoll gewählt sind.

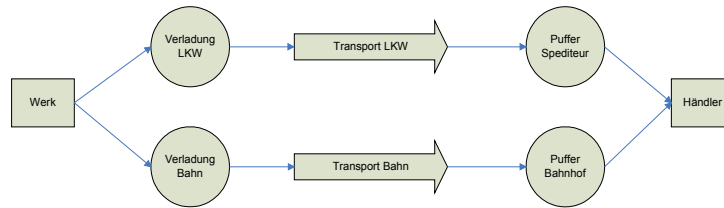


Abbildung 6.14.: Beispiel Distributionsnetzwerk

Da die Auswahl eines Distributionsweges unter Umständen nicht ausschließlich durch die Betrachtung des nächsten Teilstückes sinnvoll zu treffen ist, kann durch den Parameter `.REFERENCE` festgelegt werden, ob die oben beschriebenen Parameter auf den nächsten, den übernächsten oder den darauf folgenden Distributionskanal bezogen werden sollen⁹⁸. Soll z.B. ein Fahrzeug vom Werk in Abbildung 6.14 zum bestellenden Händler transportiert werden, gilt es zwischen zwei Transportmöglichkeiten auszuwählen. Ist der »Puffer Bahnhof« ein möglicher Kapazitätsengpass, sollte bei Überschreiten einer Auslastungsgrenze ein Transport per LKW gewählt werden.

Durch geeignete Setzung des Parameters `.REFERENCE` (auf den Wert drei) kann erreicht werden, dass bei der Wahl zwischen dem oberen und dem unteren Distributionsweg die Berechnung des Wertes der Gütefunktion auf die Auslastung von »Puffer Spediteur« und »Puffer Bahnhof« zugreift. Überschreitet die Auslastung eines der Puffer den Wert des Attributes `.CAPACITYLIMIT`, dann geht in die Bewertung dieses Transportweges der Summand `.CAPACITYLIMITRATING` ein⁹⁹. Eine stochastische Auswahl eines Distributionsweges kann durch den Parameter `.PROBABILITY` erreicht werden. Für jeden Distributionsweg kann so die Wahrscheinlichkeit spezifiziert werden, dass dieser Weg ausgewählt wird.

Der Beginn des Transportweges wird durch den Parameter `.OUT` am Werk festgelegt. Jedes Element eines Distributionsweges ist von der Klasse `PROCESS` abgeleitet (wie auch von der Klasse `PLANT`) und besitzt daher ebenfalls ein Attribut `.OUT`, welches die Möglichkeiten für den weiteren Weg der zu transportierenden Güter bestimmt.

6.7.2. Transportdisposition

Den eigentlichen Transport von Fahrzeugen oder Behältern übernehmen *Transportmittel*¹⁰⁰. Die Zeiten, an denen diese Transporte verkehren, können entweder am Bedarf ausgerichtet werden (Transport auf Abruf) oder sich an einem Fahrplan orientieren. Der Transport gemäß Fahrplan wird durch den Parameter `.TIMETABLE` festgelegt. Für den Transport auf Abruf ist eine größere Anzahl Parameter verantwortlich:

1. Der Parameter `.ONREQUEST` aktiviert den Transport auf Abruf.

⁹⁸Der Parameter `.REFERENCE` ist vom Typ »Ganze Zahl« (*int*) und kann nahezu beliebige Teilstücke des Distributionsnetzwerkes als Bezugspunkt auswählen.

⁹⁹Zum Beispiel mit einem negativen Wert, der das Ergebnis der Berechnung der Gütefunktion entsprechend beeinflusst und so die Wahrscheinlichkeit der Auswahl dieses Weges senkt.

¹⁰⁰Siehe Abbildung E.80. Transportmittel werden über Transportdefinitionen (Abb. E.79.) an Logistikdienstleistern (Abb. E.31) definiert.

2. `.REQUESTMINSTOCK` legt den Bestand in einem vorgelagerten Prozess fest, der mindestens vorhanden sein muss, um eine Transportanforderung auszulösen.
3. `.REQUESTDELAY` bestimmt die Zeit zwischen Transportanforderung und Beginn des Transportes.
4. Ein Transportmittel kann aus mehreren Transporteinheiten zusammengesetzt sein. Die Anzahl Einheiten (z.B. Waggons), aus denen ein Transport min. bestehen muss, um gestartet werden zu können, legt der Parameter `.MINUNITS` fest.
5. Die Anzahl der Fahrzeuge bzw. Behälter, die auf einer Transporteinheit bewegt werden können, werden durch den Parameter `.UNITSIZE` bestimmt.
6. Die Kapazität eines Behälters legt das Attribut `.PARTCAPACITY` der Klasse `CONTAINER` fest.
7. Welcher Behälter genutzt werden soll, wird am Attribut `.CONTAINER` der Klasse `SUPPLIERBTO` festgelegt, oder, falls eine Festlegung je Teilevariante gewünscht wird, am gleichnamigen Attribut der Klasse `SUPPLIERPRIORITISATION`. Für Zulieferer vom Typ `SUPPLIERBTS` wird der Behältertyp an Objekten der Klasse `SUPPLIERPARTINFORMATIONOUT` festgelegt.

Ist ein Transport auf Abruf parametrisiert, beginnt dieser Transport, wenn einerseits die erforderliche Anzahl Fahrzeuge bzw. Behälter für einen Transport bereit sind und andererseits, wenn hinreichend viele Transporteinheiten zusammengestellt werden können.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
CDistributionChannel			DISTRIBUTIONCHANNEL	
<i>bool</i>	<code>.OnRequest</code>	D.11	<code>.ONREQUEST</code>	E.21
<i>int</i>	<code>.RequestDelay</code>	D.11	<code>.REQUESTDELAY</code>	E.21
<i>int</i>	<code>.RequestMinStock</code>	D.11	<code>.REQUESTMINSTOCK</code>	E.21
<i>int</i>	<code>.MinUnits</code>	D.11	<code>.MINUNITS</code>	E.21
<i>int</i>	<code>.UnitSize</code>	D.11	<code>.UNITSIZE</code>	E.21
<i>CSchedule</i> 1..1 →	<code>.TransportSchedule</code>	D.11	<code>.TIMETABLE</code>	E.21
CContainerType			CONTAINER	
<i>int</i>	<code>.PartCapacity</code>	D.16	<code>.PARTCAPACITY</code>	E.13
CSupplierPartInfoOut			SUPPLIERPARTINFORMATIONOUT	
<i>int</i>	<code>.ContainerType</code>	D.16	<code>.PARTCONTAINER</code>	E.76
CSupplierPriority			SUPPLIERPRIORITISATION	
<i>int</i>	<code>.ContainerType</code>	D.38	<code>.CONTAINER</code>	E.77

Tabelle 6.19.: Zentrale Parameter Transmitteldisposition.

6.7.3. Progressive und retrograde Terminierung

Die Bestimmung der wahrscheinlichen Transportzeiten im Vorhinein ist ein wichtiger Baustein für die Berechnung des voraussichtlichen Lieferzeitpunktes und für die Berechnung der Liefertreue im Nachhinein. Für die Berechnung eines voraussichtlichen Lieferzeitpunktes ist auf den geplanten Termin zur Fertigstellung eines Fahrzeugs im Werk (siehe Kapitel 6.5.2) eine prognostizierte Durchlaufzeit durch die Distribution zu addieren.

Im vorgestellten Ansatz werden zwei Verfahren zur Berechnung von Durchlaufzei-

ten realisiert. Liegt die Netzbeschreibung in Form von hierarchisch aufgebauten Distributionskanälen mit geringen Auswahlmöglichkeiten zur Bestimmung alternativer Wege vor, wird die Durchlaufzeit abhängig von den Elementen des Distributionsnetzes berechnet. Aus den Objekten der Klassen BUFFER und DISTRIBUTIONCHANNEL werden jeweils die Werte des Attributes .PLANVALUE der Durchlaufzeitverteilung, die durch das Attribut .DELAY der Objekte referenziert wird, als Plandurchlaufzeit¹⁰¹ herangezogen¹⁰².

Werden hingegen Routing-Tabellen zur Bestimmung der Transportwege genutzt, ist es im Vorhinein weniger einfach, eine verlässliche Prognose über die zu erwartende Transportzeit abzugeben. Der konkrete Weg, den ein Fahrzeug bzw. Behälter nimmt, um zum Ziel zu gelangen, hängt z.B. vom Simulationsdatum ab, von der dynamischen Auslastung der Puffer usw. Zu dem Zeitpunkt, zu dem eine Prognose über die Distributionszeit erforderlich ist – z.B. wenn einem Kunden ein Liefertermin genannt werden soll – kann die Belastungssituation im Distributionsnetzwerk eine grundlegend andere sein, als zu dem Zeitpunkt, an dem ein Fahrzeug tatsächlich zum Kunden transportiert werden soll. Eine Wegebestimmung zu dem Zeitpunkt, an dem eine Prognose gefordert ist, führt also nicht verlässlich zu einem guten Prognoseergebnis. Zur Bestimmung der Plandurchlaufzeiten wird daher ein anderer Ansatz gewählt: Für jede Transportrelation kann eine Plandurchlaufzeit spezifiziert werden. Die Spezifikation dieser Plandurchlaufzeiten erfolgt durch Hinzufügen von Objekten vom Typ ROUTEINFORMATION zu einer Liste (Attribut .DISTANCEMATRIX) an einem Objekt vom Typ DISTRIBUTION. Dieses Objekt wiederum ist in einem Simulationsmodell genau einmal vorhanden und wird direkt am Simulationsexperiment definiert.

6.7.4. Distributionsprozesse in der Beschaffung

Die Gestaltung der Beschaffung von Ausgangsmaterialien für die Fahrzeugproduktion bestimmt in nicht unerheblichem Umfang die Eigenschaften eines Auftragsabwicklungsprozesses. Werden Teile, wie z.B. der Kabelbaum, auftragsbezogen in einem weit entfernten Land gefertigt¹⁰³, muss ein Auftrag, verglichen mit einer Fertigung in unmittelbarer Nähe des Verwendungsortes, vergleichsweise früh in den Eigenschaften festgelegt (eingefroren) werden, die Einfluss auf das beauftragte Teil (den Kabelbaum) nehmen. Den Kostenvorteilen durch günstige Löhne steht eine frühe Festlegung des Auftrags und somit eine Einschränkung der Gestaltungsmöglichkeiten der Kunden gegenüber.

Um den Einfluss der damit verbundenen »Sourcing-Entscheidungen« abbilden zu können, müssen dabei sowohl in Folge veränderte Parameter der Auftragseinplanung als auch veränderte Distributionswege vom Lieferanten zu den Werken der Automobilhersteller berücksichtigt werden.

Typ	Attribut (UML)	Abb.	Attribut (GME)	Abb.
	CDistributionSystem		DISTRIBUTION	

¹⁰¹Unter der *Plandurchlaufzeit* soll die prognostizierte Zeit verstanden werden, die ein Fahrzeug bzw. Behälter für den Weg durch einen Prozess bzw. eine Prozesskette benötigt.

¹⁰²Das Attribut wird von der gemeinsamen Basisklasse PROCESS ererbt.

¹⁰³Die Fertigung von Kabelbäumen ist sehr personalintensiv. Daher werden diese häufig in Ländern mit einem »günstigen« Lohnniveau produziert.

CLogisticsServiceProvider 0..n →	.LDL ¹⁰⁴	D.14	.LDL	E.31
CRouteInformation 0..n →	.DistanceMatrix	D.17	.DISTANCEMATRIX	E.20
CLogisticsServiceProvider		LOGISTICSSERVICEPROVIDER		
CLSPRoutes 0..n →	.LSPRoutes	D.14	.LSPROUTES	E.31
CTransportDefinition 0..n →	.TransportDefinition	D.14	.TRANSPORTDEFINITION	E.31
CTransportDefinition		TRANSPORTDEFINITION		
CTransport 1..1 →	Transport	D.15	.TRANSPORT	E.79
CLSPRoute		LSPROUTE		
CBuffer 1..n →	.Buffer	D.14	.BUFFER	E.32
CSchedule 1..1 →	.TransportSchedule	D.14	.TRANSPORTSCHEDULE	E.32
CPlant		PLANT		
CBuffer 1..1 →	.IN	D.29	.INBUFFER	E.42
CSupplier		SUPPLIER		
CBuffer 1..1 →	.OUT	D.35	.OUTBUFFER	E.71

Tabelle 6.20.: Zentrale Parameter Distribution Beschaffung.

Die Modellierung von Distributionsnetzwerken zwischen Werken von Automobilherstellern und Lieferanten kann wiederum auf zwei Arten erfolgen. Die Auswahl zwischen diesen Beschreibungsmethoden kann davon abhängig gemacht werden, inwieweit die Transportrouten, z.B. von Gebietsspediteuren¹⁰⁵, für die Beantwortung der Fragestellung relevant sind. Ist dies nicht der Fall, kann zwischen den Lieferanten und den Werken ein Transportnetz in analoger Weise zur Fahrzeugsdistribution definiert werden. Der Transportweg beginnt im Puffer, der durch das Attribut `.OUTBUFFER` an der Klasse `SUPPLIER` bzw. den abgeleiteten Klassen `SUPPLIERBTO` und `SUPPLIERBTS` definiert wird und endet im Puffer `.INBUFFER` definiert am »Werk« (`PLANT`).

Etwas komplexer ist die Situation, wenn Logistikdienstleister (definiert am Attribut `.LOGISTICSSERVICEPROVIDER` an der Klasse `DISTRIBUTION`) im Fokus der Betrachtung stehen. Da die Zahl der möglichen Transportrouten zwischen den Lieferanten eines Gebietes sehr groß sein kann, wird hier die Möglichkeit geboten, die Distributionskanäle allein durch Angabe der Entfernung zwischen zwei Punkten zu definieren. Das Attribut `.DISTANCEMATRIX` der Klasse `DISTRIBUTION` erlaubt die Spezifikation von Objekten vom Typ `ROUTEINFORMATION`, in denen Angaben über die Entfernung zwischen zwei Punkten und die dafür benötigte Zeit abgelegt werden können. Auf Basis dieser Informationen kann das Simulationswerkzeug selbsttätig Distributionskanäle erzeugen und für den Transport der Güter nutzen.

Das Attribut `.LSPROUTES` der Klasse `LOGISTICSSERVICEPROVIDER` erlaubt nun die Spezifikation von Transportrouten allein durch Aufzählung der Beladestellen¹⁰⁶, die angefahren werden sollen (Attribut `.BUFFER`). Die Zeitpunkte für diese Transporte werden am Attribut `.TRANSPORTSCHEDULE` definiert. Das Simulationswerkzeug

¹⁰⁴Tatsächliche Bezeichnung: `LogisticsServiceProvider`

¹⁰⁵In diesem Zusammenhang Spediteure, die die Güter einer Reihe von Zulieferern eines Gebietes in ein oder mehrere Werke eines Automobilherstellers liefern [Fle02, A1-15].

¹⁰⁶Die gegenwärtige Implementierung ist beschränkt auf genau eine Abladestelle (z.B. ein Werk). Eine Erweiterung der Modellierungsmöglichkeiten um zusätzliche Abladestellen ist leicht möglich, um eine Weiterentwicklung des Bewertungswerkzeugs zu unterstützen.

kann durch Ermittlung eines »passenden« ROUTEINFORMATION-Objektes die Entfernung und die Transportzeit bestimmen und somit Distributionskanäle für alle Teilstücke zwischen den angefahrenen Puffern erzeugen, eine wesentliche Erleichterung der Modellierungsaufgabe, verglichen mit der manuellen Spezifikation von unter Umständen hunderten von Distributionskanälen.

Kann für einen Weg zwischen zwei Punkten kein ROUTEINFORMATION-Objekt gefunden werden, wird die Distanz auf Basis der Informationen über geographische Länge und Breite der jeweiligen Puffer bestimmt. Als Geschwindigkeit für den Transport wird der Mittelwert aller im Objekt des Typs DISTRIBUTION spezifizierten Geschwindigkeiten ermittelt (siehe E.20).

Die von den Lieferanten hergestellten Teile werden nach Fertigstellung einer für die Befüllung eines Behälters (Typ CONTAINER) ausreichenden Anzahl in dem bereits erwähnten »OutBuffer« abgelegt. Die Fahrzeuge des Logistikdienstleisters werden in Ausführung (Typ TRANSPORT) und Anzahl an einem Objekt vom Typ LOGISTICS-SERVICEPROVIDER definiert (Attribut .TRANSPORTDEFINITION).

Die Anzahl der Behälter, die ein Transportmittel aufnehmen kann, wird auf Basis des Volumens der Behälter ermittelt. Die Abmessungen sowohl der Behälter (siehe Abbildung E.13) als auch der Transportmittel (siehe Abbildung E.13) werden hierfür herangezogen. Da das Berechnungsverfahren keine Packoptimierung durchführt und optimale Raumausnutzung des Transportmittels angenommen wird, ist das verfügbare Volumen der Transportmittel abhängig von den verwendeten Behältern entsprechend (z.B. auf Basis einer »historisch« ermittelten Raumausnutzung) nach unten zu korrigieren.

6.7.5. Zusammenfassung

Das dargestellte Schema zur Abbildung von Distributionskanälen ist aufgrund der allgemein gehaltenen Form der Parametrisierung sowohl für die Modellierung der Transportwege von Lieferanten zu den Werken von Automobilherstellern als auch für die Abbildung der Transportwege für gefertigte Fahrzeuge von den produzierenden Werken zu den jeweiligen Abnehmern in den Märkten geeignet. Es stehen dabei zwei verschiedene Schemata zur Verfügung: Zum einen die Unterteilung von Distributionswegen in Unterkanäle, wobei Transportmittel und Transportzeiten einfach abgebildet werden können, aber eine dynamische Wegfindung nicht umgesetzt werden kann.

Der zweite Ansatz zur Abbildung von Distributionsnetzen verzichtet auf die übergreifende Kennzeichnung von Transportrelationen und erfordert die Spezifikation von Regeln, die an Verzweigungen den weiteren Weg der zu transportierenden Güter beschreiben. Die vielen Parametrisierungsmöglichkeiten für solche Regeln erlauben die Abbildung auch komplexer Regeln für die Routenfindung in Transportnetzwerken.

6.8. Modellelemente im beispielhaften Zusammenspiel

Die Parametrisierung der Modellelemente eines OTD-NET-Modells bestimmt das »Verhalten« der Objekte im Zusammenspiel. Der Rahmen hierfür wird durch eine Infrastruktur gebildet, die grundlegende Dienste für die Interaktion der Modellelemente

zur Verfügung stellt. Eine beispielhafte Darstellung ausgewählter Aktionen der Modellelemente soll das Zusammenspiel verdeutlichen¹⁰⁷.

Vor Beginn der Simulation – es liegt noch keine definierte Simulationszeit vor – werden langfristige Prognosen an Werke und Zulieferer kommuniziert. Eine Kapazitätsanpassung auf Basis dieser Prognosen stellt sicher, dass die Kapazitäten der Werke bzw. Zulieferer »realistisch« eingestellt sind.

Wird nun ein bestimmter Zeitraum innerhalb des definierten Simulationszeitraumes betrachtet – die Simulationszeit ist nun definiert –, wird für diesen Zeitraum zunächst eine weitere Prognose abgegeben (z.B. vier Monate vor diesem Zeitraum), die Basis ist für die *Programmplanung*. Im Rahmen der Programmplanung wird festgelegt, welche Anzahl von Fahrzeugen im betrachteten Zeitraum produziert werden sollen. Weiterhin werden auf Basis der Prognose die Abnahmeverpflichtungen der Händler berechnet¹⁰⁸. Diese Abnahmeverpflichtungen sind ein wichtiges Charakteristikum etablierter Prozesse der Automobilindustrie. Die Hersteller produzieren entsprechend der Programmplanung und verpflichten die Händler auch dann Fahrzeuge abzunehmen, wenn diese keine Kunden hierfür haben (siehe Kapitel 2.2.1).

Die Prognose geht also auf Seiten der Automobilhersteller (PLANNING¹⁰⁹) in die Programmplanung ein (PLANT, PLANNING). Der Hersteller (DEALER, IMPORTER) legt auf Basis der Prognose mit dem Handel eine Abnahmeverpflichtung fest. Weiterhin werden die prognostizierten Werte an die Zulieferer (SUPPLIERBTO, SUPPLIERBTS) in der Simulation kommuniziert. Die Zulieferer adaptieren erforderlichenfalls auf Basis der Prognose ihre Kapazitäten (von RESOURCE abgeleitete Klassen).

Erscheint in der Simulation ein Kunde (CUSTOMERTYPE) bei einem Händler, kann dieser zunächst in einem Pool von bereits gefertigten Fahrzeugen nach einem passenden Fahrzeug suchen. Diese Suche erstreckt sich (optional) auch auf Lagerfahrzeuge bei anderen Händlern. Aus nahe liegenden Gründen ist die Vermittlung bereits gefertigter Fahrzeuge von vordringlichem Interesse sowohl für den Automobilhersteller als auch für den Händler. Kann kein Lagerfahrzeug mit passender Konfiguration bzw. Ausstattung gefunden werden, kann der Händler versuchen, einen Auftrag, den der Händler im Rahmen seiner Abnahmeverpflichtung erteilt hat, in seiner Konfiguration an den Kundenwunsch anzupassen.

Dies ist i.d.R. dann möglich, wenn der geplante Fertigungszeitpunkt noch soweit in der Zukunft liegt, dass der Hersteller noch keine Vorkehrungen für die Fertigung genau dieser Konfiguration getroffen hat (z.B. Bestellung eines passenden Kabelbaumes). Ist ein Auftrag angepasst bzw. neu erteilt, kann dem Kunden ein voraussichtlicher Liefertermin genannt werden¹¹⁰.

Rückt das betrachtete Intervall näher und ist noch kein Werk (PLANT) bestimmt¹¹¹,

¹⁰⁷Zur Vereinfachung wird die Terminologie des abgebildeten Systems verwendet, es ist also z.B. von Händlern und Kunden die Rede, wenn von den entsprechenden Objekten in der Simulation gesprochen werden müsste.

¹⁰⁸Wie die meisten der hier geschilderten Abläufe ist auch die Berechnung von Abnahmeverpflichtungen für die Händler optional. Das heißt, es ist auch möglich, die entsprechenden Parameter so zu setzen, dass die Händler nur Fahrzeuge bestellen, wenn sie Kunden hierfür haben.

¹⁰⁹Die in dem jeweiligen Zusammenhang relevanten GME-Klassen werden an einigen Stellen zur Verdeutlichung genannt.

¹¹⁰Teilweise werden vom Hersteller nur die Termine für das voraussichtliche Fertigungsende genannt. Es obliegt dann dem Händler einen Liefertermin auf dieser Basis zu nennen.

¹¹¹In vielen Fällen werden Baureihen in nur einem Werk gefertigt – hier erfolgt die Werkzuordnung

in welchem das Fahrzeug gefertigt werden soll, ist dies ein möglicher weiterer Schritt in der Auftragsbearbeitung.

Der nächste Schritt in diesem beispielhaften Ablauf ist die Festschreibung des Auftrages – der Auftrag wird *eingefroren*. Mit dem Einfrieren des Auftrags verlieren die Händler bzw. Kunden die Möglichkeit, die Konfiguration des Auftrags zu ändern und der Hersteller gewinnt die Möglichkeit, für diesen Auftrag Vorkehrungen für die Fertigung zu treffen, da die benötigten Teile nun feststehen.

Aus der Menge der eingefrorenen Aufträge wird vom Hersteller die Fertigungssequenz für einen Tag bestimmt. Die Festlegung dieser Sequenz kann dabei bereits einige Tage vor der eigentlichen Fertigung durchgeführt werden. Im Zuge der Sequenzfestlegung wird im Rahmen des vorgestellten Ansatzes die effektive Verfügbarkeit aller für die Aufträge benötigten Teile geprüft. Für die Fertigungssequenz werden nur solche Aufträge zugelassen, für die Kapazitäten vorhanden sind. Ein Zuliefererengpass verändert also die Sequenz auch in der Simulation. Neben der Kapazität der Lieferanten wird auch die tatsächliche Fertigungskapazität des Werkes geprüft.

Die eigentliche Fertigung wird durch eine Verkettung von Prozesselementen modelliert, die lediglich die Durchlaufzeiten durch die jeweiligen Teilprozesse der Fertigung (z.B. *Rohbau*, *Lack*, *Montage*) abbilden. Die Fahrzeuge, die entsprechend den Aufträgen aus dem betrachteten Zeitabschnitt gefertigt wurden, stehen in einer veränderten Reihenfolge nach dem jeweiligen Fertigungsende zum Abtransport bereit.

Die Fahrzeuge werden nun nach Zuordnung eines Transportweges (DISTRIBUTIONCHANNEL) mit den für die jeweiligen Relationen vorgesehenen Transportmitteln (TRANSPORTER) zu den Händlern transportiert. Die Ankunft des Fahrzeugs beim Händler markiert das Ende einer wichtigen Messstrecke für die Beurteilung der Leistungsfähigkeit des Auftragsabwicklungsprozesses: die *Auftragsdurchlaufzeit*. Nach der Ankunft des Fahrzeugs beim Händler wird das Fahrzeug dem Kunden übergeben und damit die Auftragsbearbeitung, zumindest im Betrachtungsrahmen dieser Arbeit, abgeschlossen.

implizit mit der Auftragsannahme durch den Hersteller.

7. Systemarchitektur

Ein wichtiges Entwurfsziel für die im obigen Unterkapitel 6.1 dargelegten Konzepte zur Modellierung von Auftragsabwicklungsprozessen ist es, eine effiziente Bewertung solcher Modelle zu ermöglichen. In Kapitel 5 wurde die diskrete, ereignisorientierte Simulation als Methode für eine solche Bewertung ausgewählt. Im folgenden Kapitel werden wesentliche Aspekte der konkreten Umsetzung des Bewertungswerkzeugs vorgestellt. Auf eine umfassende Beschreibung aller Details der Umsetzung des Bewertungswerkzeugs soll hier verzichtet werden, da z.B. die Umsetzung der »Kernfunktionen«¹ eines Werkzeuges zur diskreten, ereignisorientierten Simulation in der Literatur bereits häufig beschrieben wurde (vgl. z.B. [SB02]). Im Folgenden wird zunächst die Infrastruktur für die Implementierungsaufgabe festgelegt.

Viele der für diese Infrastruktur ausgewählten Elemente, die für die Umsetzung des Bewertungswerkzeugs eingesetzt werden, lassen sich durch Alternativen ersetzen; die gewählte »Umgebung« des Bewertungswerkzeugs ist keinesfalls die einzig mögliche. In der Grundausrichtung wird hier im Zweifel »konservativ« ausgewählt, also z.B. C++ anstatt neuerer Programmiersprachen wie C# oder Java, SQL-Datenbanken anstatt objektorientierter Datenbanken. Eine Beschränkung im Hinblick auf den Entwurf des Bewertungswerkzeugs ist mit der getroffenen Auswahl aber nicht verbunden, da die Implementierungsumgebung die erforderlichen Konzepte² der objektorientierten Programmierung unterstützt.

Auf Basis der Anforderungen, die in Kapitel 4 formuliert wurden, der in Kapitel 5 gefundenen Bewertungsmethode und der gewählten Implementierungssprache können Festlegungen für die konkrete Umsetzung getroffen werden:

- ▷ **Speicherung von Modellen und Ergebnissen (Infrastruktur).** Modelle und auch die Ergebnisse von Simulationen werden in einer *Datenbank*³ gespeichert. Aufgrund der hohen Verbreitung der Technologie soll ein »relationales Datenbanksystem«⁴ (RDBS) verwendet werden, welches die Abfragesprache SQL⁵ unterstützt.
- ▷ **Speicherung von Modellen (Format).** Die in der Implementierungsphase des Bewertungswerkzeugs mit hoher Wahrscheinlichkeit auftretenden Änderungen der Klassen⁶, die Modellelemente abbilden, ziehen Änderungen in der Tabellenstruk-

¹Die Verwaltung von Ereignissen in der zeitlichen Reihenfolge, in der diese in der Simulation auftreten usw.

²Im engeren Sinne könnte der vorgestellte Ansatz auch ohne Nutzung einer objektorientierten Programmiersprache umgesetzt werden, allerdings mit einem weitaus höheren Aufwand.

³Eine *Datenbank* ist ein auf Dauer angelegter, strukturierter Datenbestand (vgl. [SGR97, S. 165]).

⁴Ein Datenbanksystem, welches insbesondere die Daten in Form von Relationen repräsentiert. Datenbanksysteme, die Daten in Form von Tabellen speichern, werden häufig als RDBS kategorisiert (siehe [SGR97, S. 620], [DD97, S. 9]).

⁵SQL stand in der Vergangenheit für »Structured Query Language«, ist aber in der standardisierten Fassung SQL/92 nur mehr ein Name, kein Akronym (vgl. [DD97, S. 3]).

⁶Erfahrungen aus der Implementierungsphase des Werkzeuges können auf das Klassendesign zurückwir-

tur zur Speicherung dieser Modellelemente nach sich, wenn die Klassenstruktur »direkt« in die Tabellenstruktur überführt wird. Um diese Änderungen zu vermeiden, wird eine »objektrelationale« Struktur gewählt, die alle Klassen auf eine kleine Anzahl Tabellen abbildet. Die Klassenstruktur wird ebenfalls in objektrelationaler Form in eigenen Tabellen verwaltet.

- ▷ **Speicherung von Ergebnissen (Format).** Die zu erwartenden großen Datenmengen bei den Simulationsergebnissen führen zu der Entscheidung, hier das relationale Modell ohne die oben erwähnte objektrelationale Struktur zu verwenden.
- ▷ **Bearbeitung von Modellen.** Die durch die obige Festlegung auf eine objektrelationale Speicherung fixierte Tabellenstruktur mit jeweils drei Tabellen für die Objekte und drei Tabellen für die Klassen erlaubt es, ein Modellierungswerkzeug entkoppelt vom konkreten Objektlayout zu entwickeln. Das Modellierungswerkzeug arbeitet auf »Objekten« und kann die Klassenstruktur aus den hierfür erzeugten Tabellen entnehmen.
- ▷ **Infrastruktur für die Auswertung von Ergebnissen.** Für die Gewinnung von Kennzahlen werden zwei Methoden genutzt: Die Abfragesprache SQL (siehe oben) und die mehrdimensionale Analyse (OLAP).
- ▷ **Modell- und Auswertungsbearbeitung.** Werkzeuge für die Modellbearbeitung (»GME« – Graphical Modeling Environment) und für die Auswertungserstellung und Verwaltung (»OTD-Analyzer«) wurden im Fraunhofer IML⁷ u.a. mit dem Ziel der Bearbeitung von OTD-NET Modellen und der Analyse der Simulationsergebnisse entwickelt und werden hier für Modellerstellung und Analyse eingesetzt.
- ▷ **Simulationssoftware.** Eine Nutzung vorhandener Software, wie z.B. der Open-Source Simulationsumgebung »C++-Sim«⁸, wurde vermieden, weil der Anpassungsaufwand für externe Softwarepakete höher als der potenzielle Nutzen eingeschätzt wurde⁹.

Die obige Auswahl bedarf einer weiteren Konkretisierung, die nach einer Bestimmung der Betriebssystemplattform, welche im Weiteren genutzt wird, erfolgen kann.

- ▷ Als Betriebssystem wird Microsoft Windows gewählt. Die Portierung des Quellcodes des Bewertungswerkzeugs auf das Betriebssystem Linux wird in der Umsetzung vorgesehen¹⁰, aber an dieser Stelle nicht durchgeführt.
- ▷ Das Datenbanksystem wird ebenfalls von Microsoft geliefert: Microsoft SQL-Server 2000. Die Verbindung zur Datenbank wird auf der Basis von ODBC (Open

ken (vgl. [Gus02, S. 4]).

⁷Nicht im Rahmen dieser Arbeit.

⁸Siehe <http://cxxsim.ncl.ac.uk/>.

⁹So ist z.B. der Anteil des Codes des eigentlichen »Kerns« des Bewertungswerkzeugs kleiner als 2 % des Gesamtumfangs. Das heißt, der größte Teil des Implementierungsaufwands entfällt auf die Umsetzung der spezifischen Teile für die Simulation von Auftragsabwicklungsprozessen.

¹⁰Dieser Faktor spielt z.B. in der Auswahl von »Basis-Bibliotheken« eine Rolle, die genutzt werden, um Klassen wie Listen und Vektoren einsetzen zu können, ohne diese selbst zu programmieren. Es werden keine spezifischen Microsoft Bibliotheken verwendet, wie z.B. MFC (Microsoft Foundation Classes) oder Klassen aus der .NET Umgebung.

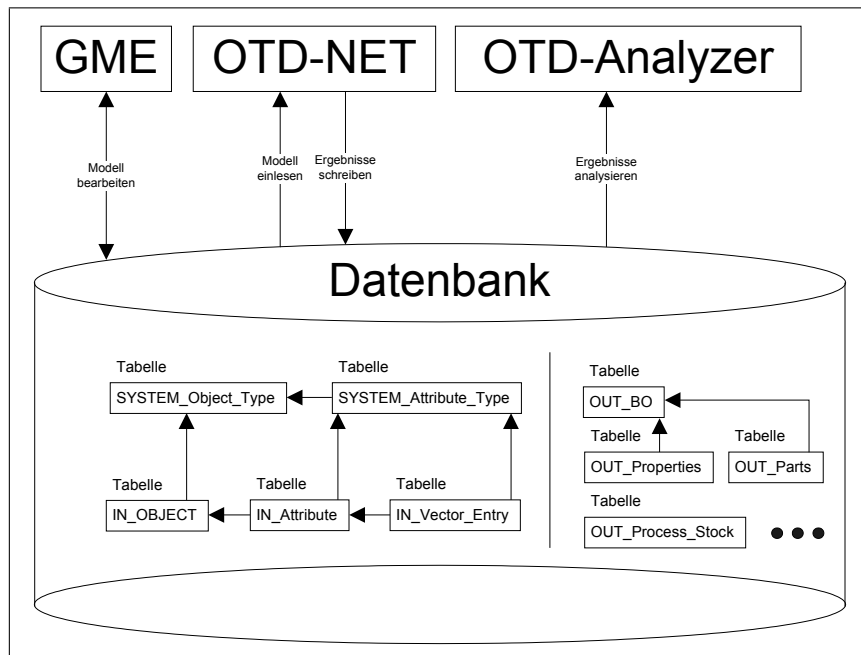


Abbildung 7.1.: Gesamtsystem zur Modellerstellung, Bewertung und Analyse

Database Connectivity) realisiert¹¹.

- ▷ Die Erstellung von UML-Diagrammen und die Erzeugung von Source-Code auf Basis dieser Diagramme wird mittels Rational Rose 2001 durchgeführt.
- ▷ Für die Übersetzung des C++-Codes in ausführbare Programme¹² wird Microsoft Visual Studio 2005 genutzt.

In den folgenden Unterkapiteln werden die konkrete Umsetzung der Speicherung der Modelle, das eigentliche Bewertungswerkzeug »OTD-NET« sowie die Möglichkeiten zur Analyse der Ergebnisse dargestellt.

7.1. Modell- und Ergebnisrepräsentation

7.1.1. Eingangsdaten

Die in Kapitel 6 beschriebenen Objekte müssen in der Phase der Modellerstellung erzeugt und parametrisiert werden, um im Anschluss als Eingangsdaten für das Be-

¹¹Microsoft bietet ein umfangreiches »Sammelsurium« von Zugriffsmöglichkeiten auf Datenbanken unter dem Dach MDAC (Microsoft Data Access Components), wie z.B. ADO, ADO.NET, OLE DB. Diese Protokolle bauen auf ODBC auf, bieten aber selbst keine effizienten Möglichkeiten zur Verarbeitung von Massendaten (Bulk operations). »Bulk-Inserts« werden in der Implementierung des Bewertungswerkzeugs intensiv genutzt, um die sehr großen entstehenden Datenmengen in die Datenbank zu schreiben.

¹²Diese Programme sind ohne Weiteres nur unter einigen Betriebssystemen von Microsoft (z.B. »Windows XP«) ausführbar. Eine Ausführbarkeit unter Linux lässt sich unter Verwendung eines anderen Compilers (z.B. »GCC«, siehe <http://www.gnu.org>) erreichen.

wertungswerkzeug zu dienen. Eine Möglichkeit der Repräsentation der Objekte auf die Tabellen einer relationalen Datenbank besteht darin, jeweils eine Klasse auf eine oder mehrere Tabellen abzubilden. So könnten z.B. Tabellen für Objekte der Klassen SUPPLIER und PLANT erzeugt werden, wobei eine Reihe von Entwurfsregeln zu beachten sind, um inkonsistente Daten durch Redundanzen zu vermeiden (vgl. [DF92, S. 465-466]).

Ein wesentlicher Gestaltungsaspekt des hier vorgestellten Ansatzes ist es, Erweiterungen der Abbildungsmöglichkeiten durch Hinzufügen oder Erweitern von Klassen bzw. Algorithmen leicht zu ermöglichen. Mit einer solchen Anforderung ist das Erfordernis der Anpassung des Datenbankschemas bei jeder Klassenänderung nicht vereinbar, insbesondere weil eine Änderung des Datenbankschemas häufig eine Anpassung des Werkzeugs zur Bearbeitung von Simulationsmodellen nach sich ziehen wird.

Durch das Konzept einer »objektrelationalen« Speicherung von Objekten können diese Nachteile vermieden werden. Die in Abbildung 7.2 dargestellte Tabellenstruktur ermöglicht es, ein Objekt und seine Attribute in lediglich drei Tabellen abzubilden. Jedes Objekt wird durch einen Eintrag in der Tabelle IN_Object und etwaige Einträge in die Tabellen IN_Attribute für 1 : 1 Attribute (ein Attributwert je Objekt) und Einträge in die Tabelle IN_Vector_Entries für 1 : n Attribute (mehrere Attributwerte möglich) abgebildet.

Die Klassenstruktur wird in zwei weiteren Tabellen festgehalten: SYSTEM_Object_Type und SYSTEM_Attribute_Type. Jede Klasse wird mit einem Eintrag in der Tabelle SYSTEM_Object_Type und n Einträgen in der Tabelle SYSTEM_Attribute_Type abgebildet¹³. Hinzukommen einige Tabellen¹⁴ für Zugriffberechtigungen eines Benutzers auf Objekte oder Attribute auf der Basis von »Access Control Lists« (Tabelle SYSTEM_OMW_ACL), für die Abbildung von Typnamen und Beschreibungen der Klassen (Tabelle SYSTEM_Stringtable) und für die Benutzerverwaltung (Tabelle SYSTEM_OMW_Role).

Diese Tabellenstruktur ermöglicht es, sowohl ein Werkzeug für die Bearbeitung von Modellen als auch ein Werkzeug zum Einlesen eines Modells in den Hauptspeicher eines Computers zu bauen, ohne die konkrete Typstruktur im Vorhinein kennen zu müssen. Klassenänderungen können so ohne Anpassung des Modelleditors und mit sehr kleinen Anpassungen am Simulationswerkzeug durchgeführt werden. Soll z.B. ein neues Attribut .WEIGHT eingelesen werden, ist hierfür lediglich eine Zeile C++-Code erforderlich (das Objekt »dbo« der Klasse CDBClass wird bereits als deklariert angenommen):

```
double weight = dbo->GetFloatValue("Weight");
```

Das heißt, eine Typänderung erfordert in dieser Struktur nur minimalen Aufwand, es kann auf die Algorithmen zur Verarbeitung etwaiger neuer Daten fokussiert werden.

7.1.2. Ergebnisse

Die Festlegung der für die Bestimmung von Kennzahlen erforderlichen Eingangsdaten folgt aus dem Berechnungsverfahren für die Kennzahlen. Im vorliegenden Fall kann

¹³In dieser Tabelle werden Attribute sowohl für 1 : 1 Relationen als auch 1 : n Relationen beschrieben.

¹⁴Nur teilweise in Abbildung 7.2 enthalten.

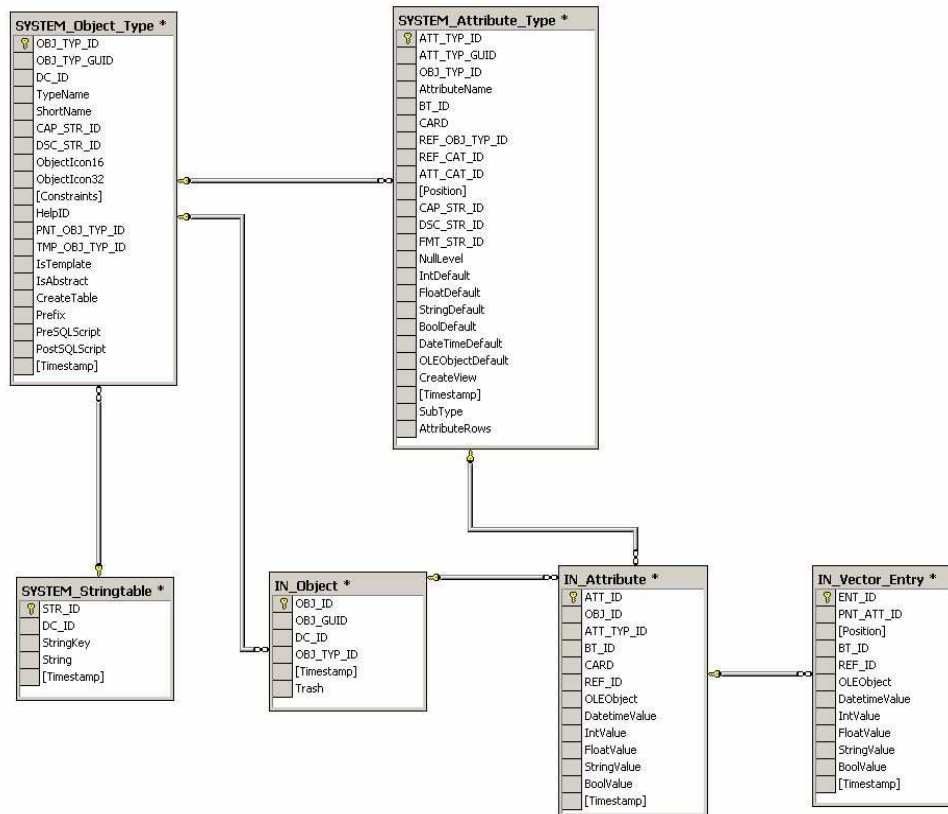


Abbildung 7.2.: Objektorientierte Tabellenstruktur

nicht von einer bereits festliegenden Menge von Kennzahlen und in Folge einer Menge von benötigten Eingangsdaten ausgegangen werden, da der Zweck der Erhebung von Kennzahlen – die Fragestellung – nicht im Vorhinein bekannt ist. Um für möglichst viele Fragestellungen eine Analysegrundlage geben zu können, ist eine entsprechend große Anzahl von Grunddaten bereitzustellen. Die Basis für diese Bereitstellung sind zum einen die Daten, die von im zu analysierenden Modell beschriebenen Objekten erzeugt werden¹⁵ (z.B. Händler, Werke, Zulieferer) und zum anderen Daten, die von Objekten erzeugt werden, die im Zuge der Bewertung eines Modells durch das Simulationswerkzeug generiert werden (z.B. Prognosen, Aufträge, Fahrzeuge).

Während die Beschreibung des Modellierungsschemas mit nur wenigen Bezügen zum Simulationswerkzeug auskommt¹⁶, welches die Bewertung der Modelle unterstützt und somit den Gestaltungsspielraum für die Implementierung dieses Werkzeugs groß lässt, ist für die Beschreibung der Ergebnisdaten eine stärkere Festlegung auf die Details der Implementierung erforderlich. Die sehr große Zahl von Ergebnissen, die vom Simulator zur Verfügung gestellt werden, lässt eine ausführliche Beschreibung

¹⁵Konkret von den Objekten im Simulator, welche auf Basis der Objekte im Simulationsmodell instantiiert werden.

¹⁶Ein Modell eines Auftragsabwicklungsprozesses, welches nach den oben beschriebenen Grundsätzen gestaltet wurde, steht für sich und kann prinzipiell von verschiedensten Werkzeugen für eine Bewertung genutzt werden.

aller Ergebnisse an dieser Stelle allerdings nicht zu. Daher wird hier nur eine Auswahl der zur Verfügung gestellten Ergebnisse präsentiert. Eine vollständige Beschreibung der vom Prototypen erzeugten Ergebnisse einschließlich einer kurzen Beschreibung ist im Anhang F aufgeführt.

Die während der Simulation erzeugten Daten¹⁷ umfassen (u.a.):

- ▷ **Märkte.** Kundenbestellung (Zeitpunkt, Ausstattungswünsche), Händlerprognosen, Händlerbestellungen, Auftragsdaten (gewünschte Eigenschaften, gelieferte Eigenschaften).
- ▷ **Werke.** Anzahlen eingeplanter Aufträge, Kapazitäten, Versorgungssituation Teile.
- ▷ **Lieferanten.** Kapazitäten, Liefererfüllung, Versorgungssituation Teile.

Für die Simulationsergebnisse wurde ein anderes, als das oben für die Eingangsdaten beschriebene Format gewählt, um eine effiziente Analyse der Daten zu ermöglichen. Um z.B. den Wert eines Attributes (z.B. `.WEIGHT`) eines Objektes in den Eingangsdaten zu bestimmen, ist zunächst einmal durch einen Blick in die Tabelle `IN_Object` die ID (eindeutiger Identifizierer) des fraglichen Objektes und die Objekt-Typ-ID dieses Objektes zu bestimmen. Im Anschluss kann in der Tabelle `SYSTEM_Attribute_Type` die Attribut-Typ-ID des fraglichen Attributes gesucht und im nächsten Schritt unter Nutzung der Objekt-ID und der Attribut-Typ-ID das Attribut aus der Tabelle `IN_Attribute` (es sei ein 1 : 1 Attribut) selektiert werden. Für die Bearbeitung einzelner Attribute ist dieser Aufwand vernachlässigbar. Werden jedoch Daten über viele Millionen Attribute¹⁸ aggregiert, führt dieser Ansatz der Datenrepräsentation zu enorm langen Berechnungszeiten in der Analysephase.

7.2. Das Bewertungswerkzeug

Die »innere« Strukturierung des Bewertungswerkzeugs ist stark an die Struktur der GME-Klassen angelehnt. Abbildung 7.3 zeigt rechts die Struktur eines konkreten Simulationsmodells, in der Mitte die Komponentensicht der UML-Darstellung und rechts die Struktur der »Solution« im C++-Entwicklungssystem. Die Strukturierung des Simulationsmodells ist mit seiner Unterteilung in die Kategorien »Produktbeschreibung«, »Werke«, »Zulieferer« sowie »Märkte« offenbar stark am Anwendungsbereich ausgerichtet. Diese Kategorien finden sich ebenso in der UML-Komponentensicht wieder, die die Strukturierung der abstrakten Meta-Darstellung wiedergibt, die eng mit der Strukturierung des C++-Entwicklungsprojektes verknüpft ist.

7.2.1. Struktur und Ablauf

Die Struktur der linken Ansicht in Abbildung 7.3 kann sehr leicht verändert werden. Die Strukturierung wird durch die Festlegung einer Modellwurzel – hier ein singuläres

¹⁷Siehe Anhang F.

¹⁸Allein ein Eintrag in der Tabelle `OUT_BO` enthält über 80 Attribute, die zu einer entsprechenden Anzahl Einträge in einer fiktiven Tabelle `OUT_Attribute` führen würden.

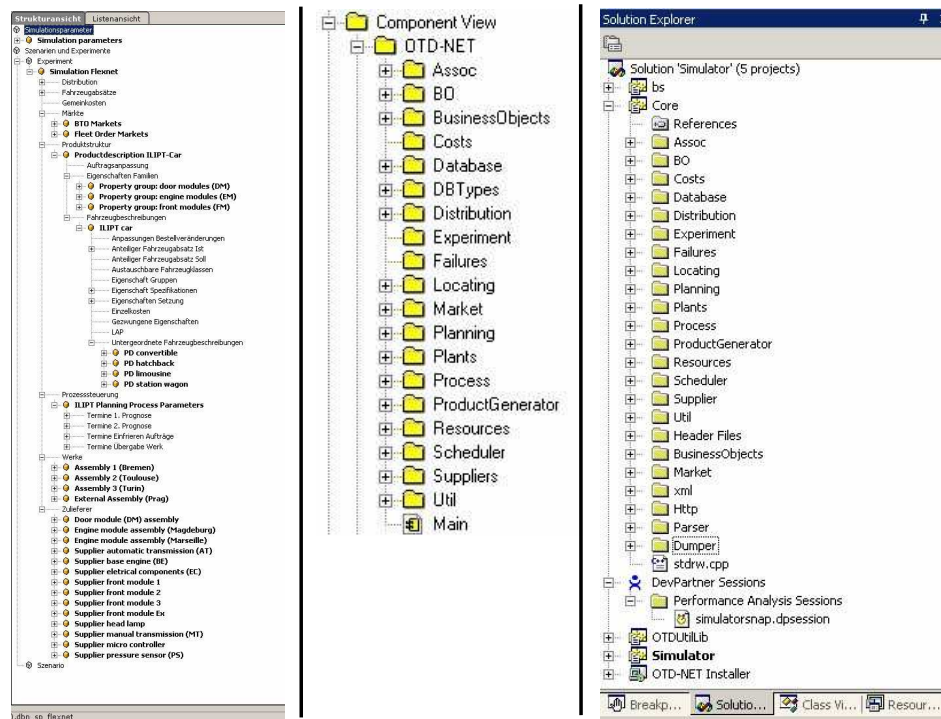


Abbildung 7.3.: Strukturierung Simulationsmodell, UML-Komponenten und Simulationstool

Objekt¹⁹ vom Typ EXPERIMENT – und die Referenzierung z.B. des Produktaufbaus (singuläres Objekt vom Typ PRODUCTSTRUCTURE) erreicht. Eine Umordnung dieser Struktur ist aus Sicht des Modellierungswerkzeugs leicht möglich und erfordert in diesem Werkzeug keine Anpassungen.

Eine Änderung der UML-Komponentenstruktur ist bereits erheblich aufwändiger. Die Ursache hierfür ist, dass die UML-Diagramme bereits in erheblichem Umfang »Wissen« aus dem Anwendungsbereich kodieren, so ist z.B. eine Prämisse bei der Gestaltung der UML-Diagramme, dass in einem Auftrag immer genau eine Eigenschaft aus einer Eigenschaftenfamilie auftritt. In noch weitaus stärkerem Umfang ist dies der Fall für den C++-Code, der neben der statischen Klassenstruktur die Regeln für die dynamische Interaktion der Objekte enthält. Hier ist die Strukturierung nur aufwändig änderbar, da die Regeln für die dynamische Interaktion der Objekte an diese Strukturierung angepasst werden müssen.

Die Relationen zwischen den Strukturen in Abbildung 7.3 sind insofern von Bedeutung, als das eine Änderung bzw. Erweiterung des Modellierungsansatzes eine Beachtung dieser Zusammenhänge erfordert. Im Folgenden soll die Vorgehensweise bei einer »einfachen« Erweiterung²⁰ der Modellparameter beschrieben werden.

Soll z.B. für bestimmte Produkte eine beschleunigte Auftragsabwicklung erreicht

¹⁹Objekte einiger Typen dürfen in einem Modell genau *einmal* vorkommen, typischerweise sind das die Wurzeln von Beschreibungshierarchien wie z.B. Produktaufbau oder Distribution.

²⁰*Einfach*, weil die Strukturen der Modelle erhalten bleiben.

werden, kann dies beispielsweise dadurch erreicht werden, dass bei der Werkseinplanung die verwendete Gütefunktion für diese Produkte so beeinflusst wird, dass eine termintreue Auslieferung im Vergleich zu anderen Produkten stärker Eingang findet. Die Schritte zur Umsetzung eines solchen »Features« sehen wie folgt aus:

1. Erzeugung eines neuen Attributes `.TIMINGPRIO` an der Klasse `PRODUCT-CLASS`.
2. Erzeugung eines neuen Attributes `.TimingPrio` an der Klasse `CBOClass`.
3. Hinzufügen von Code für das Einlesen des Attributes.
4. Änderung der Gütefunktion für die Werkseinplanung.

Das Hinzufügen eines Attributes zu einer Klasse kann »direkt« auf Tabellenebene durch einfache Manipulation der Tabellen `SYSTEM_Attribute_Type`, `SYSTEM_StringTable` und `SYSTEM_OMW_ACL` erfolgen²¹. Wie aber schon in Kapitel 6.2.2 angedeutet, kommen auch für die Definition der GME-Klassen Techniken der objektorientierten Programmierung zum Einsatz, es können also Vererbungshierarchien zur Definition der GME-Klassen genutzt werden. Diese objektorientierte Beschreibung der Klassen ist im Anhang E für viele GME-Klassen aufgeführt.

In dieser OO-Beschreibung wird z.B. das Attribut `.NAME` nur genau einmal aufgeführt, nämlich in der Definition der Klasse `SIMOBJEKT`. Diese Klasse ist die Basis-Klasse vieler GME-Klassen, die dieses Attribut erben (z.B. `PLANT`, `DEALER`). Um die gewünschte Einfachheit in der Nutzung dieses objektrelationalen Beschreibungsschemas zu erreichen, wird aus den Informationen der OO-Klassendefinition eine neue Klassendefinition erzeugt, in der alle ererbten Attribute explizit aufgeführt werden. Diese Darstellung der Klasseninformationen steht in den Tabellen mit dem Präfix `SYSTEM_`.

Das Werkzeug »OMW-Classes« erlaubt es, einerseits eine Definition der Klassen anzulegen als auch aus diesen Definitionen die `SYSTEM` Tabellen zu erzeugen (siehe Abbildung 7.4). Um die Klassendefinitionen zu repräsentieren, wird hier wieder genau dasjenige Verfahren genutzt, welches auch für die Beschreibung der Simulationsmodelle zum Einsatz kommt. Das heißt, Klassendefinitionen sind wiederum Objekte in diesem Schema und sind in den bereits oben erwähnten drei Tabellen mit dem Präfix `IN_` abgelegt.

Der zweite Schritt zur Umsetzung des gewünschten Features besteht im Hinzufügen eines Attributs zur Klasse `CBOClass`. Das verwendete Werkzeug »Rational Rose« erlaubt die Erzeugung von Code auf Basis von UML Diagrammen und erlaubt somit die Übernahme des neuen Attributs `.TimingPrio` in den C++-Sourcecode auf »Knopfdruck«. Dabei wird in der Deklaration des Attributes ein (optionaler) Präfix vorangestellt, hier ist dieser Präfix »`m_`«. Das neue Attribut trägt im C++-Code den Namen »`m_TimingPrio`«. Der Code zum Einlesen des neuen Attributes ist obigem Beispiel sehr ähnlich:

```
m_TimingPrio = dbo->GetFloatValue("TimingPrio");
```

Das neue Attribut kann nun sowohl im Modell parametrisiert als auch in den Simu-

²¹Es sind lediglich wenige Zeilen hinzuzufügen.

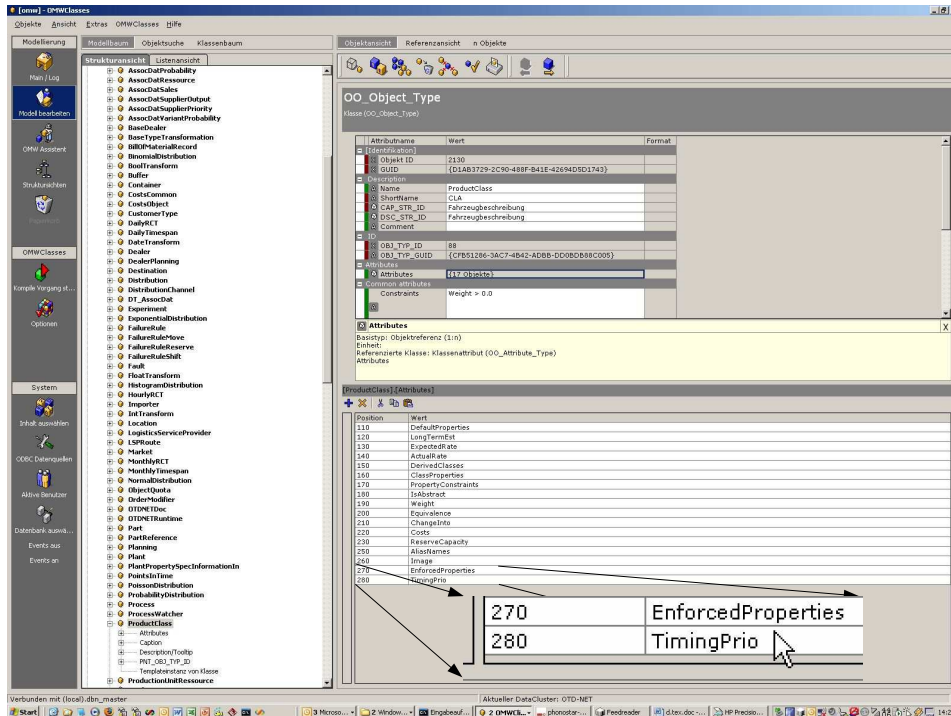


Abbildung 7.4.: Werkzeug zur Klassendefinition

lator eingelesen werden. Es fehlt der letzte Schritt: Die Nutzung des neuen Attributs in einer veränderten Gütefunktion. Unten die ursprüngliche Implementierung²².

$$\text{goodness} = \text{dpp} + (1 - \text{load}) * \text{cpp} + \text{cpd} - \text{fabs}(\text{ddiff}) * \text{c};$$

Die neue Implementierung könnte so aussehen:

```
// cbo ist ein Zeiger vom Typ CBOClass*
goodness = dpp + (1 - load) * cpp +
cpd - fabs(ddiff) * c * cbo->m_TimingPrio;
```

Nach Übersetzung des Source-Codes ist das neue Feature verfügbar. Während in obigem Beispiel der erforderliche Aufwand für die Änderung der GME-Klassen ähnlich dem Aufwand für die Anpassung der UML-Klassen und dem Hinzufügen des C++-Codes war, ist für komplexere Operationen der größere Aufwand i.d.R. im Bereich des C++-Codes zu sehen, da Änderungen von Algorithmen hier mit großer Wahrscheinlichkeit am effizientesten durchzuführen sind²³.

²² *dpp* ist die Priorität des jeweiligen Werkes, für den bestellenden Händler. *ddiff* ist die Differenz des zu bewertenden Fertigstellungstermins zum gewünschten Termin. *load* ist die Auslastung des Werkes im fraglichen Zeitraum. *cpp* entspricht dem Parameter *LoadDiffPrio* im relevanten Objekt der Klasse *Planning* (es gibt nur eines im Modell). *cpd* ist der relevante (abhängig von *ddiff*) Faktor für die Gewichtung der Auslastung des Werkes.

²³ Es ist möglich, unter Nutzung weiterer Werkzeuge außerhalb des »eigentlichen« Simulationswerkzeuges Berechnungen durchzuführen. Für die Anbindung dieser Werkzeuge kann z.B. die im Simulator implementierte HTTP-Schnittstelle verwendet werden, welche auch für die Interaktion mit einer grafischen Benutzeroberfläche genutzt wird.

7.2.2. Ablauf der Simulation

Liegt ein Simulationsmodell vor, welches mit dem Simulator bewertet werden soll, müssen, unabhängig vom Modell, eine Anzahl von Operationen durchgeführt werden, die im Folgenden kurz beschrieben werden.

Einlesen des Modells

Wie in Abschnitt 7.1.1 beschrieben, werden die Eingangsdaten eines Simulationsmodells in drei Tabellen in einer Datenbank abgelegt²⁴, die erforderlichen Typ-Informationen sind in weiteren drei relevanten Tabellen abgelegt²⁵. Die oben genannten Tabellen zur Benutzerverwaltung werden vom Simulator nicht verwendet, da der Simulator zum einen nur lesend auf das Modell zugreift und zum anderen alle Informationen innerhalb des Modells für eine Bewertung benötigt²⁶.

Alle oben genannten Tabellen werden vom Simulator im ersten Schritt gemeinsam in den Hauptspeicher gelesen, da das Einlesen ganzer Tabellen sehr viel effizienter durchgeführt werden kann. Im nächsten Schritt wird nach der Typ-ID eines singulären Objektes des Typs SIMPARAMETERS gesucht. Ist diese ID gefunden, kann nach einem Attribut gesucht werden, welches die Objekt-ID des Wurzel-Objektes des Simulationsmodells angibt, wiederum ein singuläres Objekt vom Typ EXPERIMENT.

Liegt die benötigte Information vor (die Objekt-ID), wird dieses Objekt instantiiert. Die oben bereits erwähnte Klasse CDBObjekt ermöglicht es, auf die Daten in den Eingangstabellen über einfache Selektoren zuzugreifen. Während der Instantiierung des Objektes vom Typ EXPERIMENT wird z.B. das Attribut .MARKETS eingelesen. Die Objekte der Klasse MARKET sind zu diesem Zeitpunkt noch nicht instantiiert. Diese Instantiierung wird nun (rekursiv) durchgeführt. Bei der Erzeugung der Objekte vom Typ MARKET wiederum werden Objekte vom Typ DEALER²⁷ erzeugt. Wenn die Methode zur Erzeugung des Objektes vom Typ-Experiment vollständig abgearbeitet ist, wurde rekursiv das gesamte Simulationsmodell im Hauptspeicher erzeugt. Zu diesem Zeitpunkt liegt das Simulationsmodell zweifach im Hauptspeicher vor: Einmal in den oben genannten Tabellen und zum Zweiten in Form instantiiertter Objekte.

Im nächsten Schritt kann nun die Tabellenrepräsentation des Modells aus dem Hauptspeicher entfernt werden²⁸.

Initialisierung

Nach der Erzeugung der Objekte im Hauptspeicher in dem »Format«, welches für die Simulation benötigt wird, sind zwar die während der Simulation benötigten Informationen vorhanden, aber noch nicht notwendigerweise in einer Form, die einen effizienten Zugriff ermöglicht. Soll z.B. der Zulieferer für eine Eigenschaft einer Produktklasse in einem Werk gefunden werden, muss der Zulieferer in einer Liste von

²⁴IN_Object, IN_Attribute und IN_Vector_Entries.

²⁵SYSTEM_Object_Type, SYSTEM_Attribute_Type und SYSTEM_StringTable.

²⁶Für Benutzer mit eingeschränkten Berechtigungen für das Lesen einzelner Modellbestandteile kann eine eingeschränkte Sicht auf die Ergebnisse erzeugt werden. Hierfür kann das Berechtigungskonzept der jeweils verwendeten Datenbank genutzt werden.

²⁷Bzw. davon abgeleiteter Klassen.

²⁸Diese Doppelrepräsentation ist in Bezug auf den Hauptspeicherbedarf völlig unkritisch, die folgenden Phasen der Simulation erfordern ein Vielfaches des hier verwendeten Hauptspeichers.

Objekten²⁹ gesucht werden. Diese Liste enthält Objekte, die eine Eigenschaftenspezifikation und einen Verweis auf Priorisierungsinformationen enthalten (referenzieren). Erst in den Priorisierungsinformationen stehen ein oder mehrere alternative Zulieferer. Die Priorisierung gibt an, welche Zulieferer bevorzugt genutzt bzw. in welcher Reihenfolge Zulieferer im Fall von Kapazitätsengpässen angefragt werden sollen³⁰, kann aber über den Simulationszeitraum geändert werden.

Diese Suche nach dem »zuständigen« Zulieferer wird im Laufe der Simulation sehr häufig durchgeführt. Das Konzept der Einplanung von Aufträgen in Werke nutzt eine Gütefunktion (siehe Abschnitt 6.5.2), um die Eignung einzelner Tage für eine Einplanung in einem der relevanten Werke zu prüfen. Der bzgl. der Gütefunktion am besten geeignete Tag wird dann für die Einplanung ausgewählt. Stehen z.B. fünf Werke zur Auswahl, in denen jeweils 50 Tage für eine Einplanung in Frage kommen und muss unter 10 Zulieferern der jeweils richtige gesucht werden, ist der Suchaufwand für jeden einzelnen Auftrag bereits sehr beträchtlich. Bei einer Million Aufträge ergeben sich $5 \times 50 \times 1.000.000 = 250.000.000$ Suchvorgänge unter den 10 Zulieferern, wobei während des Suchvorgangs abhängig vom zu diesem Zeitpunkt »gegenwärtigen« Simulationsdatum die Priorisierungsinformationen über relevante Zulieferer einer Eigenschaft erzeugt werden müssen.

Für sich betrachtet ist der Rechenaufwand überschaubar, jedoch sind Suchoperationen dieser Art an sehr vielen Stellen im Simulationslauf zu leisten. Um die gewünschten Laufzeiteigenschaften zu erreichen, müssen die oben dargestellten Informationen in eine Form gebracht werden, die einen effizienten Zugriff ermöglicht. Hierdurch kann zwar die Zahl der Suchoperationen nicht reduziert werden, jedoch die Dauer der einzelnen Suche.

Um dies zu erreichen, wird eine Datenstruktur erzeugt, die die Zuliefererpriorisierung für jeden Tag der Simulation, für jede relevante Eigenschaft und für jedes Werk enthält. Nun wird diese Priorisierung *einmal* vor Beginn der eigentlichen Simulation berechnet. Im Anschluss kann unter Angabe eines Simulationstages und einer Eigenschaft eine Liste von Zulieferern in der jeweils für diesen Tag richtigen Reihenfolge erhalten werden.

Als immer noch sehr zeitaufwändig erweist sich an dieser Stelle die Identifikation der relevanten Zulieferer für einen Auftrag. Es ist zunächst zu fragen: Welche Eigenschaften enthält ein Auftrag? Im nächsten Schritt dann: Welche der oben erzeugten Priorisierungslisten ist die Richtige? Das Durchsuchen der Eigenschaften eines Auftrags und das anschließende Suchen in der Lieferantenliste kann eine quadratische Zunahme des Suchaufwands in der Anzahl Eigenschaftenfamilien und Zulieferer zur Folge haben. Eine starke Verbesserung der Effizienz gegenüber der Suche in den jeweiligen Objektlisten wird hier erreicht, indem an den Eigenschaftenspezifikationen ein Vektor die Liste der Zuliefererpriorisierungen für das jeweilige Werk verwaltet³¹. Durch diese Lösung wird erreicht, dass die quadratische Abhängigkeit des Suchaufwandes nur noch unter der sehr unwahrscheinlichen Bedingungen relevant wird³².

In der Initialisierungsphase, nach dem Laden des Modells aus der Datenbank, wird

²⁹Objekte des Typs PLANTPROPERTYSPECINFORMATIONIN.

³⁰Ist ausreichend Kapazität vorhanden, wird der am höchsten priorisierte Zulieferer ausgewählt.

³¹Im Attribut `.PlantSupplierVector` in Abbildung D.7.

³²Die Liste der alternativen Zulieferer muss weiterhin untersucht werden. Im Regelfall wird aber die Liste der alternativen Zulieferer im Vergleich zur Gesamtanzahl der Zulieferer eher kurz sein.

eine Vielzahl von solchen Datenstrukturen erzeugt, um einen effizienten Zugriff auf die Daten im Simulationsmodell zu ermöglichen. Neben den offensichtlichen Vorteilen solcher Strukturen sind allerdings auch Nachteile in Kauf zu nehmen. Hier sind zum einen der potenzielle hohe Speicherverbrauch zu nennen und weiterhin die nicht unbedingte Vorteilhaftigkeit bzgl. der Effizienz.

In der hier realisierten Form wird die Vermeidung des quadratischen Laufzeitverhaltens teilweise durch ein quadratisches Ansteigen des Speicherbedarfs in der Anzahl der jeweiligen Modellelemente »erkauft«. Während aber der Vorteil durch effiziente Zugriffsstrukturen durch verkürzte Laufzeiten stets präsent ist, entstehen Nachteile nur dann, wenn der Speicherbedarf so groß wird, dass eine Berechnung nicht mehr möglich ist. Für »praktisch« in der Automobilindustrie auftretende Aufgabenstellungen ist das allerdings nicht zu erwarten und der Fortschritt der Computertechnologie verschiebt die Grenze der handhabbaren Modellgrößen immer weiter.

Ein weiterer potentieller Nachteil ist die eventuelle Unnötigkeit der Bereitstellung solcher Zugriffsstrukturen. Wenn also z.B. nur ein Werk und nur eine Eigenschaftensfamilie in einem Simulationsmodell parametrisiert sind, ist ein Effizienzgewinn unter Umständen nicht mehr nachweisbar. Allerdings werden gerade solch kleine Modelle sehr schnell bewertet, so dass auch dies kein ernsthafter Nachteil ist. Eine weitere Aktivität der Initialisierungsphase ist das Erzeugen von »Ereignissen«. So wird z.B. von einem Objekt des Typs `CDistributionChannel` der Zeitpunkt des ersten Transports berechnet, der über diesen Kanal fährt³³ und als Simulationsergebnis in eine Warteschlange gestellt.

Start der Ereignisbearbeitung

Im Anschluss an die Initialisierungsphase beginnt die »eigentliche« Simulation. Im Rahmen dieser Ereignisbearbeitung wird ein Objekt der Warteschlange entnommen, welches den frühesten »Zeitstempel« trägt³⁴. Die »Uhrzeit« in der Simulation wird auf die im Objekt vermerkte Zeit gestellt und die Bearbeitung einer Aktion, die mit diesem Ereignis assoziiert ist, wird begonnen.

Im Falle des oben genannten Beispiels wird ein Transport gestartet, d.h. es wird untersucht, wie viele Fahrzeuge oder Behälter zum Abtransport bereitstehen, diese werden unter Beachtung der Transportkapazität verladen und es wird eine Ankunftszeit für den Transport berechnet. Diese Ankunftszeit wird in einem neu erzeugten Ereignisobjekt vermerkt und dieses Objekt wird in die Warteschlange für Simulationsergebnisse gestellt. Die Bearbeitung von Ereignisobjekten führt also potenziell zu der Erzeugung neuer Ereignisse.

Ein Seiteneffekt der Ereignisbearbeitung kann die Erzeugung von Datensätzen in der Datenbank sein. Ist beispielsweise in der Simulation ein Fahrzeug bei einem Kunden angekommen, werden gleich mehrere Einträge in der Datenbank erzeugt:

- ▷ In der Tabelle `OUT_BO` werden u.a. Daten über die Fertigung des Fahrzeugs vermerkt.

³³Wenn für diesen Distributionskanal ein Fahrplan vorgesehen ist.

³⁴Es können auch mehrere Objekte denselben Zeitstempel tragen. In einem solchen Fall werden die Objekte in der Reihenfolge ihrer Erzeugung abgearbeitet.

- ▷ Die Tabelle `OUT_DealerOrder` enthält Daten über eine etwaige Händlerbestellung, die durchgeführt wurde, bevor eine Kunde das Fahrzeug bestellt hat.
- ▷ Der Zeitpunkt der Kundenbestellung sowie das gewünschte Lieferdatum wird in die Tabelle `OUT_CustomerOrder` geschrieben.
- ▷ Eine (optionale) einstufige Auflösung in Teile kann in die Tabelle `OUT_Parts` geschrieben werden³⁵.
- ▷ Von den Kunden gewünschte, vom Händler bestellte Eigenschaften und tatsächlich gelieferte Eigenschaften werden in die Tabelle `OUT_Properties` geschrieben.

Im Anschluss an die Übertragung der für die Auftragsabwicklung relevanten Daten wird der Auftrag bzw. das virtuelle Fahrzeug aus der Simulation entfernt. Es befinden sich also nur solche Daten im Speicher des Berechnungswerkzeugs, die für die Simulation benötigt werden.

Ende der Ereignisbearbeitung

Am singulären Objekt `EXPERIMENT` eines Simulationsmodells wird der Start- und der Endzeitpunkt eines Simulationsexperimentes vermerkt. Wird während der Simulation ein Ereignis erzeugt, welches einen Zeitstempel trägt, der über das Ende der Simulation hinausgeht, wird dieses Ereignis verworfen, also nicht in die Warteschlange gestellt. Mit Fortschreiten der Simulationszeit wird daher die Anzahl der Ereignisse in der Warteschlange abnehmen³⁶. Ist das letzte Ereignis aus der Warteschlange abgearbeitet, ist die Simulation beendet.

Im Anschluss werden noch ausstehende Schreiboperationen in die Datenbank abgearbeitet³⁷ und weitere »Aufräumarbeiten« durchgeführt, um belegte Ressourcen an das Betriebssystem des verwendeten Computers zurückzugeben. Nach der Terminierung des Simulationsprogramms kann die Auswertung der Daten begonnen werden.

7.2.3. Kennzahlen

Die für die Berechnung einer Kennzahl benötigten Daten sind in vielen Fällen bereits in den Simulationsergebnissen enthalten (vgl. Anhang F), so dass die Erzeugung der Kennzahl unter Nutzung der bereits genannten Techniken SQL-Abfrage und OLAP durchgeführt werden kann. Fehlen Daten für die fragliche Kennzahl, können neue Attribute in den Ausgabetafeln leicht hinzugefügt werden, obwohl zum Schreiben der Daten neben (optional) SQL eine Massendatenschnittstelle verwendet wird. Eine Datenzugriffsschicht »versteckt« die Details der Implementierung dieser Massendatenschnittstelle.

In Abbildung 7.5 werden Kennzahlen in vier Kategorien unterteilt. Wesentliche

³⁵Diese Teileauflösung ist unabhängig von der Auflösung, die durch die Zulieferer durchgeführt wird und kann eingesetzt werden, wenn das Simulationswerkzeug nur zur Erzeugung von Aufträgen verwendet wird, ohne dass die Möglichkeiten zur Simulation genutzt werden.

³⁶Fehler im Simulationsmodell können dazu führen, dass immer neue Ereignisse erzeugt werden, die den Zeitstempel der gegenwärtigen Simulationszeit tragen. In einem solchen »pathologischen« Fall muss die Simulation abgebrochen werden, da der Simulationslauf nicht eigenständig terminiert.

³⁷Das Schreiben in die Datenbank erfolgt asynchron.

Ausgangsdaten für die Bereitstellung solcher Kennzahlen werden wie folgt repräsentiert³⁸:

- ▷ **Finanzorientierte Bewertung.** Variable Kosten werden in verschiedene Kategorien unterteilt (z.B. Materialkosten, Distributionskosten, etc.). Die Parametrisierung dieser Kosten kann sowohl auf der Ebene der Eigenschaften und der Produktklassen vorgenommen werden als auch auf der Ebene von Werken oder Distributionskanälen. So können z.B. die Materialkosten als die Summe der Materialkosten der Eigenschaften plus der Materialkosten der Produktklasse definiert werden. Distributionskosten können entweder an der Produktklasse gesetzt, oder aber abhängig vom tatsächlich genutzten Transportweg am Distributionskanal definiert werden. Auch Mischformen dieser Parametrisierung sind zulässig.

Für die Berechnung der Kapitalbindungskosten ist der Zeitpunkt, an dem die Kosten anfallen, von erheblicher Bedeutung. Für die Materialkosten wird hier z.B. der Zeitpunkt des tatsächlichen Fertigungsbeginns herangezogen, für den variablen Anteil der Produktionskosten das Fertigungsende usw.³⁹. Die zeitliche »Auflösung« der Berechnung der Kapitalbindungskosten ist eine Minute.

- ▷ **Ressourcenorientierte Bewertung.** Für die Ableitung von Kennzahlen im Bereich Ressourcen stehen vielfältige Daten zur Verfügung:
 - ▷ Geplante Kapazität der Ressourcen.
 - ▷ Tatsächliche Kapazität der Ressourcen⁴⁰.
 - ▷ Abgerufene Kapazität. Wieviel Kapazität wurde tatsächlich in Anspruch genommen?
 - ▷ Unterdeckung. Es wird Kapazität auch dann zur Verfügung gestellt (optional), wenn die eingestellte Kapazität nicht mehr ausreicht. Die Unterdeckung kann im Nachhinein analysiert werden.
 - ▷ Bestände. Aus einem (einstellbaren) Initialbestand, den eintreffenden Lieferungen und dem Verbrauch ergibt sich eine Bestandshöhe, die in den Ergebnisdaten nachvollzogen werden kann. Diese Ergebnisse stehen z.B. für Werke und Zulieferer zur Verfügung. Analog kann die Anzahl Behälter oder Fahrzeuge in einem Distributionskanal verfolgt werden.
- ▷ **Prozessorientierte Bewertung.** Kennzahlen wie Durchlaufzeit oder Ausbringung können direkt den Simulationsergebnissen entnommen werden. Komplexer defi-

³⁸Details der Inhalte der im Folgenden nicht erläuterten Kennzahlen aus Abbildung 7.5 in [Kel06].

³⁹Die Berechnungsvorschrift für Kapitalbindungskosten kann auf sehr viele verschiedene Arten festgelegt werden, abhängig von den konkreten Zahlungsmodalitäten für die im Fokus stehenden Produkte oder Teile. Der hier ermittelte Wert kann jedenfalls als Grundlage einer Abschätzung dienen. Eine für den Einzelfall angepasste Berechnung kann auf Basis der Ergebnisdaten durchgeführt werden, erfordert also keine Anpassung des Simulationswerkzeugs.

⁴⁰Durch Abweichungen zwischen geplanter und tatsächlicher Kapazität lassen sich z.B. Störungen abbilden. Diese Form der Abbildung hat gegenüber der ebenfalls möglichen stochastisch auftretenden Störung den Vorteil, dass so »typische« Störmuster leichter untersucht werden können. Der Verlauf der Kapazitätsveränderung wird hier vom Modellierer bestimmt.

nierte Kennzahlen, wie die Planstabilität⁴¹, erfordern die Einbeziehung von Informationen aus den Aufträgen bzw. Fahrzeugen, die im Betrachtungszeitraum produziert werden.

- ▷ **Strukturorientierte Bewertung.** Die Komplexität als beispielhafte Kategorie im Bereich der strukturorientierten Kennzahlen kann z.B. als Produktvariabilität (siehe Kapitel 1.1) aus dem Produktaufbau abgeleitet werden.

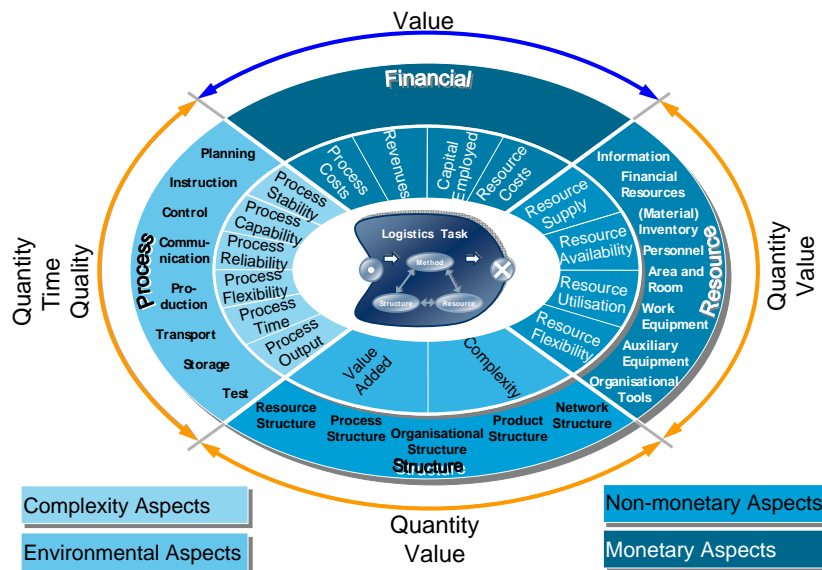


Abbildung 7.5.: Kennzahlenkategorien (Quelle: [Kel06, S. 16], vergl. auf Seite 28)

Die Möglichkeiten zur Ableitung von Kennzahlen aus den Simulationsergebnissen sind aufgrund der Vielzahl der im Verlauf der Simulation erzeugten Daten sehr umfassend. Einfache Kennzahlen, wie z.B. Durchlaufzeiten und Bestände sind häufig ohne weitere Berechnung den Simulationsergebnissen zu entnehmen. Kennzahlen, die durch Verknüpfung und/oder Aggregation von Ergebnisdaten entstehen, lassen sich durch geeignet definierte OLAP-Auswertungen erzeugen. Fehlen Informationen in den Ausgaben, die im Verlauf der Simulation im Werkzeug berechnet werden, können diese durch sehr kleine Code-Änderungen in die Ausgaben integriert werden.

7.2.4. Zusammenfassung

Das Bewertungswerkzeug – der Simulator – »spielt« die Abläufe innerhalb des abgebildeten Systems nach, indem zu den Zeitpunkten, in denen eine Zustandsveränderung im Modell stattfinden soll, ein Händler z.B. eine Bestellung abgibt, die für diesen Zeitpunkt vorgesehenen Aktionen durchgeführt werden. Die »Spuren« dieser

⁴¹Die Abweichung zwischen geplanter Fertigung und der tatsächlich stattgefundenen Fertigung. Wie auch für andere Kennzahlen gilt hier, dass diese auf höchst unterschiedliche Weise definiert sein kann. Die Planstabilität kann sich z.B. auf die Einhaltung der Reihenfolge konkreter Aufträge beziehen, wie auch auf die Einhaltung bloßer Anzahlen und geplanter Teilebedarfe.

Aktionen finden sich in der Datenbank, in der der Simulator wichtige Daten über die durchgeführten Aktionen ablegt. Hier lassen sich z.B. der Zeitpunkt eines Auftragseinganges, der erste geplante Fertigungstag für das im Auftrag bestellte Fahrzeug, der Transportweg des Fahrzeugs in einzelnen Abschnitten bis hin zur Auslieferung an den bestellenden Händler feststellen. Aus der Summe der Ergebnisdaten können zahlreiche Kennzahlen abgeleitet werden, die für eine Bewertung eines Prozessentwurfes benötigt werden.

Aus technischer Sicht können drei wesentliche Vorgänge in der Simulation in dieser hier umgesetzten Form unterschieden werden:

1. **Einlesen des Modells.** Aus dem Prozessmodell in der Datenbank wird ein strukturähnliches Modell im Hauptspeicher eines Rechners erzeugt.
2. **Ereignisdiskrete Simulation.** Das Modell wird im Hauptspeicher in zeitdiskreten Schritten »durchgespielt«.
3. **Schreiben von Ergebnissen.** Bereits während der Simulation werden Ergebnisse in die Datenbank geschrieben. Wird beispielsweise ein Fahrzeug an einen Kunden geliefert, werden Informationen über diesen Vorgang gesammelt und dokumentiert.

Ein weiteres wesentliches Merkmal der Umsetzung des Bewertungswerkzeugs ist die hauptspeicherbasierte Durchführung der Simulation. Alle für einen Simulationslauf benötigten Daten werden zu Beginn der Simulation in den Hauptspeicher des Computers geladen. Diese unmittelbare Verfügbarkeit aller benötigten Daten führt einerseits zu einer sehr zeiteffizienten Bewertung und erzwingt auf der anderen Seite eine sehr speichereffiziente Abbildung des Modells im Hauptspeicher, um trotz der Beschränkungen gegenwärtiger Rechner eine Bewertung auch umfangreicher Modelle zu ermöglichen.

7.3. Validierung

Um aus einem Simulationsmodell gültige Aussagen über ein Realsystem ableiten zu können, führt CHUNG (vgl. [Chu04, S. 160]) zwei Vorbedingungen an: Zum einen muss das Modell die in der Spezifikation aufgeführten Eigenschaften besitzen (»Building the model correctly«) und zum anderen muss das Modell das Realsystem angemessen abbilden (»Building the correct model«).

Abbildung 7.6 zeigt den Zusammenhang zwischen den verschiedenen Verifikations- und Validierungsaufgaben. Zur Bedeutung der *Verifikation* schreibt CHUNG (vgl. [Chu04, S. 160]):

It is pointless, perhaps impossible, to attempt to see if the model represents reality if the model does not even operate as intended. In other words, you should not attempt to validate a model that has not successfully undergone the verification process.

Allerdings wird hier unter *Verifikation* nicht eine *formale Verifikation* verstanden⁴².

⁴²CHUNG führt verschiedene Techniken zur Verifikation an (vgl. [Chu04, S. 161-169]): Nutzung von Animation, Verfolgung einzelner Objekte innerhalb der Simulation, Betrachtung der Ereignisbearbeitung Schritt um Schritt etc.

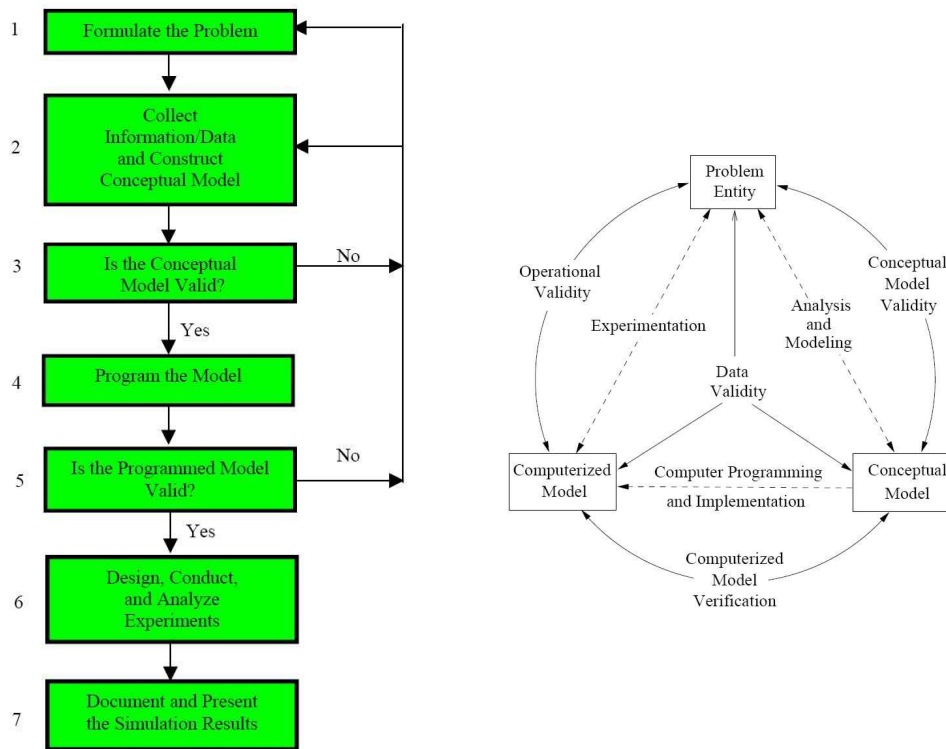


Abbildung 7.6.: Schritte der Durchführung einer Simulationsstudie (Quellen: [Law03, S. 67] und [Sar98, S. 123])

PAGE differenziert⁴³ (vgl. [Pag91, S. 16]):

Die Modellverifikation befasst sich mit der korrekten Abbildung des konzeptuellen Modells auf das Computer Modell. Es wird überprüft, ob das Programm eine korrekte Implementation des (formalen) konzeptuellen Modells darstellt. Trotz der üblichen Bezeichnung »Verifikation« handelt es sich normalerweise nicht um eine formale Programmverifikation, sondern vielmehr um systematische Programmtests.

Die formale Verifikation der Korrektheit eines Programms zeigt, ob ein Programm für alle zulässigen Eingangsdaten terminiert und korrekte Ergebnisse liefert. Somit ist eine formale Verifikation dem oben beschriebenen Ansatz der »systematischen Programmtests« prinzipiell vorzuziehen, da eine erfolgreiche formale Verifikation Fehler im Programm ausschließen kann. Allerdings ist die formale Verifikation außer für sehr kleine Programme sehr aufwändig und kann ihrerseits wiederum Fehler enthalten.

Bei der Modellvalidierung können verschiedene Methoden zum Einsatz kommen, die nachfolgend genannt und kurz beschrieben werden.

Eine ausführliche Beschreibung ist bei PAGE [Pag91, S. 150] zu finden:

▷ **Plausibilitätsprüfung.** Bei der Plausibilitätsprüfung wird das plausible Verhalten

⁴³Orthographie geändert.

des Modells und seiner Komponenten z.B. mittels analytischer Vergleichsrechnungen und/oder Expertenbefragungen untersucht.

- ▷ **Sensitivitätsanalyse.** Bei dieser Analyse wird untersucht, wie die Output-Daten auf Veränderungen bei den Input-Daten und/oder der Modellstruktur reagieren. Dabei soll auch festgestellt werden, von welchen Parametern die Modellergebnisse am stärksten abhängen. Dadurch können evtl. Strukturfehler erkannt sowie kritische Parameter identifiziert werden, die im Modell wie auch im Realsystem besonders beachtet werden müssen.
- ▷ **Outputvergleich.** Bei der Modellvalidierung nimmt der Outputvergleich eine zentrale Stellung ein. Bei ihm werden ein oder mehrere Kennzahlen aus dem Realsystem erhoben oder berechnet und mit den entsprechenden Werten des Simulationsmodells verglichen. Geeignete Maßzahlen sind z.B. der Mittelwert, die Stichprobenvarianz⁴⁴, der Median sowie das Minimum und Maximum (vgl. [Pag91, S. 151]).
- ▷ **Prognostische und dynamische Gültigkeitsprüfung.** Wenn das Modell in der Praxis als Planungs- und Entscheidungsinstrument eingesetzt werden soll, ist diese Art der Modellvalidierung zusätzlich anzuwenden. Bei ihr steht der Nachweis der prognostischen und dynamischen Validität im Vordergrund. »Bei der prognostischen Gültigkeitsprüfung muss sich erweisen, ob die Prognosen des Modells mit den eintretenden Ereignissen übereinstimmen (Prognosefähigkeit) [Pag91, S. 154].

Die hier aufgeführten Maßnahmen der Modellvalidierung stellen einen Maximalkatalog dar, von dem für den überwiegenden Teil der Modellstudien nur eine bestimmte Anzahl anwendbar ist (vgl. [Pag91, S. 154]). Welche der Methoden der Validierung im Einzelfall zum Einsatz kommen, ist somit vom Modelltyp und von den verfügbaren Daten abhängig (vgl. [Pag91, S. 155]).

Eine wichtige Voraussetzung für die Validierung eines Modells ist die Kenntnis des Verwendungszwecks. Ein Modell kann bzgl. eines Verwendungszwecks gültig sein und bzgl. eines anderen ungültig (vgl. [Sar98, S. 121]). Für den im Rahmen dieser Arbeit umgesetzten Simulator ist dieser Verwendungszweck nicht exakt a priori bekannt, daher kann eine Validierung im obigen Sinne nicht geleistet werden. Trotzdem können im Rahmen der Verifikation (im Sinne von Programmtests) Techniken genutzt werden, die für die Validierung zum Einsatz kommen, indem Test-Modelle mit einem dann bekannten Einsatzzweck definiert werden.

Der im Rahmen dieser Arbeit entwickelte Simulator wurde zahlreichen Software-Tests unterzogen⁴⁵. Die grundlegenden Techniken hierzu werden in der Literatur ausführlich behandelt (vgl. z.B. [Bal99, S. 501-538], [Bal98, S. 391-502]) und sollen an dieser Stelle daher nicht noch einmal dargestellt werden.

⁴⁴Die Stichprobenvarianz oder auch nur Varianz ist ein Maß für die absolute Größe der Streuung um den Mittelwert [Har89, S. 117f]. Sie misst den quadratischen Abstand zum arithmetischen Mittel (Mittelwert) [Har89, S. 43f].

⁴⁵Die im Rahmen dieser Arbeit erzeugte Software erfüllt im Hinblick auf Software-Tests nicht Anforderungen, wie sie z.B. für Datenbanken oder ERP-Systeme gelten müssen. Solche Tests sind äußerst aufwändig und im Rahmen dieser Arbeit nicht zu leisten.

7.4. Abgleich mit den Anforderungen

Auf Basis der Ergebnisse der Kapitel 6 und 7 wird im Folgenden der hier entwickelte Modellierungsansatz auf den Erfüllungsgrad hinsichtlich der in Kapitel 4 aufgeführten Anforderungen überprüft.

Anforderung 1 (Seite 57): Die geforderten fünf Modellelemente sind vollständig vorhanden, ebenso die geforderten Parameter.

Anforderung 2 (Seite 57): Der durchgängig hohe Detaillierungsgrad erlaubt die gewünschte Ableitung von Systemlasten auch für Teilprozesse.

Anforderung 3 (Seite 58): Die Ableitung von Kennzahlen ist auch für Teilprozesse möglich, da der Weg der Leistungsobjekte durch diese Teilprozesse in den Ausgaben detailliert dokumentiert wird.

Anforderung 4 (Seite 58): Kennzahlen können auf Basis der vielfältigen zur Verfügung stehenden Simulationsergebnisse frei definiert und somit den jeweiligen Erfordernissen angepasst werden. Falls erforderlich kann eine Erweiterung der Menge der Ausgabedaten leicht vorgenommen werden.

Anforderung 5 (Seite 59): Alle in der Anforderung genannten Informationen sind verfügbar.

Anforderung 6 (Seite 60): Das hier entwickelte Werkzeug (»OTD-NET«) kann im Rahmen einer Simulation Aufträge erzeugen, die als »Systemlast« für diese Simulation dienen.

Anforderung 7 (Seite 61): Der hier umgesetzte Algorithmus zur Auftragserzeugung berücksichtigt für die Auftragserzeugung Konstruktionsregeln, d.h. keiner der erzeugten Aufträge verletzt die Konstruktionsregeln. Die Formulierung widersprüchlicher Konstruktionsregeln wird allerdings nicht überprüft. In einem solchen Fall werden Aufträge erzeugt, die die »Standardsetzung« enthalten.

Anforderung 8 (Seite 63): Die umgesetzte Form der Produktbeschreibung ist an vielfältige Einsatzzwecke anpassbar.

Anforderung 9 (Seite 63): Das Schema der Produktbeschreibung erlaubt die Umsetzung sowohl sehr einfacher als auch äußerst komplexer Produktstrukturen.

Anforderung 10 (Seite 64): Die Einbauraten der Eigenschaften bei den erzeugten Aufträgen nähern die Vorgaben sehr gut an (siehe Abbildung B.9 auf Seite 225); ca. 4000 Aufträge pro Woche).

Anforderung 11 (Seite 65): Für die Abbildung von Einplanungsalgorithmen stehen zahlreiche Parameter in den relevanten Modellbausteinen zur Verfügung. Darüber hinaus gehende Anforderungen lassen sich leicht durch eine Erweiterung des Klassenmodells (siehe oben) bzw. der Implementierung umsetzen.

Anforderung 12 (Seite 65): Die Abbildung von Ressourcen ermöglicht die Einstellung von Plan- und Ist-Kapazitäten. Weiterhin können stochastische Kapazitätsveränderungen parametrisiert werden.

Anforderung 13 (Seite 67): Die geforderte Flexibilität ist in vollem Umfang vorhanden. Priorisierungsregeln werden durch eine Gütefunktion für die Einplanung umgesetzt.

Anforderung 14 (Seite 67): Die Repräsentation von Aufträgen im Bewertungswerkzeug enthält Informationen über das Durchlaufen aller relevanten, d.h. in den Planungsalgorithmen zugreifbaren, Statuspunkte. Werden für erweiterte Planungsal-

gorithmen zusätzliche Informationen erforderlich, können diese leicht dem Informationsmodell des Bewertungswerkzeugs hinzugefügt werden.

Anforderung 15 (Seite 68): Störungen können sowohl als Abweichungen der tatsächlichen von der geplanten Kapazität als auch als stochastische Einflüsse abgebildet werden.

Anforderung 16 (Seite 69): Entscheidungen der kurzfristigen Kapazitätsplanung (z.B. Vorproduktion von Teilen bei Lieferanten) können automatisiert vom Bewertungswerkzeug getroffen werden. Eine Beeinflussung dieser Planung ist durch eine geeignete Parametrisierung möglich.

Anforderung 17 (Seite 70): Die Zuteilung von Kapazitäten an die jeweiligen Abnehmer kann sowohl auf Prognosebasis als auch auf Basis von vorhandenen Aufträgen erfolgen.

Anforderung 18 (Seite 70): Die Abdeckung von Bedarfsschwankungen durch Vor- und Nachproduktion durch Zulieferer ist möglich.

Anforderung 19 (Seite 71): Für die Abbildung von Distributionsnetzwerken gefertigter Fahrzeuge stehen umfangreiche Möglichkeiten zur Verfügung. Neben den Materialflüssen können auch Verfahren zur Disposition von Transportmitteln repräsentiert werden.

Anforderung 20 (Seite 72): Alle Möglichkeiten der Abbildung von Distributionsnetzwerken für gefertigte Fahrzeuge stehen auch für die Distribution von Teilen zur Verfügung. Für den Transport von Teilen können Behälter bestimmt werden, die mit Informationen zu Fassungsvermögen und Ausmaßen versehen werden können.

Anforderung 21 (Seite 73): Die Bewertung von Modellen in der geforderten Größe ist möglich.

Anforderung 22 (Seite 73): Die Bewertung vergleichbar großer Modelle, wie in der Anforderung, ist in deutlich unter 24 Stunden möglich.

Anforderung 23 (Seite 74): Modellaufbau bzw. -veränderung können durch »externe« Werkzeuge erfolgen. Dies kann entweder durch direkten Zugriff auf die Datenbank erfolgen, oder durch Bearbeitung einer XML-Datei, die das Simulationsmodell repräsentiert.

7.5. Zusammenfassung

Die Bewertung von Auftragsabwicklungsprozessen wird durch eine dreigeteilte Vorgehensweise unterstützt:

1. Abbildung eines konzeptionellen Prozessmodells in eine datenbankgestützte Repräsentation (OTD-NET Modell).
2. Einlesen des Modells in das Bewertungswerkzeug und dynamische ereignisdiskrete Simulation. Simulationsergebnisse werden in die Datenbank geschrieben.
3. Auswertung der Ergebnisse mittels Datenbankabfragen.

Vor- und nachgelagerte Schritte, wie z.B. die adäquate Problemformulierung oder die Dokumentation der Ergebnisse, vervollständigen die Vorgehensweise, so dass mittels des entwickelten Instrumentariums, in Verbindung mit bewährten Vorgehensmo-

dellen zur Durchführung von Simulationstudien, eine Bewertung von Auftragsabwicklungsprozessen mittels Simulation möglich wird.

Aufgrund der Entscheidung, sowohl den Modellierungsansatz als auch das Bewertungswerkzeug auf relativ allgemeinen Ansätzen basieren zu lassen – UML und die Programmiersprache C++ – konnten die in Kapitel 4 formulierten Anforderungen sehr weitgehend umgesetzt werden.

8. Beispielhafter Einsatz

Im Anschluss an die Beschreibung der Umsetzung des Simulationswerkzeugs soll an dieser Stelle anhand eines konkreten Beispiels gezeigt werden, dass der in Kapitel 6 dargestellte Modellierungsansatz grundsätzlich geeignet ist, die Anforderungen aus Kapitel 4 zu erfüllen.

Das ausgewählte Beispiel wurde innerhalb des EU-Projekts ILIPT entwickelt¹, unter anderem um aufzuzeigen, dass unter bestimmten Rahmenbedingungen eine Lieferzeit von fünf Tagen innerhalb Europas für ein nach Kundenwünschen spezifiziertes Fahrzeug aus Sicht eines »Endkunden« erreicht werden kann. Das Teilprojekt (»Thema II«), in dem das Simulationsmodell entwickelt wurde, trägt den Namen »FlexNet«, daher wird im Folgenden vom *FlexNet-Modell* gesprochen.

Die Untersuchung dieser Fragestellung erfordert eine »gleichmäßig« detaillierte Abbildung des gesamten Auftragsabwicklungsprozesses, so dass das resultierende Modell sowohl komplex genug ist, um zu zeigen, dass auch anspruchsvolle Aufgabestellungen bearbeitet werden können, aber andererseits einfach genug ist, um an dieser Stelle die Modellstruktur umfassend beschreiben zu können. Eine detaillierte Beschreibung des hier vorgestellten Modells kann im entsprechenden ILIPT Deliverable [WSWK06] gefunden werden.

Eine detaillierte Beschreibung der erforderlichen Schritte zur Erzeugung eines Simulationsmodells kann z.B. in [Law03] gefunden werden.

8.1. Aufgabenstellung

Das wesentliche Ziel für die Erstellung des im Folgenden beschriebenen Modells ist die Überprüfung eines Konzeptes, mit dem die Herstellung eines BTO-Fahrzeugs innerhalb von fünf Tagen für den Europäischen Markt ermöglicht werden soll. Die Modellstruktur berücksichtigt in Bezug auf Werksstandorte und Zuliefererstandorte die gegenwärtige Situation², kann aber auch über gegenwärtige Strukturen hinausgreifen, da die zu prüfenden Konzepte im Jahr 2015 betrieben werden sollen.

8.2. Modellumfang

Der benötigte Modellumfang hängt von der konkreten Fragestellung ab, die im vorliegenden Fall zum einen auf übergreifende Kennzahlen wie z.B. die Lieferzeit abzielt, aber zum anderen auch eine Abbildung des Liefernetzwerks erfordert, weil gerade

¹Daher sind alle spezifischen Bezeichnungen innerhalb des Modells in englischer Sprache.

²Die Standorte von Werken der Automobilhersteller und Zulieferer im Modell werden möglichst so gewählt, dass sich an den jeweiligen Orten tatsächlich eine Fertigungseinrichtung eines Zulieferers befindet. Es wird aber nicht sichergestellt, dass in diesen Fertigungseinrichtungen zum heutigen Zeitpunkt die Produkte gefertigt werden können, die im Modell von diesen Orten geliefert werden.

die Untersuchung des Zusammenspiels zwischen Automobilherstellern und Zulieferern von besonderem Interesse in Bezug auf die Erreichung des Ziels von fünf Tagen Lieferzeit ist³.

8.2.1. Produktstruktur

Es werden die Fahrzeugmodelle

- ▷ Station Wagon
- ▷ Hatchback
- ▷ Limousine
- ▷ Convertible

im Modell abgebildet.

Die Anzahl der Eigenschaftsfamilien wird so gewählt, dass eine Betrachtung der Lieferantenanbindung sinnvoll möglich ist:

- ▷ Engine Modules
- ▷ Front Modules
- ▷ Door Module

Eine Liste der Eigenschaften in diesen Eigenschaftsfamilien ist im Anhang B abgebildet. Der Screenshot der Bedienoberfläche in Abbildung 8.1 zeigt einen Ausschnitt der Produktstruktur des Beispiels. Wurzel der Produktklassen ist die Klasse PD ILIPT CAR, von welcher die oben genannten Varianten abgeleitet werden. Weil der Betrachtungszeitraum des Modells weit in der Zukunft liegt und daher keine verlässlichen Annahmen über die Abweichungen der Einbauraten der (z.B.) Motorvarianten zwischen Limousinen und Schrägheck-Fahrzeugen (Hatchback) getroffen werden können, werden im Modell die Einbauraten nur einmal an der Klasse PD ILIPT CAR festgelegt.

Alle abgeleiteten Klassen »erben« die Einbauraten. Ebenso werden die »Setzungen« (Default Properties in der Abbildung) vererbt. Jedes Fahrzeug der oben definierten Produktklassen enthält also ein Triebwerk des Typs *EM gasoline automatic*, falls nicht im Verlauf des Auftragserzeugungsprozesses aufgrund der parametrisierten Einbauraten diese Setzung überschrieben wird.

Ein weiteres wichtiges Element des Modells ist die Stückliste. In der Stückliste wird festgelegt, welche Teile in ein Fahrzeug verbaut werden sollen. In Abbildung B.4 im Anhang sind zwei Stücklistenstrukturen in der Definitionsansicht in der Bedienoberfläche zu sehen. Die Stücklisten können abhängig vom Fertigungsstandort festgelegt werden. Diese Möglichkeit wurde in dem Modell genutzt, so dass für einen Motor aus Marseille ein anderer Basismotor verwendet wird, als für einen Motor aus Madgeburg.

³Die sehr kurzen »Vorwarnzeiten« in einem solchen Szenario führen dazu, dass Probleme in der Teileversorgung nahe liegend zu manifesten Effekten im Bereich der Lieferzeit führen.

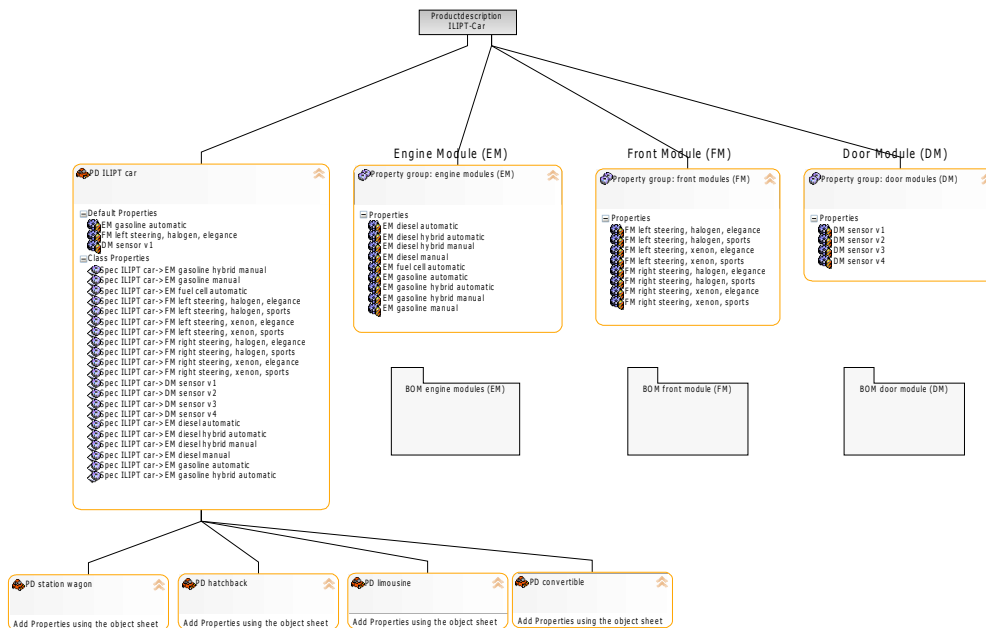


Abbildung 8.1.: Produktstruktur Beispielmodell

8.2.2. Märkte

Die Auswahl der Märkte für das Simulationsmodell hat zahlreiche Auswirkungen: Kunden in den jeweiligen Märkten haben andere Präferenzstrukturen, kaufen also anders ausgestattete Fahrzeuge in u.U. abweichender Saisonalität, haben eine unterschiedlich hohe Bereitschaft Lagerfahrzeuge abzunehmen und schließlich ist der Distributionsweg der Fahrzeuge unterschiedlich lang. Eine für das FlexNet-Modell wichtige Rahmenbedingung ist, dass Fahrzeuge nur dann gebaut werden, wenn auch tatsächlich ein Kunde hierfür gefunden ist, dass Fahrzeug also ausschließlich im Kundenauftrag, in der vom Kunden gewünschten Spezifikation gefertigt wird (BTO). Da die Verteilung der Fahrzeuge innerhalb der Märkte nicht im Fokus der Untersuchung steht, wird pro Markt genau ein Händler angenommen, der stellvertretend die Fahrzeuge entgegennimmt. Die Märkte im Einzelnen:

1. Österreich
2. Benelux
3. Frankreich
4. Deutschland
5. Italien
6. Spanien
7. Schweiz
8. Großbritannien

9. Tschechische Republik, Ungarn, Slowakei, Slowenien
10. Dänemark, Norwegen, Schweden, Finnland
11. Estland, Lettland, Litauen, Polen
12. NAFTA⁴
13. Großkunden

Die Märkte sind zum einen nicht ausschließlich eigenständige Länder, sondern fassen teilweise mehrere Länder ähnlicher Charakteristika zu einem Markt zusammen und beinhalten zum anderen auch Kundengruppen, wie z.B. Fahrzeugvermieter, die ein besonderes Bestellverhalten haben⁵.

8.2.3. Prozesssteuerung

Die Aufgabenstellung für das Modell macht klare Vorgaben, welche Ziele bei der Auftragseinplanung zu verfolgen sind. Es soll die grundsätzliche Erreichbarkeit einer Lieferzeit von fünf Tagen für ein BTO-Fahrzeug überprüft werden. Aufgrund dieser ehrgeizigen Zielsetzung ist der Spielraum für die Auftragseinplanung gering.

Die für Fertigung und Distribution zur Verfügung stehende Zeit wird durch die benötigte Zeit für die Produktion der Teile für die Fahrzeugproduktion (drei Tage, siehe Abschnitt 8.2.4) auf zwei Tage beschränkt⁶. Drei Tage Zeit für die Fertigung der Teile bedeuten, dass die Fahrzeugspezifikation drei Tage vor Fertigungsbeginn feststehen muss, d.h. der Kundenauftrag muss vorliegen und wird eingefroren.

Aus diesen engen zeitlichen Restriktionen folgt unmittelbar, dass bei der Auftragsplanung Priorität auf der frühestmöglichen Fertigung liegen muss, Auslastungskriterien können nur eine untergeordnete Rolle spielen. Die Parameter für die Gütefunktion der Auftragseinplanung (siehe Abschnitt 6.5.2) werden daher in dieser Weise festgelegt.

8.2.4. Werke und Zulieferer

Um eine zeitgerechte Belieferung der gewählten Märkte innerhalb von nur wenig mehr als einem Tag⁷ sicherstellen zu können, müssen die Fertigungsstandorte geographisch möglichst in der Nähe der Abnehmer gewählt werden (siehe Abbildung B.2). Eine gute Abdeckung Europas wird erreicht mit den Standorten⁸:

1. Bremen

⁴Unterzeichner des North American Free Trade Agreement, Kanada, USA, Mexico. Für diesen Markt gilt die Lieferzeitsvorgabe nicht.

⁵Für Großkunden wird hier angenommen, dass diese ihre Bestellungen 60 Tage vor der geplanten Fertigstellung der Fahrzeuge abgeben.

⁶Bei einem Lieferzeitziel von fünf Tagen.

⁷Fünf Tage Gesamtlieferzeitsvorgabe abzüglich drei Tagen für die Teilefertigung und einigen Stunden (3 bis 7 im Modell) für die Fahrzeugfertigung.

⁸Ein Ziel für die Standortwahl, neben der geographischen Abdeckung, war es, zwar durchaus existierende Fertigungsstandorte von Automobilherstellern zu wählen, hierbei sollte jedoch max. ein existierender Fertigungsstandort jeweils eines Herstellers in der Liste erscheinen, um das Modell nicht oberflächlich herstellerspezifisch aussehen zu lassen.

2. Toulouse
3. Turin
4. Prag

Die Zulieferer, die unmittelbar die OEMs beliefern (1st-Tier Zulieferer), stellen teilweise die Versorgung mehrerer Werke sicher und sind räumlich zwischen jeweils zwei belieferten Werken lokalisiert, um möglichst kurze Belieferungswege zu realisieren. Der Standort Magdeburg beliefert die Werke Bremen und Prag, der Standort Marseille beliefert die Werke in Toulouse und Turin (jeweils) mit Motoren.

Die nachgelagerten Zulieferer (2nd-Tier), die wiederum die oben genannten 1st-Tier Zulieferer beliefern, sind im Modell an folgenden Standorten angeordnet:

1. Bad Cannstatt (Deutschland)
2. Corbetta (Italien)
3. Chavanod (Frankreich)
4. Pamplona (Spanien)

Die Zulieferer im Modell produzieren zum Teil auftrags- und zum Teil lagerorientiert (siehe Abbildung B.1). Jedoch ist die Fertigung der Motoren im Modell so organisiert, dass nicht nur die 1st-Tier Zulieferer, sondern auch die 2nd-Tier Lieferanten auftragsorientiert arbeiten (kurz: BTO/BTO). Die Fertigung der Frontmodule erfolgt auftragsorientiert durch die 1st-Tier Zulieferer und lagerorientiert durch die 2nd-Tier Zulieferer (kurz: BTO/BTS). Dieser 2nd-Tier Zulieferer hat seinen Standort in Mohelnice (Tschechische Republik).

Ziel dieser Aufteilung der Zulieferer im Modell in dieser Weise ist die Untersuchung des Einflusses der Organisation der Lieferketten auf den Durchlauf des Kundenauftrages. Gelingt es, variantenreiche Produkte auch beim 2nd-Tier Zulieferer auftragsbezogen zu fertigen, kann hierdurch unter Umständen eine Kostenreduktion erzielt werden.

8.2.5. Distribution

Distributionsprozessen kommen an zwei Stellen eine wichtige Bedeutung zu: Zum einen ist zu prüfen, ob es möglich ist, Teile zwischen den Zulieferern untereinander und zum Werk zeitgerecht zu transportieren und zum anderen ist zu ermitteln, in welcher Entfernung vom Werk noch Fahrzeuge⁹ innerhalb von fünf Tagen Lieferzeit geliefert werden können. Dies hängt unter anderem davon ab, von welchen Werken ein Markt beliefert wird, welche Fahrzeitregelungen für die Transporte gelten (Sonntagsfahrverbote) usw.

Abbildung B.2 zeigt, welche Regionen von den jeweiligen Werken innerhalb von 16 Stunden erreicht werden können, wenn zwei Fahrer pro LKW und ein LKW-typisches Geschwindigkeitsprofil unterstellt werden¹⁰.

⁹Build-to-Order gefertigte Fahrzeuge.

¹⁰Zum Beispiel 70 Km/h Durchschnittsgeschwindigkeit auf Autobahnen, Pausenzeiten etc.

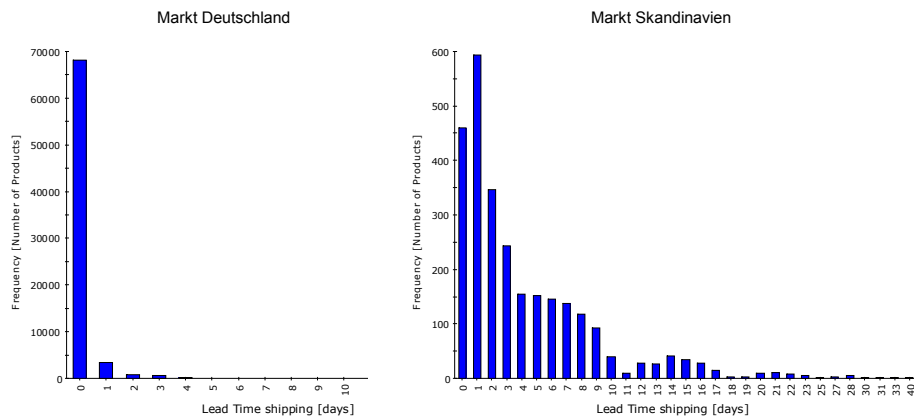


Abbildung 8.2.: Transportzeit im Vergleich

8.3. Ergebnisse

Die oben gestellte Frage nach der Machbarkeit von fünf Tagen Lieferzeit für auftragsbezogen gefertigte Fahrzeuge kann nun schlüssig beantwortet werden. Bereits in der Modellierungsphase zu treffende Annahmen, wie die Festlegung auf ein Zeitraster für Einplanung der Aufträge und Produktion der benötigten Teile, sowie die eigentliche Produktion und die Distribution zeigen deutlich, dass dieses Lieferzeitziel sehr ehrgeizig gesetzt ist.

Abbildung 8.2 zeigt die Verteilung der Transportzeiten für Fahrzeuge von Kunden zweier ausgewählter Märkte: Deutschland und Skandinavien. Unmittelbar auffällig ist die Spreizung der Ergebnisse von minimal unter 24 Stunden (Null Tage in den Diagrammen) bis hin zu 40 Tagen. Eine Ursache für diese Spreizung sind offenbar die deutlich größeren Entfernungen, die zwischen Werk und Kunden zu überbrücken sind, allerdings können die im Maximum 40 Tage Transportzeit (Zeit zwischen Produktionsende und Ankunft beim Händler) so nicht erklärt werden.

Eine Betrachtung der einzelnen Fahrzeuge mit einer solchen langen Transportzeit zeigt, dass diese Fahrzeuge sehr lange auf den Transport warten mussten. Diese lange Wartezeit entsteht dadurch, dass es sehr lange dauert, bis genügend Fahrzeuge für einen LKW-Transport in die skandinavischen Märkte zusammenkommen. Obwohl das Marktvolumen der skandinavischen Märkte mit 0,58 % vergleichsweise gering ist, erscheint diese Wartezeit intuitiv nicht plausibel. Eine Erklärung hierfür findet sich, wenn die Verteilung der Marktvolumina auf die Werke betrachtet wird (siehe Tabelle B.2). Die weitaus meisten Fahrzeuge für die skandinavischen Märkte werden aufgrund der räumlichen Nähe im Werk Bremen produziert.

Dies ist jedoch aus Kapazitätsgründen nicht für alle Fahrzeuge möglich. Ist das Werk Bremen ausgelastet, werden auch die anderen Fertigungsstandorte genutzt, um Nachfragespitzen zu befriedigen. Da diese Ausweichstrategie nur in Spitzenzeiten genutzt wird, dauert es teilweise sehr lange, bis z.B. im Werk Prag eine ausreichende Anzahl Fahrzeuge für einen LKW-Transport in die betreffende Region bereitsteht.

Diese unbefriedigende Situation für skandinavische Kunden kann z.B. durch eine Änderung der Transportdisposition verbessert werden. Um das Verbesserungspotenzi-

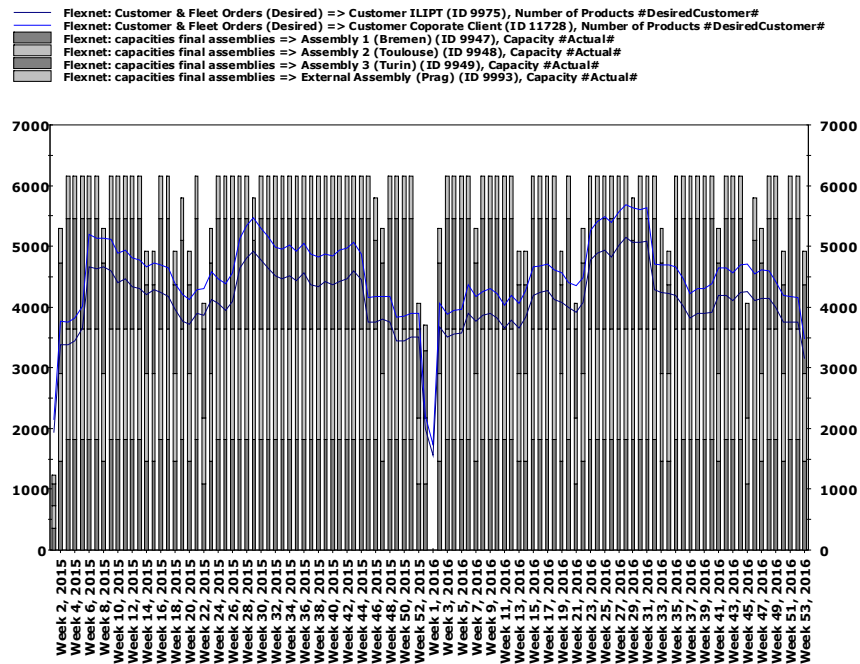


Abbildung 8.3.: Kapazitäten der Werke und Kundennachfrage

al aufzuzeigen, kann ein Experiment mit Transporten, die bereits dann starten, wenn lediglich ein Fahrzeug bereit steht, durchgeführt werden. Die maximale Lieferzeit für die skandinavischen Märkte kann so von 45 auf 2 Tage gesenkt werden¹¹.

Neben der Lieferzeit ist aus Sicht der Fahrzeughersteller die Werksauslastung ein wichtiges Kriterium, um die Kosten für die Realisierung einer solchen kurzen Lieferzeit abzuschätzen. Abbildung 8.3 zeigt die Kundennachfrage sowie die Werkskapazität, die im Experiment zur Verfügung stand, um diese Nachfrage zu befriedigen. Während ein Vergleich zwischen Kundennachfrage und Werkskapazität in Abbildung 8.3 vermuten lässt, dass stets ausreichend Kapazität zur Verfügung steht, zeigt die Darstellung der Werksauslastung in Abbildung 8.4, dass die reservierte Kapazität in Spitzenzeiten ausgelastet wird. Eine Reduktion der Kapazität¹² bringt folglich in diesen Spitzenzeiten eine Zunahme der Lieferzeit mit sich. Die Anzahl der Fahrzeuge, die mit einer Lieferzeit unter fünf Tagen (europäische Märkte) ausgeliefert werden können, sinkt von 51 % auf 38 %.

Ziel der Abbildung des Lieferantennetzwerks im Modell war es, ein Funktionieren dieses Netzwerkes unter den gesetzten Rahmenbedingungen zu zeigen. Zu diesen Rahmenbedingungen gehören zeitliche Vorgaben, die einzuhalten sind, um auf einen Abruf eines Kunden zeitgerecht reagieren zu können. Diese Kunden können dabei

¹¹Ein praxistauglicher Kompromiss könnte beispielsweise ein Transport sein, der Fahrzeuge unabhängig von der Anzahl dann abfährt, wenn im Auftragsbestand nicht genügend Fahrzeuge zu finden sind, um einen Transport in absehbarer Zeit mit voller Beladung starten zu können.

¹²Siehe Abbildung B.8.

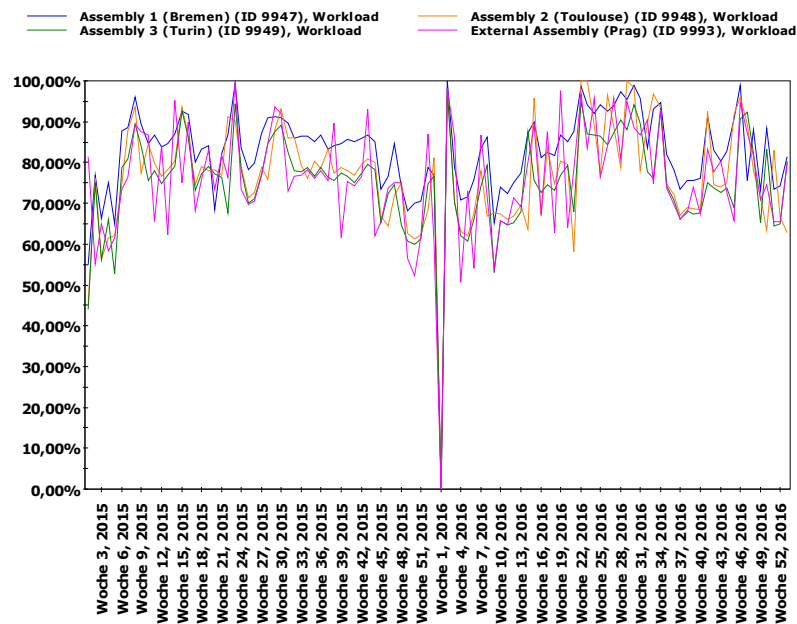


Abbildung 8.4.: Auslastung der Werke

sowohl Automobilhersteller für 1st-Tier Lieferanten als auch Zulieferer im Fall von 2nd-Tier Lieferanten sein. Konkret wird also untersucht, ob es gelingt, innerhalb von drei Tagen Teile vom 2nd-Tier Lieferanten zum 1st-Tier Lieferanten und dessen Produkte wiederum zum Fahrzeughersteller zu liefern. Nur wenn diese fristgerecht beim OEM eintreffen, kann dieser Fahrzeuge in der geplanten Reihenfolge fertigen.

Obwohl die Situation, dass Teile nicht fristgerecht geliefert werden, in der Simulation auftritt – verursacht z.B. durch »Verpassen« des gewünschten Zeitfensters für die Lieferung, wird dennoch ein gutes Ergebnis in Bezug auf die Lieferzeit erreicht. Störungen der Lieferanten sind im Modell aber nicht vorgesehen, d.h. es wird eine ausgezeichnete Versorgungstreue der Fertigung der Automobilhersteller angenommen.

Weitere Ergebnisse können der oben genannten Studie »Simulation-based Evaluation of Collaborative Execution Processes – D5.10« entnommen werden.

8.4. Weitere Anwendungsbeispiele

Das oben beschriebene Anwendungsbeispiel aus dem Projekt ILIPT zeigt die Eignung des in dieser Arbeit entwickelten Instrumentariums für die in Kapitel 3.1.1 beschriebene Aufgabenstellung. Aus den zahlreichen weiteren gelösten Aufgabenstellungen sollen im Folgenden einige weitere Beispiele kurz geschildert werden, in Bezug zu den entsprechenden Abschnitten aus Kapitel 3. Dies kann nicht für alle Fragestellungen aus Kapitel 3 geschehen, da die durchgeführten Projekte teilweise nicht veröffentlicht werden dürfen.

1. **Störungsmanagement**, Kapitel 3.3.2. Für Volkswagen de Mexico wurde eine

Simulationsstudie zum Serienanlauf des New Beetle Cabrio durchgeführt. Studieninhalt war u.a. die Entwicklung von Störungsstrategien; d.h. es wurde nach geeigneten Strategien gesucht, um Aufträge im Fall von Kapazitätsengpässen wieder in die Produktion einzuplanen (vgl. [SH02]).

2. **Werksprogrammplanung**, Kapitel 3.2.2. Von Volkswagen Nutzfahrzeuge wurde unter Nutzung des hier entwickelten Instrumentariums ein System entwickelt, welches bei der Planung von »machbaren« Werksprogrammen unterstützt (vgl. [Mey06]).
3. **Bedarfs- und Kapazitätsmanagement**, Kapitel 3.2.1. Für die Audi AG wurde ein Prototyp eines Systemes entwickelt, welches die Veränderung der Teilebedarfe in Folge von Programmänderungen berechnet.
4. **Distributionsplanung**, Kapitel 3.3.1. Für die Audi AG wurde ein Modell entwickelt, welches die Berechnung der Distributionskosten abhängig von veränderten Transportstrategien unterstützt (Vortrag WILL [Wil04b]).
5. **Gestaltung von Liefernetzwerken**. Für DaimlerChrysler wurde ein Simulationsmodell entwickelt, welches die strategische Logistikplanung für das globale Liefernetz für eine neue Motorengeneration (HDEP – Heavy Duty Engine Plattform) unterstützt (vgl. [Wel05]). Neben der Kapazitätsplanung wurden hier auch Aspekte der Distributionsplanung behandelt (Vermeidung von Flugtransporten).

Die genannten Beispiele zeigen neben der Eignung des hier entwickelten Ansatzes für zahlreiche Aufgabenstellungen auch die Variabilität des Ansatzes bzgl. der Detaillierung einzelner Prozessabschnitte. So wird im ILIPT-Modell in den Bereichen Produktbeschreibung, Abbildung der Werke der OEMs und der Abbildung der Lieferkette ein mittleres Abstraktionsniveau gewählt.

In dem für Volkswagen de Mexico entwickelten Simulationsmodell ist die Produktbeschreibung detaillierter als im ILIPT-Modell ausgeführt. Die Abbildung von Regeln des Störungsmanagements im Werk des Herstellers erfolgt sehr detailliert, aber es werden keine Zulieferer betrachtet und auch die Distribution wird nur rudimentär widergespiegelt.

Ein nochmals detaillierteres Produktmodell findet im Fall der Werksprogrammplanung Verwendung. Störungsregeln werden hier nur in abstrakter Form eingestellt, wie auch die Distribution der gefertigten Fahrzeuge. Die Abbildung der Werkskapazitäten bzw. der Fertigungsrestriktionen spielt jedoch eine sehr wichtige Rolle. Das Lieferantennetzwerk für das Werk wird nicht im Detail betrachtet.

Das Modell für das Bedarfs- und Kapazitätsmanagement enthält eine sehr komplexe Produktbeschreibung, alle weiteren Modellelemente sind nur minimal vorhanden. Das Modell zur Distributionskostenanalyse für Audi legt den Fokus auf die sehr detaillierte Abbildung der Distribution gefertigter Fahrzeuge mit zahlreichen Steuerungsregeln für die Transportmitteldisposition und vielen Kosteninformationen. Die Produktbeschreibung ist hier auf ein Minimum beschränkt.

Schließlich ist im Fall der Liefernetzgestaltung für DaimlerChrysler der Betrachtungsgegenstand ein Motor anstatt eines Fahrzeugs. Die Produktbeschreibung der Motoren ist von mittlerer Komplexität; hingegen ist die Abbildung der Kapazitätsverwaltung der Zulieferer sehr detailliert.

8.5. Laufzeitverhalten

Die Eingangsdaten des Simulationsmodells umfassen über 2.500 Objekte¹³ mit über 75.000 Attributen¹⁴. Damit ist das Modell in Bezug auf die Eingangsparameter eher den kleinen Modellen zuzurechnen. Zum Vergleich nachfolgend zwei Größenbeispiele für weitere Modelle: Ein Modell, mit dem die weltumspannenden Versorgungsstrukturen für eine Motorenfertigung eines großen LKW-Herstellers¹⁵ über die ganze Produktionsdauer der Motorenplattform untersucht wurden, enthält ca. 3.900 Objekte, aber über 2.100.000 Attribute. Bezogen auf die »Akteure« im Netzwerk – beide Modelle bilden Lieferanten- und Werksstrukturen detailliert ab – sind die Modelle gut vergleichbar; die ebenfalls ähnlich detaillierte Produktbeschreibung führt insgesamt auf eine nahezu identische Anzahl Objekte. Die hohe Anzahl Attribute erklärt sich durch die definierten alternativen Szenarien, die parallel im Modell gehalten werden sowie den langen Betrachtungszeitraum des Modells.

Ein Modell, in welchem eine detaillierte Produktbeschreibung eine große Rolle spielt¹⁶, zeigt ein anderes Bild. Hier beträgt die Anzahl der Objekte über 55.000, die Anzahl der Attribute ca. 1.000.000, bei nur einem Szenario. Die Anzahl der »Akteure« im Netzwerk ist auf der anderen Seite hier sehr klein. Zulieferer, Werke und Distributionsstrukturen sind auf ein Minimum reduziert, so dass in der Summe über 99 % der Modellgröße auf die detaillierte Produktbeschreibung zurückzuführen sind.

Diese sehr unterschiedliche Verteilung der Objektanzahlen auf Modellbereiche¹⁷ in den genannten Beispielen zeigt, dass durchgängig hohe Anforderungen an die Effizienz der Datenverarbeitung im Simulationswerkzeug zu stellen sind: Große Datenmengen können im Bereich der Produktbeschreibung ebenso auftreten wie im Bereich der Distributionsstrukturen.

Für das ILIPT-Modell beträgt die Laufzeit, bezogen auf die CPU-Zeit¹⁸, ca. 15 Min auf einem Rechner mit einem 3GHz Pentium IV Prozessor. Die Datenbank mit den Ergebnissen umfasst im Anschluss nahezu 1 GB. Verdoppelt man die Anzahl der Aufträge vier Mal hintereinander¹⁹, ergibt sich eine Berechnungszeit für ein also 16-fach vergrößertes Modell von knapp sechs Stunden (CPU-Zeit) und eine Datenbankgröße von 14,8 GB. Der maximale Hauptspeicherbedarf während der Berechnung ist 1,92 GB. Die Anzahl der Fahrzeug-Datensätze beträgt 7,68 Mio., die Anzahl der Eigenschaften in den Aufträgen 46 Mio. Trotz der nicht ganz linearen Zunahme der Berechnungszeit gelingt es also, auch sehr große Modelle in kurzer Zeit zu bewerten.

¹³Ermittelt als Anzahl Einträge der Tabelle `IN_Object`.

¹⁴Ermittelt als Anzahl Einträge der Tabelle `IN_Attribute`, addiert zur Anzahl Einträge der Tabelle `IN_Vector_Entry`.

¹⁵Daimler-Chrysler HDEP, siehe oben

¹⁶Audi Bedarfs- und Kapazitätsmanagement, siehe oben

¹⁷Bezogen auf die Eingangsdaten.

¹⁸Die Zeit, die der Prozessor mit der Modellbewertung beschäftigt war. Die insgesamt benötigte Zeit wird z.B. durch Wartezeiten auf Massendatenspeicher verlängert.

¹⁹Die Speicherausstattung des Testrechners erlaubte eine Berechnung größerer Modelle nicht. Signifikant größere Modelle erfordern eine (relativ unproblematische) Portierung des Berechnungswerkzeugs auf Rechnerarchitekturen mit einem größeren Adressraum als 4 GB. Solche Rechner sind auch im Consumer-Bereich verfügbar.

8.6. Zusammenfassung

Das vorgestellte Beispiel aus dem ILIPT-Projekt nutzt zahlreiche Eigenschaften des hier entwickelten Modellierungs- und Bewertungsansatzes. Die Fragestellung erfordert die Abbildung der Prozesse zwischen Kundenauftrag und Auslieferung des Fahrzeugs in einem Detaillierungsgrad, der es erlaubt, diejenigen dynamischen Effekte zu beobachten, die eine einfache Beantwortung der Frage nach der Machbarkeit einer Lieferzeit von fünf Tagen erschweren.

Die Prozessparameter sind so gewählt, dass eine Lieferzeit von fünf Tagen für einen sehr großen Teil Europas erreicht werden sollte. Als Beispiel seien hier die Kapazitäten der Werke genannt, die stets größer als der Marktbedarf sind. Die Lieferanten- und Transportkapazitäten sind ebenfalls so ausgelegt, dass in jeder Woche in Summe stets mehr Kapazität zur Verfügung steht, als bei statischer Betrachtung benötigt wird.

Die oben gezeigten Ergebnisse belegen, dass Unterschiede in den Schichtmodellen, unterschiedliche Urlaubstage in Ländern bzw. Unternehmen, Schwankungen der Kundennachfrage und Zeitverbräuche in der Informationsweiterleitung dazu führen, dass die durch die Simulation ermittelten Kennzahlen ein differenzierteres Bild zeichnen, als die allein statische Betrachtung. Die ermittelten Ergebnisse können auf die Ausgangsfragestellung zurückwirken: Ist eine Ausrichtung des betroffenen Netzwerks auf eine Lieferzeit von fünf Tagen überhaupt sinnvoll? Die dynamische Bewertung zeigt, welche Herausforderungen in den Teilprozessen hierfür zu bewältigen sind.

Im Anschluss an die hier vorgestellte eher »globale« Fragestellung nach der generellen Machbarkeit ergeben sich weitere Fragen, die in den Teilprozessen gestellt werden müssen. Für die Frage der generellen Machbarkeit wurde angenommen, dass eine Teileversorgung innerhalb von drei Tagen möglich ist. Vorbedingung für die vorgesehenen sehr kurzen Montagezeiten sind hier Modulkonzepte, die innerhalb des ILIPT-Projekts entwickelt wurden. Da der Einsatz solcher Modulkonzepte zu einem Ansteigen der Variantenanzahl in der Lieferkette führen kann, ist die Frage nach einer Machbarkeit von drei Tagen Durchlaufzeit für die Teileversorgung von ähnlicher Komplexität wie die hier beantwortete Frage und kann wiederum anhand eines Simulationsmodells untersucht werden.

Obwohl die Produktstruktur des »ILIPT Car« eher einfach gehalten ist und die Möglichkeiten des erarbeiteten Produktbeschreibungsansatzes nicht vollständig nutzt, wird dennoch deutlich, welchen Einfluss diese Beschreibung auf die Möglichkeiten des Ansatzes hat. Die Anteile der verschiedenen Baureihen, zusammen mit den Einbauraten für die Eigenschaften des Fahrzeugs, ergeben die »Systemlast« des Lieferantennetzwerkes. Lieferanten bestimmen über Stücklisten, welche für jeden Knoten des Netzwerkes im Modell parametrisiert sind, welche Teile von den jeweiligen Sublieferanten, in welcher Anzahl zu beziehen sind.

Eine detaillierte Abbildung der vier Werke innerhalb des Simulationsmodells ist erforderlich, um eine realistische Verteilung der Aufträge auf die Werke zu erhalten. Nur wenn die Märkte von den jeweils nächsten Werken beliefert werden, ist die erwünschte kurze Lieferzeit zu erreichen.

Ein wesentlicher Beitrag zur fristgerechten Auslieferung der Fahrzeuge wird durch die Lieferanten geleistet. Nur wenn der allergrößte Anteil der benötigten Teile zum gewünschten Zeitpunkt verfügbar ist, können die Fahrzeuge wie gewünscht gebaut werden. Das für die Teileproduktion vorgesehene Zeitfenster von drei Tagen ist dabei

insofern herausfordernd, als die Module des Fahrzeugs Build-to-Order bis hin zum 2nd-Tier Lieferanten gebaut werden sollen. Innerhalb des Simulationsmodells müssen sowohl die Informationsflüsse als auch die Materialflüsse so detailliert abgebildet werden, dass nachvollziehbar wird, ob die Prämissen für den Modellaufbau – hier die drei Tage für die auftragsbezogene Teileproduktion – im Jahr 2015 realisierbar sind.

Schließlich und nicht zuletzt ist es die Fahrzeugdistribution, die mit sehr kurzen Zeitspannen auskommen muss, um die Fahrzeuge von den Werken zu den Abnehmern zu befördern. Neben der grundsätzlichen Aussage über den Einfluss der Fahrzeugdistribution auf die Lieferzeit in den Märkten erlaubt das Modell auch Aussagen zu Problembereichen, die vielleicht nicht direkt im Fokus der Betrachtung stehen, aber dennoch gelöst werden müssen wie z.B. im Fall der Fahrzeuge, die in den Werken »stehen bleiben«, weil nicht genügend Fahrzeuge für die jeweilige Relation gefertigt werden.

Die Untersuchung der Umsetzbarkeit der in ILIPT entworfenen Planungsprozesse macht den Nutzen des hier vorgestellten Ansatzes deutlich:

- ▷ **Überprüfung des Prozessentwurfs.** Die Abbildung eines Prozessentwurfes in einem Simulationsmodell macht es erforderlich, exakt zu benennen, welche Schritte in einer bestimmten Planungssituation durchzuführen sind. Die Modellierung der ILIPT Prozessentwürfe zeigte z.B. Mängel im Bereich des »Exception Handling«, also der Behandlung unvorhergesehener Störfälle in der Lieferkette auf.
- ▷ **Ableitung von Anforderungen an die jeweiligen Teilprozesse.** Zum Beispiel können durch statische Berechnung festgelegte Zeitintervalle durch die dynamische Betrachtung präzisiert werden. Für die Herstellung der Teile, die für die Fahrzeugfertigung benötigt werden, wurden drei Tage veranschlagt. Aus Sicht der einzelnen Zulieferer kann nun bestimmt werden, wie viel Zeit für einzelne Bearbeitungsschritte zur Verfügung steht. Dieses Zeitmaß ist dabei kein fixer Wert über mehrere Monate hinweg, sondern schwankt dynamisch. Beeinflussende Faktoren sind hier z.B. Wochenenden, Feiertage (regional, national unterschiedlich), unterschiedliche Schichtlängen, Fahrzeitregelungen für Transporte etc.
- ▷ **Übergreifende Bewertung.** Ziel der Abbildung der ILIPT Prozesse in einem Simulationsmodell war insbesondere der Nachweis gesamthafter Erreichung der vorgegebenen Ziele. Konkret: Ist eine Lieferzeit von fünf Tagen für ein BTO-Fahrzeug in Europa möglich? Es konnte sowohl der gewünschte Nachweis erbracht als auch angegeben werden, welche Regionen mit welcher Lieferzeitverteilung beliefert werden können.
- ▷ **Unterstützung der Strategieentwicklung.** Die aufgezeigten Eigenschaften des Prozesses zeigen die Handlungsfelder der zukünftigen Arbeit auf: Die Prozesse sind z.B. im Bereich des Exception Handling detaillierter zu formulieren und auch die zeitlichen Vorgaben der Outbound Distribution sind zu überdenken, um eine ökonomisch wie auch ökologisch vertretbare Lösung zu finden. Hier zeigen sich wiederum die Vorteile der detaillierten Abbildung: Es ist sowohl die Integration von Exception Handling Regeln sinnvoll und möglich als auch die Entwicklung von Distributionsstrategien, wie z.B. die Nutzung alternativer Transportmittel (Bahn), für bestimmte Regionen und Zeitfenster.

9. Zusammenfassung und Ausblick

Zahlreiche Veränderungen im Umfeld der Automobilhersteller haben in der Vergangenheit und werden in der Zukunft die Anpassung von Prozessen erforderlich machen. Die zunehmende Komplexität der logistischen Netzwerke, in die die Automobilhersteller eingebunden sind und die sich verschiebende Machtbalance zwischen den Partnern im Netzwerk, z.B. durch die zunehmende Anzahl elektronischer Komponenten in einem Fahrzeug¹ sowie die abnehmende Fertigungstiefe führen zu Netzwerken, deren reibungsloses Funktionieren nicht mehr durch vertraute Vorgehensweisen sichergestellt werden kann.

So kann eine kapazitative Abhängigkeit von elektronischen Komponenten nicht beliebig durch Bestände entschärft werden, da die Vernetzung der Steuergeräte einen exakten Abgleich von Hard- und Softwareständen erfordert. Das heißt, nur ganz bestimmte Versionsstände passen zusammen. Bestände eines Steuergerätes können so schnell unbenutzbar werden. Die Verringerung von Beständen spart also nicht nur Kosten, z.B. in Form reduzierter Kapitalbindung, sondern reduziert auch das Risiko große Bestände verschrotten zu müssen.

Die verringerte »Neigung« der Lieferanten, ihre Kapazitätssteuerung dem Diktat der OEMs zu unterwerfen und die ebenfalls verringerte Möglichkeit durch Bestände die Kapazitäten von OEMs und Lieferanten zu entkoppeln, erzwingen Prozesse bei den OEMs, die diesen Faktoren Rechnung tragen. Es ist nahe liegend, dass diese neuen Prozesse es nicht mehr erlauben, allein durch die Planung des oder der OEMs das logistische Netzwerk zu steuern; die Kapazitäten sind in einem viel mehr dezentralen Ansatz zu steuern. Diese Dezentralität führt jedoch zu einer Dynamik der Netzwerke, die ohne rechnergestützte Modelle kaum mehr überschaubar ist.

Der im Rahmen dieser Arbeit entwickelte Ansatz erlaubt es, ganzheitliche Modelle von Auftragsabwicklungsprozessen in ihrer Einbettung in logistische Netzwerke zu untersuchen und so zu Entwürfen zu kommen, die den veränderten Rahmenbedingungen Rechnung tragen. In Kapitel 4 wurden Anforderungen definiert, die ein solcher Ansatz erfüllen sollte, um im oben skizzierten Bereich sinnvoll eingesetzt werden zu können. Eine wichtige Annahme für die Formulierung der Anforderungen ist es, dass es sinnvoll und möglich ist, Modelle auf einem so detaillierten Niveau abzubilden, dass jeder einzelne Auftrag mit seinen Eigenschaften von der Bestellung bis hin zur Auslieferung verfolgt werden kann. Anforderungen wurden u.a. in folgenden Bereichen gestellt (exemplarisch):

- ▷ **Systemlast.** Abbildung der Systemlast, so dass die relevanten Ressourcen im Modell realitätsnah belastet werden. Diese Anforderung umfasst zum einen die Produktbeschreibung und zum anderen die Strukturen, die die Nachfrage im Modell erzeugen (z.B. Händler).

¹Ein Automobilhersteller kann Druck auf einen Blechteilehersteller ausüben, aber kaum auf einen Chiphersteller.

- ▷ **Werke.** Die Aufträge als Bestandteil der Systemlast müssen entsprechend den algorithmischen Vorgaben aus dem Auftragsabwicklungsprozess auf Werke verteilt werden.
- ▷ **Zulieferer.** Die Anbindung der Lieferanten an das Werk muss so gestaltet sein, dass das Wechselspiel zwischen Kapazitätsplanung auf Vorhersagebasis und der Fertigung auf Basis tatsächlicher Nachfrage und tatsächlichen Kapazitäten eingefangen werden kann.
- ▷ **Distribution.** Die Distribution sowohl von Behältern (Inbound) bzw. Fahrzeugen (Outbound) zwischen Lieferanten, Werken und Händlern muss so gestaltet sein, dass die Einflüsse der Distribution (z.B. Sonntagsfahrverbot) auf Teile- und Fahrzeuglieferungen abgebildet werden können.
- ▷ **Modellgröße und Laufzeit.** Der Ansatz muss es erlauben, eine Bewertungskomponente so zu implementieren, dass Modelle in der erforderlichen Detaillierung erzeugt und bewertet werden können. Die Anzahl der Aufträge in einem solchen Modell kann viele Millionen betragen, wobei für jeden Auftrag wiederum eine Vielzahl von Informationen zu verarbeiten ist.

Eine Untersuchung prinzipiell geeigneter Methoden zur Bewertung von Auftragsabwicklungsprozessen, unter Berücksichtigung der genannten Anforderungen in Kapitel 5, zeigt, dass die dynamische ereignisorientierte Simulation günstige Eigenschaften für den Anwendungsbereich ausweist. Trotz der weiten Verbreitung dieser Methode zur Untersuchung vielfältiger Fragestellungen bringt die Verwendung etablierter, am Markt verfügbarer, Werkzeuge nur geringe Vorteile, aber gewichtige Nachteile, da nur sehr wenig der von diesen Werkzeugen gebotenen Funktionalität im Anwendungsfall »Simulation von Auftragsabwicklungsprozessen« einsetzbar ist.

Diesen geringen Vorteilen stehen gewichtige Nachteile gegenüber, wie z.B. das Risiko, dass die zu handhabenden Datenmengen zu einem Versagen des gewählten Werkzeugs führen. Daher wird in Kapitel 5 eine weit verbreitete »allgemeine« Programmiersprache (C++) für die Umsetzung des Bewertungswerkzeugs ausgewählt. Als Beschreibungssprache für die Modellierungselemente für Auftragsabwicklungsprozesse wird die Unified Modeling Language (UML) festgelegt.

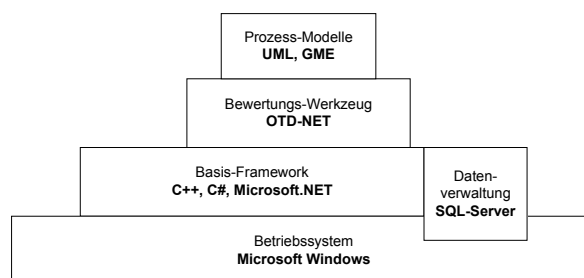


Abbildung 9.1.: Realisierungsumgebung für das Berechnungswerkzeug

Der in Kapitel 6 entwickelte Modellierungsansatz schafft eine enge Verbindung zwi-

schen in UML definierten Klassen und deren Implementierung in C++ und den Klassen, die vom Modellierungswerkzeug genutzt werden, um die Abbildung von Auftragsabwicklungsprozessen zu unterstützen (siehe Abbildung 9.2). Diese enge Verbindung bildet einerseits die Basis für eine sehr effiziente Bewertung mittels ereignisdiskreter Simulation, erlaubt aber auf der anderen Seite eine Modellierungsarbeit ohne Details der Implementierung zu kennen. Die eigentlichen Simulationsmodelle werden in einem datenbankgestützten Format oder alternativ in XML repräsentiert.

Damit ist sowohl einerseits die Möglichkeit eröffnet, ohne Kenntnisse der Implementierungsdetails Prozess-Modelle zu entwerfen und zu bewerten als auch auf der anderen Seite, durch die große strukturelle Ähnlichkeit zwischen den Klassenstrukturen der Datenbankmodelle und den UML-Strukturen, sehr leicht Erweiterungen der Abbildungsmöglichkeiten vornehmen zu können.

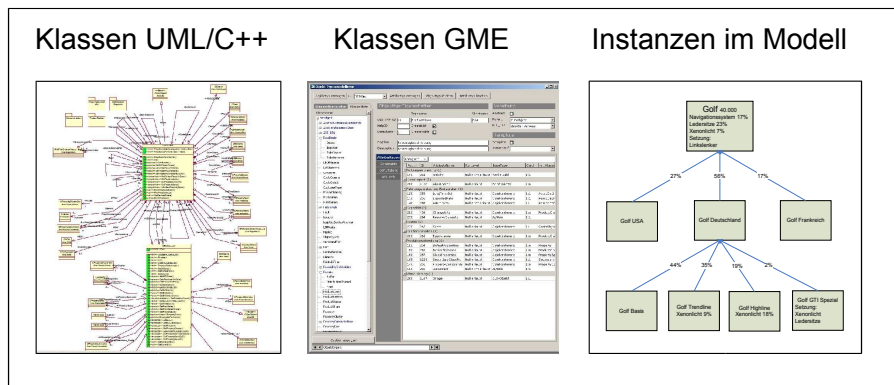


Abbildung 9.2.: Klassendefinition UML bzw. GME

Das eigentliche Bewertungswerkzeug ist ein Element einer Modellierungsumgebung bestehend aus einem Modelleditor (Graphical Modeling Environment – GME), dem Bewertungswerkzeug und einer Analysekomponente (OTD-Analyzer)². Die Arbeitsschritte der Modellerzeugung und Bewertung sind dann:

1. Erzeugung eines Simulationsmodells mittels der GME. Ergebnis ist ein Simulationsmodell in einem »objektrelationalen« Format. Sind große Mengen von Datenbankobjekten zu erzeugen, kann eine Schnittstelle zur Automatisierung dieses Prozesses genutzt werden.
2. Start des Bewertungswerkzeugs OTD-NET. Das Modell aus der Datenbank wird in den Hauptspeicher gelesen und so aufbereitet, dass alle Informationen, die im Vorfeld einmalig berechnet werden können, schnell zugreifbar zur Verfügung stehen. Die Bewertung des Modells wird durch ein schrittweises »Durchspielen« des Modells geleistet. Für die Analyse wichtige Schritte werden dabei in der Datenbank dokumentiert.
3. Aufbereitung der Ergebnisdaten und Analyse. Die Ergebnisdaten werden mehr-

²GME und OTD-Analyzer wurden nicht im Rahmen dieser Arbeit entwickelt, wenn auch mit dem vordringlichen Ziel OTD-NET-Modelle zu erzeugen und zu analysieren.

dimensional (OLAP) aufbereitet und können dann mittels des Analysewerkzeugs untersucht werden. Eine SQL-gestützte Auswertungsmöglichkeit wird ebenfalls geboten.

Die beispielhafte Erprobung des Bewertungswerkzeugs in Kapitel 8 an einem Beispiel aus dem EU Forschungsprojekt ILIPT zeigt, dass die formulierten Anforderungen an den Ansatz erfüllt werden. Ein wesentliches Charakteristikum des zugrunde liegenden Konzepts ist dabei seine Adaptierbarkeit an neue Fragestellungen, so dass auch zukünftigen Anforderungen Rechnung getragen werden kann. Neben dem vorgestellten ILIPT-Anwendungsfall wurden mit dem hier entwickelten Ansatz zahlreiche weitere Fragestellungen von mehreren Automobilherstellern untersucht. Diese Untersuchungen haben einen enger gefassten Fokus, als das hier vorgestellte ILIPT-Modell und zeigen somit nicht die Abdeckung des gesamten Auftragsabwicklungsprozesses von der Bestellung eines Fahrzeugs bis hin zur Auslieferung, belegen aber auf der anderen Seite, dass die detaillierte Abbildung auch in Teilprozessen verlässliche Aussagen zulässt.

Schwerpunkte dieser weiteren Untersuchungen waren z.B. die Distribution bereits gefertigter Fahrzeuge eines deutschen Automobilherstellers, die Möglichkeiten zur Umsetzung von Just-in-Sequence-Konzepten in der Motorenfertigung und die Bewertung eines global verteilten Lieferantennetzwerkes für die Motorenfertigung mit 1st- und 2nd-Tier Lieferanten. Die letzten beiden Beispiele zeigen, dass zwar der Entwicklungsschwerpunkt auf der Abbildung der Fahrzeugfertigung lag, die Konzepte aber sehr gut auf Auftragsabwicklungsprozesse anderer Produkte übertragbar sind.

Potenzielle Einsatzmöglichkeiten für OTD-NET und den zugrundeliegenden Ansatz wurden bereits in Kapitel 3 thematisiert und reichen von sehr langfristigen Planungen, die den gesamten Lebenszyklus eines Produktes abdecken bis hin zu Fragestellungen, die im operativen Geschäft angesiedelt sind. Der Ansatz ist für alle in Kapitel 3 skizzierten Einsatzmöglichkeiten sehr gut geeignet. Für einen Einsatz im operativen Planungsbetrieb sind zweifellos zahlreiche unternehmensspezifische Anforderungen zu finden, die von der vorliegenden Implementierung des Werkzeugs nicht erfüllt werden. Insbesondere das Thema Sicherheit wurde im Rahmen der Implementierung nur rudimentär betrachtet.

Neben den vielen bereits dargestellten Einsatzmöglichkeiten besteht eine Chance darin ein übergreifendes Modell zu erzeugen, welches Informationen aus den Teilbereichen des Auftragsabwicklungsprozesses zusammenfasst. Planungen für Motoren und Getriebe hängen nahe liegend stark von den geplanten Volumina für die jeweiligen Baureihen zusammen, in denen diese Aggregate verbaut werden. Die offensichtliche wechselseitige Beeinflussung dieser Planungen macht ein integriertes Planungsmodell, welches Restriktionen aus beiden Bereichen enthält, lohnenswert. Eine Erweiterung eines solchen Modells um weitere »Heavy Items«, also als kritisch bekannte Teile, führt zu einem Modell, welches verlässliche Daten z.B. für die Transportmittelplanung bereitstellen kann.

Werden Informationen über die Distributionswege der Fahrzeug-Baureihen hinzugefügt, wird es möglich, eine hochwertige Prognose über die Fahrzeugvolumina abzugeben, die z.B. in Verschiffungshäfen ankommen. Eine ständige Aktualisierung der Programmplanung aus den operativen Systemen der Fahrzeughersteller erlaubt eine stets aktuelle Prognose über die benötigten Ressourcen im Distributionsbereich. Händ-

ler können so z.B. mit maximal einem Tag Verzögerung darüber informiert werden, dass durch einen Kapazitätsengpass bei einem Zulieferer die Auslieferung einer neuen Baureihe verspätet erfolgt.

Zusammenfassend kann festgestellt werden, dass die Nutzung sehr detaillierter, über den gesamten Auftragsabwicklungsprozess integrierter Planungsmodelle ein vielversprechender Weg ist, zahlreichen Aufgabenstellungen im Bereich der Automobilindustrie zu begegnen.

Abbildungsverzeichnis

1.1. Vergleich alternativer Konzepte im Auftragsabwicklungsprozess	5
2.1. Zentrale Unternehmensprozesse in der Automobilindustrie (Quelle: [BW00, S. 7])	7
2.2. Prozessketten im Bereich Auftragsabwicklungsprozess (vgl. z.B. [Gra01, S. 75], [Sta01b, S. 48])	8
2.3. Lagerorientierte (Build-To-Stock) vs. auftragsorientierte (Build-To-Order) Bestellabwicklung	10
2.4. Übersicht: Prozesse der Auftragsabwicklung und Umfeld	13
2.5. Terminologie Produktaufbau (beispielhafte Zuordnung zu Ebenen): Baureihe, Marktausführung, Karosserieform, Modell, Eigenschaft . .	14
2.6. Auftragsmanagement (vgl. [HP04, S. 26])	16
2.7. Anteil Kundenaufträge (vgl. [Mey04b, S. 9])	17
2.8. Teilprozesse der Auftragseinplanung (vgl. [HP04, S. 29])	19
2.9. Kennzahlen im Auftragsabwicklungsprozess	27
2.10. Kennzahlen im Auftragsabwicklungsprozess am Beispiel Volkswagen (Quelle: [Hoo04, S. 16])	27
2.11. Kennzahlenkategorien (Quelle: [Kel06, S. 16])	28
3.1. Bestandsprofil über die Supply Chain der Automobilhersteller (vgl. [HP04, S. 61])	32
3.2. Das »atmende Unternehmen« (vgl. [Har96, S. 38])	33
3.3. Konstruktionsregeln am Beispiel	41
4.1. Grundsätze ordnungsmäßiger Modellierung (GoM I)	54
4.2. Anteile von Eigenschaften auf Märkten	62
5.1. Events, Aktivitäten, Prozesse (Quelle: [Pag91, S. 27])	79
5.2. Prozesskettenelement des Dortmunder Prozesskettenparadigmas (angelehnt an: [Kuh95, S. 47] und [WQ97, S. 18])	83
5.3. Business Process Definition Meta-Model der Object Management Group (Quelle: [OMG04, S. 2])	84
5.4. Klassifizierung von Simulationswerkzeugen (Quelle: [WN00, S. 426])	89
6.1. Klassenstruktur Prozesse	97
6.2. Klassenstruktur Planung/Produktbeschreibung	98
6.3. UML-, GME-Klassen und Objekte im Zusammenhang	99
6.4. Ableitung von Produktklassen	105
6.5. Aktivitätsdiagramm Auftragserzeugung (vereinfachte Darstellung) . .	110
6.6. Modellelemente Märkte und Händler	117

6.7. Statuspunkte Auftragseinplanung	118
6.8. Einsatz von Ressourcenobjekten	122
6.9. Modellelement »Werk«	125
6.10. Bestimmung des Einplanungsintervalls	128
6.11. Platzorientierte Ressourcenverwaltung	130
6.12. Modellelement Zulieferer	133
6.13. Modellelemente Distribution	139
6.14. Beispiel Distributionsnetzwerk	143
7.1. Gesamtsystem zur Modellerstellung, Bewertung und Analyse	152
7.2. Objektrelationale Tabellenstruktur	154
7.3. Strukturierung Simulationsmodell, UML-Komponenten und Simulationswerkzeug	156
7.4. Werkzeug zur Klassendefinition	158
7.5. Kennzahlenkategorien (Quelle: [Kel06, S. 16], vergr. auf Seite 28)	164
7.6. Schritte der Durchführung einer Simulationsstudie (Quellen: [Law03, S. 67] und [Sar98, S. 123])	166
8.1. Produktstruktur Beispielmodell	173
8.2. Transportzeit im Vergleich	176
8.3. Kapazitäten der Werke und Kundennachfrage	177
8.4. Auslastung der Werke	178
9.1. Realisierungsumgebung für das Berechnungswerkzeug	184
9.2. Klassendefinition UML bzw. GME	185
A.1. C++-Klassen im Entwicklungssystem Visual Studio 2005	216
A.2. Werkzeug für die Bearbeitung von UML-Modellen – Rational Rose	217
B.1. Modellstruktur	218
B.2. Montagestandorte	219
B.3. Modulstruktur Fahrzeug	221
B.4. Stücklisten Motoren	221
B.5. Liefernetzstruktur Motoren	222
B.6. Liefernetzstruktur Front-Modul	223
B.7. Durchschnittliche Lieferzeit Scandinavien und Deutschland	224
B.8. Kapazitäten der Werke und Kundennachfrage	224
B.9. Abweichung Einbauraten Soll/Ist	225
D.1. Klassenstruktur CSimObject	228
D.2. Klassenstruktur CExperiment	229
D.3. Klassenstruktur CSimParameters	230
D.4. Zentrale Klassen der Produktbeschreibung	231
D.5. Klassenstruktur Produktbeschreibung (CBOStructure und CBOClass)	232
D.6. Klassenstruktur Eigenschaftenfamilien (CPropertyCluster)	233
D.7. Klassenstruktur Eigenschaftenspezifikationen (CPropertySpec)	234
D.8. Klassenstruktur Stückliste (CBOM)	235

D.9. Klassenstruktur Auftragsanpassung (COrderModifier)	236
D.10. Klassenstruktur Distributionssystem (CDistributionSystem) . .	237
D.11. Klassenstruktur Distribution (CDistributionChannel)	238
D.12. Klassenstruktur Transportrelation (CTransportRelation)	239
D.13. Klassenstruktur Routing-Tabelle (CRoutingTable, CRouting) .	240
D.14. Klassenstruktur Logistikdienstleister (CLogisticsServiceProvider)	241
D.15. Klassenstruktur Transportmittel (CTransport)	242
D.16. Klassenstruktur Behälter (CContainer)	243
D.17. Klassenstruktur Distribution (CRouteInformation)	244
D.18. Klassenstruktur Transportauftrag (CTransportOrder)	245
D.19. Klassenstruktur Markt (CMarket)	246
D.20. Klassenstruktur Händler (CDealer)	247
D.21. Klassenstruktur Händlertypen (CDealer, CDomesticDealer, CImporter)	248
D.22. Klassenstruktur Kundentypen (CCustomerType)	249
D.23. Klassenstruktur Planungsparameter Händler (CPlanningDealer) .	250
D.24. Klassenstruktur Ressourcen (CResource)	251
D.25. Klassenstruktur Ressourcen-Blatt (CResourceLeaf)	252
D.26. Klassenstruktur Wertverlauf-Ressourcen (CResourceAssocData)	253
D.27. Klassenstruktur Wochenressource (CResourceWeekTime)	254
D.28. Klassenstruktur Prozess (CProcess)	255
D.29. Klassenstruktur Werk (CPlant)	256
D.30. Klassenstruktur Störungen (CFailure)	257
D.31. Klassenstruktur Suchmuster (CProductPattern)	258
D.32. Klassenstruktur Wahrscheinlichkeitsverteilung (CProbabilityDistribution)	259
D.33. Klassenstruktur Auftragsverwaltung (COrderBank)	260
D.34. Klassenstruktur Prozesssteuerung (CPlanning)	261
D.35. Klassenstruktur Zulieferer (Basis) (CSupplier)	262
D.36. Klassenstruktur Zulieferer (BTO) (CSupplierBTO)	263
D.37. Klassenstruktur Zulieferer (BTS) (CSupplierBTS)	264
D.38. Klassenstruktur zur Verwaltung von Abnehmern der Zulieferer (BTO) (CSupplierCustomer)	265
D.39. Klassenstruktur Wertverlauf (CAssocData)	266
D.40. Klassenstrukturen weiterer Wertverläufe	267
D.41. Klasse CSchedule	268

Literatur

- [Aab04] Aaby, Anthony A.: *Introduction to Programming Languages*. Walla Walla College, 2004. <http://www.cs.wwc.edu/~aabyan/Logic/index.html>.
- [Acc02] Accenture: *Automotive Order-to-Delivery (OTD): Leveraging Locate-to-Order and Build-to-Order to Unlock Value*. Accenture, Mai 2002. http://www.accenture.com/xd/xd.asp?it=enweb&xd=industries%5Cproducts%5Cautomotive%5Cauto_otd.xml.
- [Ace97] Acel, P.: *Methode zur Durchführung betrieblicher Simulationen – effiziente Optimierung der diskreten Simulation*. ARGESIM / ASIM Verlag, 1997.
- [Ada98] Adam, Dietrich: *Produktions-Management*. Gabler Verlag, 1998. 9. Auflage.
- [ADFS02] Alkire, Kara F., Jeffrey Davis, Michael S. Flynn, and Maitreya Kathleen Sims: *Order-to-delivery – fertile ground for improvement*. Technical report, Office for the Study of Automotive Transportation University of Michigan Transportation Research Institute, 2002. <http://www.osat.umich.edu/OTD.doc>.
- [AM02] Anderson, Edward G. and Douglas J. Morrice: *Capacity and backlog management in queuing-based supply chains*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/175.pdf>.
- [AS00] Arief, L.B. and N.A. Speirs: *A uml tool for an automatic generation of simulation programs*. Technical Report TR-731, Department Computing Science, University of Newcastle, 2000.
- [ATH05] ATHENA: *What is Enterprise Modelling*. Athena European Integrated Project, 2005. <http://www.athena-ip.org>.
- [AU92] Aho, Alfred V. and Jeffrey D. Ullman: *Foundations of Computer Science*. Computer Science Press, 1992.
- [Aut05] Automobilwoche: *VW arbeitet am günstigen »Welt-Auto«*. Automobilwoche, (12), 2005. Autorenkürzel »kro«.

- [BA03] Becker, J. und L. Algermissen: *Grundsätze ordnungsmäßiger Modellierung - Über Konstruktivisten, Handels-Hs und Referenzmodelle*. 2003. Proceedings of the Informatiktage 2003, Bad Schussenried. 2003.
- [Bal98] Balzert, Helmut: *Lehrbuch der Softwaretechnik: Software Management, Software Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, 1998.
- [Bal99] Balzert, Helmut: *Lehrbuch Grundlagen der Informatik: Konzepte, Notationen in UML, Java, C++, Algorithmik und Software-Technik*. Spektrum Akademischer Verlag, 1999.
- [Bal05] Balzert, Helmut: *Lehrbuch der Objektmodellierung*. Elsevier, Spektrum, Akademischer Verlag, 2005, ISBN 3-8274-1162-9.
- [Bau03] Baumgarten, H.: *Fünf Thesen zur erfolgreichen Gestaltung von Supply Chain Collaboration*. BVL – Bundesvereinigung Logistik, 2003. Präsentation Beilage (Compact Disc) in » Supply Chain Collaboration«.
- [Bau04] Baumgärtel, Hartwig: *Gestaltung und Bewertung neuartiger Informationsprozesse im Zuliefernetzwerk der Automobilindustrie*. In: *Vortrag auf der Fachtagung »Heute bestellt, morgen geliefert: Zukünftige Anforderungen an die Order-to-Delivery-Prozesse der Automobilindustrie« am 22.4.04*. Veranstalter: Fraunhofer IML, 2004.
- [BB99] Bause, Falko and Heinz Beilner: *Intrinsic problems in simulation of logistic networks*. In *Simulation in Industry*, pages 26–28, Oktober 1999. 11th European Simulation Symposium and Exhibition (ESS99).
- [BBF⁺02] Bause, F., H. Beilner, M. Fischer, P. Kemper, and M. Voelcker: *The procc/b toolset for the modelling and analysis of process chains*. In Field, T., P.G. Harrison, J. Bradley, and U. Harder (editors): *Computer Performance Evaluation, Modelling Techniques and Tools*, pages 51–70. Springer LNCS 2324, 2002. Proc. of the 12th Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (TOOLS 2002).
- [BBH⁺01] Baker, Dennis, Donald Bridges, Regina Hunter, Gregory Johnson, Joseph Krupa, James Murphy, and Ken Sorenson: *Guidebook to Decision-Making Methods*. Department of Energy, USA, 2001.
- [BBK01] Bause, F., H. Beilner und P. Kemper: *Zur prozessorientierten Modellierung von Logistiknetzwerken*. In: Enderlein, H., J. Petermann und S. Wirth (Herausgeber): *Tagungsband zur 2. Chemnitzer Netztagung »Vernetzt planen und produzieren VPP 2001«*, Seiten 65–68, Chemnitz, 2001. TU Chemnitz, SFB 457, 20.-21. September.
- [BBS03] Bause, Falko, Heinz Beilner und Mathias Schwenke: *Semantik des ProC/B-Paradigmas*. Technischer Bericht 03001, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, März 2003.

- [BBT⁺99] Beilner, H., F. Bause, H. Tatlitürk, A. van Almsick und M. Völcker: *Zum B-Modellformalismus – Version B1*. Technischer Bericht 99002, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, Juni 1999.
- [BCG⁺98] Brainerd, Walt, Ron Cytron, Ralph E. Griswold, Glenn Grotzinger, Dennis M. Ritchie, and Steve Summit: *Handbook of Programming Languages : Volume II Imperative Programming Languages*. MacMillan Publishing Company, 1998.
- [BCN96] Banks, J., J. S. Carson, and B. L. Nelson: *Discrete-Event Simulation*. Prentice-Hall, 1996.
- [BDKR01] Baron, C.P., U. Dietel, D. Kreppenhofer und M. Rabe: *Handlungsanleitung Simulation*. In: *Handlungsanleitung Simulation in Produktion und Logistik*, San Diego, 2001. SCS International.
- [Bec95] Becker, J.: *Die Grundsätze ordnungsmäßiger Modellierung und ihre Einbettung in ein Vorgehensmodell zur Erstellung betrieblicher Informationsmodelle*. 1995. <http://www.wi-inf.uni-duisburg-essen.de/MobisPortal/pages/rundbrief/pdf/Beck98.pdf>.
- [BFK01] Bause, F., M. Fischer, and P. Kemper: *Performance and cost analysis of supply chain models*. In *Proc. of the SeoulSim 2001 Conference*, pages 425–434, Oktober 2001.
- [BG97a] Beitz, Wolfgang und Karl Heinrich Grote (Herausgeber): *Taschenbuch für den Maschinenbau*. Springer Verlag, 1997. 19. Auflage.
- [BG97b] Braun, Jochen und Frank Gehr: *Prozessgestaltung in der Versorgungslgistik – der kooperative Ansatz eröffnet neue Perspektiven*. Band 12 der Reihe *IM – Die Fachzeitschrift für Information Management & Consulting*, Sonderheft 1997 „Business Engineering“, Seiten S. 26–32. IMC GmbH, November 1997.
- [Bie03] Bierwith, Thomas: *Virtuelle Logistikplanung für die Automobilindustrie*. Dissertation, Technische Universität Clausthal, 2003.
- [BJ03] Baumgaertel, Hartwig and Ulrich John: *Combining agent-based supply net simulation and constraint technology for highly efficient simulation of supply networks using aps systems*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1765–1773, Baltimore, 2003. INFORMS Simulation Society.
- [Ble79] Bleibohm, Gunter: *Auftragsabwicklung, maschinelle Stücklistengenerierung und Baubarkeitsprüfung in einer variantenreichen Fertigung, dargestellt am Beispiel einer Omnibusproduktion*. Dissertation, Technische Universität Braunschweig, 1979.

- [BM02] Bause, Falko und Christian Möller: *Kapazitätsbestimmung eines Shuttle-Zuges mit Hilfe der Warteschlangentheorie*. Distribution (Logistik in Warenfluss und Verteilung), (4):51–53, 2002.
- [Boo94] Booch, Grady: *Object-oriented analysis and design with applications*. The Benjamin/Cummings Publishing Company, Redwood City, Menlo Park, Reading, 1994. 2. Auflage.
- [BRJ99] Booch, G., J. Rumbaugh, and I. Jacobson: *The Unified Language User Guide*. Addison-Wesley, 1999.
- [Bro02] Brockhaus: *Der Brockhaus Computer und Informationstechnologie*. F.A. Brockhaus, 2002.
- [Bro06] Brockhaus: *Brockhaus - Die Enzyklopädie: in 30 Bänden*. F.A. Brockhaus, 2006. 21., neu bearbeitete Auflage.
- [BRS95] Becker, Jörg, Michael Rosemann und Reinhard Schütte: *Grundsätze ordnungsmäßiger Modellierung*. Wirtschaftsinformatik, 37(5), 1995.
- [Bru04] Bruckner, Werner: *Änderungswünsche am Auto bringen die Fertigungsplanung auf Hochtouren*. VDI Nachrichten, (11.06.2004):30, 2004.
- [BS04] Becker, J. und R. Schütte: *Handelsinformationssysteme*. Moderne Industrie, 2004. 2. Auflage.
- [BSGI00] Becker, J., R. Schütte, T. Geib und H. Ibershoff: *Grundsätze ordnungsmäßiger Modellierung (GoM)*. Technischer Bericht, Institut für Wirtschaftsinformatik, Westfälische Wilhelms-Universität Münster, IDS Scheer AG, Josef Friedr. Bremke & Hoerster GmbH & Co., 2000. Schlußbericht.
- [BW00] Baumgarten, Helmut und Stephan Walter: *Trends und Strategien in der Logistik 2000+ – Eine Untersuchung der Logistik in Industrie, Handel, Logistik-Dienstleistung und anderen Dienstleistungsunternehmen*. Technische Universität Berlin, 2000.
- [BWL⁺03] Bock, Dieter, Ulrich Weingarten, Matthias Laforsch, Timo Langemann und Thorsten Breithor: *Supply Chain Collaboration*. BVL – Bundesvereinigung Logistik, 2003.
- [BWZ02] Baumgarten, Helmut, Hans Peter Wiendahl und Joachim Zentes: *Logistik-Management: Strategien – Konzepte – Praxisbeispiele*. Springer Verlag, 2002.
- [BY97] Bar-Yam, Yaneer: *Dynamics of Complex Systems*. Westview Press, 1997.
- [CB02] Chwif, Leonardo and Marcos Ribeiro Pereira Barretto: *Supply chain analysis: Spreadsheet or simulation?* In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002.

- [Chu04] Chung, Christopher A.: *Simulation Modeling Handbook*. CRC Press, 2004.
- [CIM05] CIMOSA: *Business Process Modelling and Standardisation*. CIMOSA Association e.V., 2005. www.cimosa.de.
- [CK04] Cuntz, Nicolas und Ekkart Kindler: *On the semantics of EPCs: Efficient calculation and simulation*. In: *EPK 2004 / Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, Seiten 7–26. Gesellschaft für Informatik, Oktober 2004.
- [CMM67] Conway, Richard W., William L. Maxwell, and Louis W. Miller: *Theory of Scheduling*. Addison-Wesley, 1967.
- [Cor03] Corporation, Oracle: *Oracle olap : Application developer's guide*. Internet-Dokument, 2003. http://download-uk.oracle.com/products/bi/pdf/9204_olap_appdev_guide.pdf.
- [CV03] Chen, David and François B. Vernadat: *Enterprise interoperability: A standardisation view*. In *Enterprise inter- and intra-organizational integration – Building International Consensus*. Kluwer Academic Publishers, 2003. IFIP TC5/WG5 12. International Conference on Enterprise Integration and Modeling Technology (ICEIMT'02).
- [CW01] Conti, Robert F and Malcolm Warner: *A customer-driven model of job design: Towards a general theory*. Technical report, University of Cambridge – Judge Institute of Management, 2001.
- [Czi89] Czichos, Horst (Herausgeber): *Die Grundlagen der Ingenieurwissenschaften*. Springer Verlag, 1989. 29. Auflage.
- [Dai04] DaimlerChrysler04: *DaimlerChrysler Global Ordering*. s d & m – Software Design & Management, 2004. Projektinformationsblatt http://www.capgemini.com/hightechauto/automotive/SS_DaimlerChrysler.pdf.
- [DBD03] Dalal, Malay A., Henry Bell, and Mike Denzien: *Initializing a distribution supply chain simulation with live data*. In *Proceedings of the 2003 Winter Simulation Conference*, Baltimore, 2003. INFORMS Simulation Society.
- [DD97] Date, C.J. and Hugh Darwen: *The SQL Standard*. Addison-Wesley, 1997. Fourth Edition.
- [Díe04] Díez, Ana Belén García: *Deliverable d.a1.1.1 – first version of state of the art in enterprise modelling techniques and technologies to support enterprise interoperability work package – a1.1*. Technical report, ATHENA Project, Project No. 507849, 2004. »Public Deliverable« des ATHENA Projektes, Ana Belén García Díez ist »Document Owner«.

- [DF92] Date, C. J. and Ronald Fagin: *Simple conditions for guaranteeing higher normal forms in relational databases*. ACM Transactions on Database Systems, 17(3):465–476, 1992. <http://www.almaden.ibm.com/cs/people/fagin/tods92.pdf>.
- [DFK⁺02] Duarte, Brett Marc, John W. Fowler, Kraig Knutson, Esma Gel, and Dan Shunk: *Parameterization of fast and accurate simulations for complex supply networks*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/176.pdf>.
- [DV03] Dilling, Christian und Marcus Völker: *Beispielmodellierung eines Güterverkehrszentrums im ProC/B-Paradigma*. Technischer Bericht 03016, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.
- [ea95] al., V. Claus et: *Teubner-Taschenbuch der Mathematik – Teil II*. B.G. Teubner, 1995. 7. Auflage.
- [Eco] Economist: *Fighting back – what carmakers are doing to counter relentless competition*. The Economist. Survey: Car Industry, 2. September 2004.
- [EHP94] Ernest H. Page, Jr.: *Simulation Modeling Methodology: Principles and Etiology of Decision Support*. PhD thesis, Virginia Polytechnic Institute and State University, 1994.
- [Eli02] Elias, Simon: *New car buyer behaviour*. Technical report, Cardiff University, April 2002.
- [Eng88] Engesser, Hermann: *Duden Fremdwörterbuch*. Dudenverlag, 1988.
- [EP00] Eriksson, Hans Erik and Magnus Penker: *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons, 2000.
- [Eve95] Eversheim, Walter: *Prozessorientierte Unternehmensorganisation – Konzepte und Methoden zur Gestaltung »schlanker« Organisationen*. Springer Verlag, 1995.
- [EWW00] Elias, Simon, Ben Waller, and Geoff Williams: *Downstream distribution big picture mapping – an analysis of the generic current customer order-fulfilment system from sale to manufacturer, and manufacturer to delivery*. Technical report, Lean Enterprise Research Centre, ICDP, 2000. <http://www.3daycar.com/mainframe/publications/library/Downstreamap.pdf>.
- [FD03] Fischer, M. und Ch. Dilling: *Analytisch-numerische Techniken zur Lagerbestand-Analyse unter Berücksichtigung einer zeitlich-variierenden Belastung*. Technischer Bericht 03013, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.

- [Feu02] Feurer, Rainer: *Status of Built to Order Strategy and Time Quality within BMW Group*. Automotive News Europe, November 2002. <http://europe.autonews.com/files/Feurer.pdf>.
- [Fis73] Fishman, Georg S.: *Concepts and Methods in discrete event digital simulation*. Wiley-Interscience publication, 1973.
- [Fis03] Fischer, Markus: *Numerische Verfahren zur quantitativen Analyse von ProC/B und Petri Netz Modellen*. Technischer Bericht 03012, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.
- [FKWT03] Fischer, Markus, Peter Kemper, Zenghui Wu und Carsten Tepper: *Abbildung von ProC/B nach Petri-Netzen – Version 2*. Technischer Bericht 03011, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.
- [Fül04] Fülöp, János: *Introduction to decision making methods*. In *Workshop on Biodiversity & Ecosystem Informatics, Dec. 13-15*, Olympia, Washington, 2004. The Evergreen State College.
- [Fle02] Fleischmann, Bernhard: *Begriffliche Grundlagen (A 1.1), Systeme der Transportlogistik (A 1.2)*. In: Arnold, Dieter, Heinz Isermann, Axel Kuhn und Horst Tempelmeier (Herausgeber): *Handbuch Logistik*. Springer Verlag, 2002.
- [FP97] Frank, Ulrich und Michael Prasse: *Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen – veranschaulicht am Beispiel von OML und UML*. Technischer Bericht 6, Institut für Wirtschafts- und Verwaltungsinformatik der Universität Koblenz-Landau, September 1997.
- [FPB95] Frederick P. Brooks, Jr: *The Mythical Man-Month*. Addison-Wesley, 1995. Anniversary Edition.
- [FvL03] Frank, Ulrich und Bodo L. van Laak: *Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen*. Technischer Bericht 34, Institut für Wirtschafts- und Verwaltungsinformatik der Universität Koblenz-Landau, Januar 2003.
- [GAvV00] Gordijn, Jaap, Hans Akkermans, and Hans van Vliet: *Business modelling is not process modelling*. In *Conceptual Modeling for E-Business and the Web*, pages 40–51, London, 2000. Springer Verlag. Proceedings of the Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling.
- [GDM03] Gebhardt, Jörg, Heinz Detmer, and Anders L. Madsen: *Predicting parts demand in the automotive industry – an application of probabilistic graphical models*. 2003.

- [Gel02] Gelbmann, Gerhard: *An outline of pragmatologic model-theory (sec. stachowiak), semiotic subjectivity ii*. 2002. Second Lecture by Dr. Gerhard GELBMANN, guest researcher at the WAB (Wittgenstein Archives at the University of Bergen), held at the HIT Centre, 3rd June 2002, 15:15-17:00.
- [Gem05] Gemini, Cap: *Cars Online 04/05 – Driving Growth through Collaboration*. Capgemini, 2005. <http://www.capgemini.com/hightechauto/automotive/CapgeminiCarsOnline0405.pdf>.
- [GHIMM87] Gal, T., R. Horst, H. Isermann und H. Müller-Merbach: *Grundlagen des Operations Research - Band 1*. Springer Verlag, 1987.
- [GNS03] Ganapathy, Subhashini, S. Narayanan, and Krishnamurthy Srinivasan: *Simulation based decision support for supply chain logistics*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1013–1020, Baltimore, 2003. INFORMS Simulation Society.
- [Gra00] Graf, Hartmut: *Beschaffungslogistik in einem Fahrzeugmontagewerk mit E-Business*. VDI Berichte, (1571):5–17, 2000.
- [Gra01] Grafen, Barbara: *Prozeßorientierte Auftragsabwicklung in der Automobilindustrie*. Dissertation, Philipps-Universität Marburg, 2001.
- [Gru05] Gruppe, MAN Nutzfahrzeuge: *Logistik-Leitfaden*. Internet-Dokument, 2005. http://www.man-mn.com/datapool/mediapool/101/Logistikleitfaden_d.pdf.
- [Grz02] Grzan, Stjepan: *Enabling technology for an indoor location aware information system*. Diplomarbeit, Universität Stuttgart, Stuttgart, 2002.
- [GSS02] Garcia, Eduardo Saggiaro, Caio Fiuza Silva, and Eduardo Saliby: *A simulation model to validate and evaluate the adequacy of an analytical expression for proper safety stock sizing*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/172.pdf>.
- [Gus02] Gustafson, David: *Theory and Problems of Software Engineering*. McGraw-Hill, 2002.
- [Har96] Hartz, Peter: *Das atmende Unternehmen. Jeder Arbeitsplatz hat einen Kunden*. Campus Verlag, 1996.
- [HC94] Hammer, Michael und James Champy: *Business Reengineering – Die Radikalkur für das Unternehmen*. Campus Verlag, 1994.

- [HG00] Holweg, Matthias and Anthony Greenwood: *Product variety, life cycles, and rate of innovation – trends in the uk automotive industry*. Technical report, Cardiff University, Juli 2000. <http://www.3daycar.com/mainframe/publications/library/ProductVariety.pdf>.
- [HJ00] Holweg, Matthias and Professor Daniel T. Jones: *The challenge of building cars to order – can current automotive supply systems cope?* Technical report, Cardiff University, April 2000. <http://www.3daycar.com/mainframe/publications/library/buildtoorder.pdf>.
- [HM03] Holweg, Matthias and Joe Miemczyk: *Delivering the »3-day car« – the strategic implications for automotive logistics operations*. Journal of Purchasing and Supply Management, (9):63–71, 2003.
- [Hol00] Holweg, Matthias: *The order fulfilment process in the automotive industry – conclusions of the current state analysis*. Technical report, Cardiff University, July 2000. <http://www.3daycar.com/mainframe/publications/library/1stYrSystemReport.pdf>.
- [Hol01] Holweg, Matthias: *Vehicle supply systems in the uk – a historic perspective on sales sourcing and product proliferation*. Technical report, Lean Enterprise Research Centre, Cardiff Business School, März 2001. <http://www.3daycar.com/mainframe/publications/library/salessourcing.pdf>.
- [Hol02] Holweg, Matthias: *The 3daycar component supplier study – investigating the implications of responsive vehicle supply on the component supply chain*. Technical report, Lean Enterprise Research Centre, Cardiff Business School, Februar 2002. <http://www.3daycar.com/mainframe/publications/library/Componentsupply.pdf>.
- [Hol04] Holweg, Matthias: *The 3daycar programme*. In *ILIPT 3DayCar Meeting, Dresden 21.12.2004*, 2004.
- [Hoo04] Hoogestraat, Karl Dirk: *Prozesse im Fokus – Projekt Kunde–Kunde*. 2004. Vortrag auf der Fachtagung »Heute bestellt, morgen geliefert: Zukünftige Anforderungen an die Order-to-Delivery-Prozesse der Automobilindustrie« am 22.4.04. Veranstalter: Fraunhofer IML.
- [How00] Howard, Mickey: *Paint shop survey – a report on the current state of automotive painting and its impact on customer order fulfilment*. Technical report, University of Bath, Januar 2000. <http://www.3daycar.com/mainframe/publications/library/paint.pdf>.
- [How02] Howard, Mickey: *3daycar production flexibility – solving the barriers to build-to-order with production and process technol-*

- ogy. Technical report, University of Bath, Cardiff University, ICDP, 2002. <http://www.3daycar.com/mainframe/publications/library/prodflex.pdf>.
- [HP04] Holweg, Matthias and Frits K. Pil: *The Second Century: Reconnecting Customer and Value Chain Through Build-To-Order; Moving Beyond Mass and Lean Production in the Auto Industry*. MIT Press, 2004.
- [HPV05] Howard, Mickey, Philip Powell, and Richard Vidgen: *Automotive industry information systems: From mass production to build-to-order*. Journal of Cases on Information Technology, 7(1):16–30, 2005.
- [HR98] Hlupic, Vlatka and Stewart Robinson: *Business process modelling and analysis using discrete-event simulation*. In Medeiros, D.J., E.F. Watson, J.S. Carson, and M.S. Manivannan (editors): *Proceedings of the 1998 Winter Simulation Conference*, pages 1363–1369, 1998. www.informs-sim.org/wsc98papers/186.PDF.
- [HT00] Harrington, H. James and Kerim Tumay: *Simulation Modeling Methods: To Reduce Risks and Increase Performance*. McGraw-Hill Companies, 2000.
- [HU90] Hopcroft, John E. und Jeffrey D. Ullman: *Einführung in die Automaten-theorie, Formale Sprachen und Komplexitätstheorie*. Addison-Wesley, 1990.
- [HVP03] Howard, Mickey, Richard Vidgen, and Philip Powell: *Overcoming stakeholder barriers in the automotive industry: building to order with extra-organizational systems*. Journal of Information Technology, 18(1):27–43, 2003.
- [HVPG01] Howard, Mickey, Richard Vidgen, Philip Powell, and Andrew Graves: *Planning for i.s. related industry transformation: The case of the 3day-car*. In *Proceedings of the 9th European Conference on Information Systems*, Bled, JUNI 2001.
- [HvR00] Hommes, Bart Jan and Victor van Reijswoud: *Assessing the quality of business process modelling techniques*. 2000. <http://csdl.computer.org/comp/proceedings/hicss/2000/0493/01/0493toc.htm>, Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 1.
- [HWW00] Hellingrath, Bernd, Jürgen Wloka und Axel Wagenitz: *Simulation des Order-to-Delivery Prozesses in der Automobilindustrie*. VDI Berichte, (1571):93–102, 2000.
- [IEE98] IEEE: *IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97*. The Institute of Electrical and Electronics Engineers (IEEE), 1998.

- [iHSV03] Hove, Anne Schulze im, Frank Stüllenberg und Marcus Völker: *Erweiterung des ProC/B-Paradigmas zur Abbildung entscheidungsrelevanter Kosten und zur Kostenverrechnung*. Technischer Bericht 03019, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.
- [JEJ95] Jacobson, Ivar, Maria Ericsson, and Agneta Jacobson: *The Object Advantage*. Addison-Wesley, 1995.
- [Jen96] Jensen, Kurt: *Coloured Petri Nets*. Springer Verlage, 1996. Basic Concepts, Analysis Methods and Practical Use, Volume 1.
- [JGG⁺02] Joines, Jeffrey A., Deepak Gupta, Mahmut Ali Gokce, Russell E. King, and Michael G. Kay: *Supply chain multi-objective simulation optimization*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/176.pdf>.
- [JGR99] Jacobson, I., G. Booch G., and J. Rumbaugh: *The Unified Software Development Process*. Addison-Wesley, 1999.
- [JJE02] Johansson, Björn, Joacim Johnsson, and Ulf Eriksson: *An evaluation of discrete event simulation software for »dynamic rough-cut analysis«*. 2002. The 35th CIRP-International Seminar on Manufacturing Systems, 12-15 Mai 2002, Seoul, Korea.
- [JJV97] Janssen, Wil, Henk Jonkers, and Jack Verhoosel: *What Makes Business Processes Special? An evaluation framework for modelling languages and tools in Business Process Redesign*. 1997. <http://www.ahaha.demon.nl/docs/caise97.pdf>.
- [JLvB⁺03] Jonkers, Henk, Marc Lankhorst, René van Buuren, Stijn Hoppenbrouwers, Marcello Bonsangue, and Leendert van der Torre: *Concepts for modelling enterprise architectures*. 2003. Landelijk Architectuur Congres 2003, Nieuwegein, http://www.lac2003.nl/academic_papers/H.Jonkers.pdf.
- [Jon87] Jones, Simon L. Peyton: *The Implementation of Functional Programming Languages*. Prentice-Hall, 1987.
- [JR98] Joines, Jeffrey A. and Stephen D. Roberts: *Business process modelling and analysis using discrete-event simulation*. In Medeiros, D.J., E.F. Watson, J.S. Carson, and M.S. Manivannan (editors): *Proceedings of the 1998 Winter Simulation Conference*, pages 141–149, 1998. www.informs-sim.org/wsc98papers/018.PDF.
- [JR99] Joines, Jeffrey A. and Stephen D. Roberts: *Simulation in an object-oriented world*. In *Proceedings of the 1999 Winter Simulation Conference*, pages 132–140, Baltimore, 1999. INFORMS Simulation Society.

- [Kac02] Kaczmarek, Michael: *Definition von Anforderungen an die Modellierung und Analyse der Supply Chain*. Technischer Bericht 02007, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, Oktober 2002.
- [Kel06] Keller, Matthias: *8.4.1b kpi-based evaluation system*. Technical report, ILIPT Projekt, 2006. »Public Deliverable« des ILIPT Projektes.
- [KG95] Kosturiak, Jan und Milan Gregor: *Simulation von Produktionssystemen*. Springer Verlag, 1995.
- [KH02] Kuhn, Axel und Bernd Hellingrath: *Supply Chain Management – Optimierte Zusammenarbeit in der Wertschöpfungskette*. Springer Verlag, 2002.
- [KH04] Kristjansson, Arnar H. and Hans Petter Hildre: *A framework for evaluating platforms in product developing organizations*. In *Workshop Product Structuring '04*, Göteborg, März 2004. Chalmers University of Technology. http://web.mofit.chalmers.se/m/wingquist/workshop03_04/.
- [KK01] Kaiser, Andreas and Wolfgang Küchlin: *Automotive product documentation*. In Monostori, Laslo, Jozsef Vancza, and Moonis Ali (editors): *Proc. 14th Intl. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2001)*, LNAI, Berlin, Heidelberg, New York, JUNI 2001. Springer-Verlag.
- [KLS02] Käppner, Martin, Frank Laakmann und Niklas Stracke: *Dortmunder Prozesskettenparadigma – Grundlagen*. Technischer Bericht 02005, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, Juli 2002.
- [KLT⁺02] Kumara, Soundar R.T., Yong Han Lee, Kaizhi Tang, Chad Dodd, Jeffrey Tew, and Shang Tae Yee: *Simulation anywhere any time: Web-based simulation implementation for evaluating order-to-delivery systems and processes*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, pages 1251–1259, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/168.pdf>.
- [KMN⁺01] Kopperger, Dietmar, Inka Mörschel, Rainer Nägele, Marc Opitz, Peter Schreiner, Christian Schwengels, Tek Seng The und Ilga Vossen: *Business Process Management Tools – Eine evaluierende Marktstudie über aktuelle Werkzeuge*. Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, 2001.
- [Kno02] Knolmeyer, Gerhard F.: *On the economics of mass customization*. In *Moving into Mass Customization – Information Systems and Management Principles*, pages 3–17, 2002.

- [Knu81] Knuth, Donald Ervin: *The Art of Computer Programming*. Addison-Wesley, 1981.
- [Kof03a] Kofman, Ernesto: *Discrete Event Based Simulation and Control of Continuous Systems*. PhD thesis, Universidad Nacional de Rosario, 2003.
- [Kof03b] Kofman, Ernesto: *Discrete Event Based Simulation and Control of Continuous Systems*. PhD thesis, Universidad Nacional de Rosario, 2003.
- [Koh76] Kohlas, J.: *Simulationsmethoden*. In: *Computergestützte Planungssysteme*, Seiten 223–246. Physica, 1976.
- [KS00] Kilger, Christoph and Lorenz Schneeweiss: *Demand fulfilment and atp*. In Stadler, Hartmut and Christoph Kilger (editors): *Supply Chain Management and Advanced Planning*, pages 179–195. Springer Verlag, 2000.
- [KSW03] Klingebiel, Katja, Stefan Spittank und Axel Wagenitz: *Simulation schafft Klarheit*. *Logistik Heute*, 7-8:46–47, 2003.
- [KTB⁺04] Kim, Chang Seop, James Tannock, Mike Byrne, Richard Farr, Bing Cao, and Mahendrawathi Er: *State-of-the-art review techniques to model the supply chain in an extended enterprise*. Technical report, VIVACE Projekt, 2004. »Public Deliverable« des VIVACE Projektes.
- [Kug03] Kugeler, Martin: *Supply Chain Management und Customer Relationship Management – Prozessmodellierung für Extended Enterprises*. In: *Prozessmanagement – Ein Leitfaden zur prozessorientierten Organisationsgestaltung*, Seiten 469–505, Berlin, Heidelberg, New York, Hongkong, 2003. Springer Verlag. Vierte, korrigierte und erweiterte Auflage.
- [Kuh95] Kuhn, Axel: *Prozessketten in der Logistik*. Verlag Praxiswissen, 1995. In Zusammenarbeit mit Chr. Baron, St. Bernemann, J. Kaeseler, Chr. Manthey, S. Wenzel, G. Winz.
- [Law03] Law, Averill M.: *How to conduct a successful simulation study*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 66–70, Baltimore, 2003. INFORMS Simulation Society.
- [Lem00] Lemoine, Paul: *Neue Fertigungsstrategien zur Unterstützung von Order-To-Delivery am Beispiel von Ford Saarlouis*. VDI Berichte, (1571):37–47, 2000.
- [Leo00] Leong, Ang Cheng: *Idef* - a comprehensive modelling methodology for the development of manufacturing enterprise systems*. Technical Report MIT/00/022/OSCA, Singapore Institute of Manufacturing Technology – Operations & Supply Chain Applications Group, Manufacturing Information Technology Division, 2000.
- [LG94] Ljung, Lennart and Torkel Glad: *Modeling of dynamic systems*. Prentice Hall, 1994.

- [LJCG03] Lendermann, Peter, Nirupam Julka, Lai Peng Chan, and Boon Ping Gan: *Integration of discrete event simulation models with framework-based business applications*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1797–1804, Baltimore, 2003. INFORMS Simulation Society.
- [LK91] Law, A. M. and W. D. Kelton: *Simulation Modeling Analysis*. McGraw-Hill, 1991.
- [LMP99] Lozinski, Victor, Peter Mühleck und Adam Polly: *Neue Wege in der Abbildung variantenreicher Produktstrukturen*. Digital World, (März):42–44, 1999.
- [LNH03] Laakmann, Frank, Kasra Nayabi und Ralf Hieber: *Marktstudie 2003 – Supply Chain Management Software*. Fraunhofer IPA, Fraunhofer IML, ETH Zürich BWI, 2003.
- [Mat03] Matloff, Normann: *Simulation methodology*. Internet-Dokument, 2003. <http://heather.cs.ucdavis.edu/matloff/256/PerfModeling/Sim.html>.
- [May03] May, Margaret: *Business Process Management*. FT Prentice Hall, 2003.
- [ME05] Matthäus Esterházy, FlexNet Team: *D4.03-d5.03 vision for future collaborative planning and execution processes (white paper)*. Technical report, ILIPT Projekt, 2005. »Public Deliverable« des ILIPT Projektes.
- [Mey97] Meyer, Bertrand: *Object-Oriented Software Construction*. Prentice-Hall, 1997. 2. Auflage.
- [Mey04a] Meyer, Herbert: *Supply chain planning in the German automotive industry*. OR Spectrum, 26(4):447–470, 2004.
- [Mey04b] Meyr, Herbert: *Kurz- und mittelfristige Planung in der Automobilindustrie zwischen Heute und Morgen*. Technischer Bericht, Lehrstuhl für Produktion und Logistik Universität Augsburg, 2004. http://www.wu-wien.ac.at/home/presse/FNL/FNL4/index/planung_auto.
- [Mey06] Meyer, Christian: *Produktionsprogramme simulationsunterstützt beplanen – Ein Praxisbeispiel von Volkswagen Nutzfahrzeuge*. 2006. 24. Dortmunder Gespräche, 13.-14.9.2005.
- [Min67] Minsky, Marvin Lee: *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.
- [MMC04] Mercer Management Consulting, Fraunhofer Gesellschaft: *Future Automotive Industry Structure (FAST) 2015 – die neue Arbeitsteilung in der Automobilindustrie*. VDA – Verband der Automobilindustrie, 2004.
- [MNZ02] Mane, Arvid, Saeid Nahavandi, and Jingxin Zhang: *Sequencing production on an assembly line using goal chasing and user defined algorithm*.

- In *Proceedings of the 2002 Winter Simulation Conference*, pages 1269–1273, Dezember 2002.
- [MRW04] Motta, Marco, Iwo Riha und Stefan Weidt: *Simulation eines Regional-lagerkonzeptes*. Technischer Bericht 03032, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2004.
- [MW03] Mayo, Donna D. and Knud Erik Wichmann: *Tutorial on business and market modeling to aid strategic decision making: System dynamics in perspective and selecting appropriate analysis approaches*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1569–1577, Baltimore, 2003. INFORMS Simulation Society.
- [Neu95] Neufeldt, Victoria (editor): *Webster's New World College Dictionary*. Macmillan, 1995.
- [NHFF06] Nyhuis, Peter, Nils Hagen, Martin Felder, and Christian Frühwald: *Prozessmanagement im Überblick*. In Hagen, Nils, Peter Nyhuis, Christian Frühwald, and Martin Felder (editors): *Prozessmanagement in der Wertschöpfungskette*, pages 7–22, Bern, Stuttgart, Wien, 2006. Haupt Verlag.
- [NM93] Neumann, Klaus und Martin Morlock: *Operations Research*. Hanser, 1993.
- [NP98] Nikoukaran, Jalal and Vlatka Hlupic Ray J. Paul: *Criteria for simulation software evaluation*. In Medeiros, D.J., E.F. Watson, J.S. Carson, and M.S. Manivannan (editors): *Proceedings of the 1998 Winter Simulation Conference*, pages 399–406, 1998. <http://www.inf.uach.cl/escheihing/asignaturas/info273/lecturas/p399-nikoukaran.pdf>.
- [NZ98a] Nüttgens, Markus and Volker Zimmermann: *Business process modeling with epc and uml: Transformation or integration?* In Schader, M. and A. Korthaus (editors): *The Unified Modeling Language - Technical Aspects and Applications, Proceedings 1. GROOM-Workshop (Mannheim, Oktober 1997)*, pages 250–261, Wiesbaden, 1998. Physica-Verlag. <http://www.iwi.uni-sb.de/nuettgens/Veroeff/Artikel/umlep/umlep.ps>.
- [NZ98b] Nuettgens, Markus und Volker Zimmermann: *Geschäftsprozessmodellierung mit der objektorientierten Ereignisgesteuerten Prozeßketten (oEPK)*. In: Maicher, M. und H. J. Scheruhn (Herausgeber): *Informationsmodellierung – Branchen, Software- und Vorgehensreferenzmodelle und Werkzeuge*, Seiten 23–26, Wiesbaden, 1998. <http://www.iwi.uni-sb.de/nuettgens/Veroeff/Artikel/Kpmg/Kpmg.ps>.
- [Oak97] Oakshott, Les: *Business Modelling and Simulation*. FT Prentice Hall, 1997.

- [Ohl00] Ohl, Stefan: *Prognose und Planung variantenreicher Produkte am Beispiel der Automobilindustrie*. VDI Verlag, 2000.
- [OLCB01] Oliver, Keith, Tim Laseter, Anne Chung, and Dain Black: *Logistics endgame – industry study findings*. Technical report, Booz Allen Hamilton, 2001.
- [OMG03a] OMG: *OMG Unified Modeling Language Specification*. Object Management Group, März 2003. Version 1.5.
- [OMG03b] OMG: *UML 2.0 OCL Specification*. Object Management Group, März 2003.
- [OMG04] OMG: *BUSINESS PROCESSES AND THE OMG*. Object Management Group (OMG), 2004. www.omg.org.
- [OMG05] OMG: *Unified Modeling Language: Superstructure*. Object Management Group, August 2005. Version 2.0.
- [OR03] Owen, Martin and Jog Raj: *BPMN and Business Process Management*. Popkin Software, September 2003. www.bpmn.org.
- [Ott02] Ottomeier, Martin: *BMW peilt erfolgreich über den Daumen*. Computerwoche, Seiten 1–2, 2002. Quelle: Computerwoche Archiv (www.computerwoche.de), Artikel vom 16.10.2002.
- [Pag91] Page, Bernd: *Diskrete Simulation - Eine Einführung mit Modula-2*. Springer Verlag, 1991.
- [Pan03] Panetto, Hervé: *Uml semantics representation of enterprise modeling constructs*. In *Enterprise inter- and intra-organizational integration – Building International Consensus*. Kluwer Academic Publishers, 2003. IFIP TC5/WG5 12. International Conference on Enterprise Integration and Modeling Technology (ICEIMT'02).
- [Pat05] Patwardhan, Bipin: *Introduction to the streaming simd extensions in the pentium iii: Part i*. Dr. Dobbs Journal – Microprocessor Resources, 2005. http://www.x86.org/articles/sse_pt1/simd1.htm.
- [PC99] Pinedo, Michael and Xiuli Chao: *Operations Scheduling with Applications in Manufacturing and Services*. Irwin/McGraw-Hill, 1999.
- [PCDM⁺04] Pouchard, Line, Anne Françoise Cutting-Decelle, Jean Jacques Michel, Robert Young, and Bishnu Das: *Utilizing standard-based approaches for information sharing and interoperability in manufacturing decision support*. In *Proceedings of the Flexible Automation and Intelligent Manufacturing Conference, 14th International Conference*, pages 366–379, Juli 2004.
- [Pen03] Pender, Tom: *UML Bible*. John Wiley & Sons, 2003.

- [Pet02] Petit, Michaël: *Report on the state of the art in enterprise modelling*. Technical report, University of Namur, 2002. Deliverable D1.1 in EU-Projekt IST - 2001 - 34229, <http://www.ueml.org>.
- [Pfo04] Pfohl, Hans Christian: *Logistiksysteme – Betriebswirtschaftliche Grundlagen*. Springer Verlag, 2004. 7. Auflage.
- [PH04] Pil, Frits K. and Matthias Holweg: *Linking product variety to order-fulfillment strategies*. *Interfaces*, 34(5):394–403, 2004.
- [PS02] Piller, Frank T. and Christof M. Stotko: *Mass customization: four approaches to deliver customized products and services with mass production efficiency*. In *Proceedings of the 2002 IEEE International Engineering Management Conference (Vol. II)*, pages 773–778. IEEE, August 2002.
- [Qua98] Quatrani, Terry: *Visual Modeling with Rational Rose and UML*. Addison-Wesley, 1998.
- [Qua02] Quack, Karin: *Alle Zulieferer haben den Durchblick*. Computerwoche, Seiten 1–5, 2002. Quelle: Computerwoche Archiv (www.computerwoche.de), Artikel vom 8.3.2002.
- [Rai98] Rai, Gurjit: *Process simulation and modeling*. Technical report, University of Calgary, 1998. Web Report <http://sern.ucalgary.ca/~grai/seng/609/simulationfinalreport3.html>.
- [Rau99] Raufeisen, Michael: *Konzept zur Komplexitätsmessung des Auftragsabwicklungsprozesses – Eine empirische Untersuchung*. TCW Transfer-Centrum GmbH, München, 1999.
- [RC03] Rossetti, Manuel D. and Hin Tat Chan: *A prototype object-oriented supply chain simulation framework*. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1612–1620, Baltimore, 2003. INFORMS Simulation Society.
- [Ren00] Renner, Peter: *Kundenorientierung in der Automobilindustrie*. VDI Berichte, (1571):23–33, 2000.
- [Ren04] Renner, Peter: *Kundenorientierter Vertriebs- und Produktionsprozess (KOV) – KOV in der Praxis*. 2004. Vortrag auf der Fachtagung »Heute bestellt, morgen geliefert: Zukünftige Anforderungen an die Order-to-Delivery-Prozesse der Automobilindustrie« am 22.4.04. Veranstalter: Fraunhofer IML.
- [RG01] Ramakrishnan, Raghu and Johannes Gehrke: *Database Management Systems*. McGraw Hill, 2001. Second Edition.
- [Rit04] Rittscher, Jens: *Strategieflexible Entscheidungsunterstützung in der Ablaufplanung für Produzenten*. Verlag Praxiswissen, 2004.

- [RJB99] Rumbaugh, J., I. Jacobson, and G. Booch: *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.
- [Ros05] Rose, Bernhard: *Teure Kostentreiber*. *Automobil Industrie*, 50(1-2):16–18, 2005.
- [RS03] Rittscher, Jens und Stefan Spittank: *Anlauf-Prozesse optimieren*. *Produktion*, (41), 2003.
- [RWW03] Rabeler, Carl, Len Wyatt, and Dave Wickert: *Microsoft sql server 2000 analysis services performance guide*. Internet-Dokument, 2003. <http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ansvcspg.msp>.
- [Sar98] Sargent, Robert G.: *Verification and validation of simulation models*. In Medeiros, D.J., E.F. Watson, J.S. Carson, and M.S. Manivannan (editors): *Proceedings of the 1998 Winter Simulation Conference*, pages 141–149, 1998. www.informs-sim.org/wsc98papers/018.PDF.
- [SB98] Schütte, Reinhard und Jörg Becker: *Subjektivitätsmanagement bei Informationsmodellen*. 1998. *Modellierung '98 – Proceedings des GI-Workshops in Münster*, 11.-13. März 1998.
- [SB02] Schriber, Thomas and Daniel Brunner: *Inside discrete-event simulation software: How it works and why it matters*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format im Verzeichnis: <http://www.informs-sim.org/wsc02papers/>.
- [SB05] Schenk, Michael und Dietmar Bufka: *Kundenwunsch als Maßstab*. *Logistik Heute*, 27(6):28–29, 2005.
- [Sch90] Scheer, A. W.: *Wirtschaftsinformatik*. Springer Verlag, 1990. 3. Auflage.
- [Sch95] Schütte, Reinhard: *Basispositionen in der Wirtschaftsinformatik – ein gemäßigt-konstruktivistisches Programm*. In: Becker, J., W. König, R. Schütte, O. Wendt und S. Zelewski (Herausgeber): *Wirtschaftsinformatik und Wissenschaftstheorie – Bestandsaufnahme und Perspektiven*, Seiten 211–241, Wiesbaden, 1995.
- [Sch97] Schmidt, Ulrich: *Angewandte Simulationstechnik für Produktion und Logistik*. Verlag Praxiswissen, 1997.
- [Sch98a] Schönsleben, Paul: *Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsprozessen*. Springer Verlag, 1998.
- [Sch98b] Schütte, R.: *Grundsätze ordnungsmäßiger Referenzmodellierung – Konstruktion configurations- und anpassungsorientierter Modelle*. Dr. Th. Gabler Verlag, 1998, ISBN 3-409-12843-3.

- [Sch98c] Schütte, Reinhard: *Vergleich alternativer Ansätze zur Bewertung der Informationsmodellqualität*. 1998. Fachtagung Modellierung betrieblicher Informationssysteme 15. und 16. Oktober 1998 Universität Koblenz.
- [Sch98d] Schwermer, M.: *Modellierungsvorgehen zur Planung von Geschäftsprozessen*. Fraunhofer IRB Verlag, 1998.
- [Sch99a] Schütte, Reinhard: *Die neuen Grundsätze ordnungsmäßiger Modellierung*. 1999. Paper zum Forschungsforum '97, Leipzig 16.-20. September 1997.
- [Sch99b] Schütte, Reinhard: *Zum Realitätsbezug von Informationsmodellen*. EMISA Forum, 9(2):26–36, 1999.
- [Sch99c] Schöttner, Josef: *Produktdatenmanagement in der Fertigungsindustrie*. Carl Hanser Verlag, 1999.
- [Sch01] Scheer, August Wilhelm: *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Springer Verlag, 2001. 4. Auflage.
- [Sch05] Schmitz, Michael: *Modellierung und Simulation von Gebietsspediteurnetzen in der Automobilindustrie*. Verlag Praxiswissen, 2005.
- [SGR97] Stickel, Eberhard, Hans Dieter Groffmann und Karl Heinz Rau (Herausgeber): *Gabler-Wirtschaftsinformatik-Lexikon*. Gabler Verlag, 1997.
- [SGT⁺00] Schlenoff, Craig, Michael Gruninger, Florence Tissot, John Valois, Josh Lubell, and Jintae Lee: *The process specification language (psl) overview and version 1.0 specification*. Technical Report NISTIR 6459, National Institute of Standards and Technology, Juli 2000.
- [SH02] Stoßberg, Tilmann und Bernd Hellingrath: *OTD-Sim: Order-to-Delivery-Prozesse effizienter gestalten*. 2002. BVL-Kongress, 16. Oktober 2002.
- [Sha75] Shannon, Robert E.: *Systems Simulation*. Prentice Hall, 1975.
- [SK01] Sinz, Carsten and Wolfgang Küchlin: *Dealing with temporal change in product documentation for manufacturing*. In *IJCAI-01 Workshop on Configuration in conjunction with the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, August 2001. <http://www.soberit.hut.fi/pdmg/IJCAI2001ConfWS/>.
- [SK05] Sivaraman, Eswar and Manjunath Kamath: *Verification of business process designs using maps*. In Golden, Bruce L., S. Raghavan, and Edward A. Wasil (editors): *THE NEXT WAVE IN COMPUTING, OPTIMIZATION, AND DECISION TECHNOLOGIES*. Springer Verlag, 2005.
- [SN00] Scheer, August Wilhelm and Markus Nüttgens: *Aris architecture and reference models for business process management*. In *Business Process*

- Management - Models, Techniques, and Empirical Studies*, pages 366–379, 2000.
- [SS97] Schwegmann, Ansgar und Bernhard Schlagheck: *Integration der Prozeßorientierung in das objektorientierte Paradigma: Klassenzuordnungsansatz vs. Prozeßklassenansatz*. In: *Arbeitsberichte des Instituts für Wirtschaftsinformatik – Arbeitsbericht 60*, Münster, 1997. Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster.
- [SS99] Sheridan, Patrick and Jean Sekula: *Iterative UML Development Using Visual C++ 6.0*. Wordware Publishing, 1999.
- [SS04] Schrapel, Rainer und Thomas Sauter: *Horizontale und vertikale Integration am Beispiel des Auftragsmanagements bei der MCG der Daimler-Chrysler AG*. In: *Informatik 2004, 34. Jahrestagung der Gesellschaft für Informatik, Workshop „Architekturen verbinden! – Horizontale und vertikale Integration von IT-Systemen und Services in der Praxis“*. Universität Ulm, September 2004. http://www.sdm.de/download/pdf/v04_sauterschrapel.pdf.
- [SSE02] Schenk, Michael and Ralph Seelmann-Eggebert: *Mass customization facing logistics challenges*. In *Moving into Mass Customization – Information Systems and Management Principles*, pages 41–57, 2002.
- [Sta73] Stachowiak, Herbert: *Allgemeine Modelltheorie*. Springer, 1973, ISBN 3-211-81106-0.
- [Sta01a] Stautner, Ulrich: *Das neue BMW Vertriebs- und Produktionssystem*. In: *Innovative Logistik in der Automobilindustrie*. VDI Verlag, 2001. VDI-Berichte 1628.
- [Sta01b] Stautner, Ulrich: *Kundenorientierte Lagerfertigung im Automobilvertrieb*. Deutscher Universitäts-Verlag (Gabler Edition Wissenschaft), 2001.
- [Str95] Stroustrup, Bjarne: *Why c++ is not just an object-oriented programming language*. Technical report, 1995. Addendum to OOPSLA95 Proceedings, ACM OOPS Messenger, October 1995, <http://www.research.att.com/~bs/oopsla.pdf>.
- [Str97] Stroustrup, Bjarne: *The C++ Programming Language*. Addison-Wesley, 1997. Third Edition.
- [Tea03] Team, 3DayCar Research: *Towards a customer driven system : A summary of the 3daycar research programme*. Technical report, Lean Enterprise Research Center, ICDP, University of Bologna, University of Bath, 2003.
- [Tep04] Tepper, Carsten: *Prozessablauf-Visualisierung von ProC/B-Modellen*. Technischer Bericht 04003, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2004.

- [Thi03] Thiele, Lothar: *Discrete Event Systems*. Computer Engineering and Networks Laboratory Swiss Federal Institute of Technology (ETH) Zurich, 2003. http://www.tik.ee.ethz.ch/tik/education/lectures/DES/Book/des_book_intro.pdf.
- [Tho05] Thomas, Oliver: *Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation*. Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), Mai 2005. Heft 184.
- [TVVB02] Tewoldeberhan, Tamrat W., Alexander Verbraeck, Edwin Valentin, and Gilles Bardonnet: *An evaluation and selection methodology for discrete-event simulation software*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/010.pdf>.
- [VA02] VOLKSWAGEN AG, K-DOB-4, EDI Koordination: *EDI Implementation Guidelines von Volkswagen/Audi*. Internet-Dokument, 2002. <http://www.vwgroupsupply.com/br/file/70AB8A62600000E35B0C6DF2103CB228.pdf>.
- [vdA92] Aalst, W.M.P. van der: *Timed coloured Petri nets and their application to logistics*. PhD thesis, Eindhoven University of Technology, 1992.
- [vdAD02] Aalst, Wil van der and Jörg Desel: *On the semantics of epcs: A vicious circle*. In *EPK 2002 / Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, pages 71–79. Gesellschaft für Informatik, 2002.
- [vdADK02] Aalst, Wil van der, Jörg Dresel, and Ekkart Kindler: *On the semantics of epcs: A vicious circle*. In *EPK 2002*, pages 71–80. Gesellschaft für Informatik, 2002. Geschäftsprozessmodellierung mit Prozessketten.
- [VDI93] VDI: *VDI 3633, Blatt 1, Simulation von Logistik-, Materialfluß- und Produktionssystemen – Grundlagen*. VDI-Handbuch Materialfluß und Fördertechnik, Band 8, 1993. Stand 11.2004.
- [VDI96] VDI: *VDI 3633, Begriffsdefinitionen*. VDI-Handbuch Materialfluß und Fördertechnik, Band 8, 1996. Stand 11.2004.
- [VDI97a] VDI: *VDI 3633, Blatt 3, Simulation von Logistik-, Materialfluß- und Produktionssystemen – Experimentplanung und -auswertung*. VDI-Handbuch Materialfluß und Fördertechnik, Band 8, VDI-Handbuch Betriebstechnik, Teil 1, 1997. Stand 12.2005.
- [VDI97b] VDI: *VDI 3633, Blatt 4, Auswahl von Simulationswerkzeugen – Leistungsumfang und Unterscheidungskriterien*. VDI-Handbuch Materialfluß und Fördertechnik, Band 8, VDI-Handbuch Betriebstechnik, Teil 1, 1997. Stand 11.2004.

- [VDI00] VDI: *VDI 3633, Blatt 5, Simulation von Logistik-, Materialfluß- und Produktionssystemen – Integration der Simulation in innerbetriebliche Abläufe*. VDI-Handbuch Materialfluß und Fördertechnik, Band 8, VDI-Handbuch Betriebstechnik, Teil 1, 2000. Stand 11.2004.
- [Voß06] Voß, Holger: *ForLog-Studie: Logistik-Outsourcing in der Automobilindustrie – Eine Untersuchung zur Flexibilität*. Technischer Bericht, 2006. ForLog – Bayerischer Forschungsverbund Supra-adaptive Logistiksysteme.
- [VS02] Vastag, Alex und Andreas Schürholz: *Distribution*. In: Arnold, Dieter, Heinz Isermann, Axel Kuhn und Horst Tempelmeier (Herausgeber): *Handbuch Logistik*. Springer Verlag, 2002. Kapitel B 5.
- [vU01] Uthmann, Christoph von: *Geschäftsprozesssimulation von Supply Chains*. Society for Computer Simulation International, 2001.
- [WB02] Waller, Ben and Monica Bartolini: *What is the pay-off from 3daycar? evaluating the case for profit improvement from a completely build to order system with 3 day order to delivery capability*. Technical report, University of Bath, Cardiff University, ICDP, 2002.
- [WBN⁺02] Wenzel, Sigrid, Jochen Bernhard, Stefan Nickel, Dietmar Hietel, Alexander Lavrov, Görges Deerberg, Kerstin Schwarze-Benning, Hans Jürgen Körner, Wolfgang Appelt und Elke Hinrichs: *SILVER – Simulationsbasierte Systeme zur Integration logistischer und verfahrenstechnischer Entscheidungsprozesse*. 2002. <http://silver.itwm.fhg.de/Übersicht/Berichte/berichte.html>.
- [WBN⁺03a] Wenzel, Sigrid, Jochen Bernhard, Stefan Nickel, Dietmar Hietel, Alexander Lavrov, Görges Deerberg, Kerstin Schwarze-Benning, Hans Jürgen Körner, Wolfgang Appelt und Elke Hinrichs: *SILVER – Simulationsbasierte Systeme zur Integration logistischer und verfahrenstechnischer Entscheidungsprozesse – Arbeitsbereich 2 – Gesamtkonzepte*. 2003. <http://silver.itwm.fhg.de/Übersicht/Berichte/berichte.html>.
- [WBN⁺03b] Wenzel, Sigrid, Jochen Bernhard, Stefan Nickel, Dietmar Hietel, Alexander Lavrov, Görges Deerberg, Kerstin Schwarze-Benning, Hans Jürgen Körner, Wolfgang Appelt und Elke Hinrichs: *SILVER – Simulationsbasierte Systeme zur Integration logistischer und verfahrenstechnischer Entscheidungsprozesse – Arbeitsbereich 3 – Gekoppelte Modelle zur hybriden Simulation*. 2003. <http://silver.itwm.fhg.de/Übersicht/Berichte/berichte.html>.
- [WBN⁺04] Wenzel, Sigrid, Jochen Bernhard, Stefan Nickel, Dietmar Hietel, Alexander Lavrov, Görges Deerberg, Kerstin Schwarze-Benning, Hans Jürgen Körner, Wolfgang Appelt und Elke Hinrichs: *SILVER – Simulationsbasierte Systeme zur Integration logistischer und verfahrenstechnischer Entscheidungsprozesse – Arbeitsbereich 4 – Proto-*

- typ. 2004. <http://silver.itwm.fhg.de/Übersicht/Berichte/berichte.html>.
- [Web06] Webster, Sarah A.: *Top dealer says there are more unsold cars than reported*. 2006. Detroit Free Press (online), 26.10.2006.
- [Wel05] Weller, Rüdiger: *Logistische Planung eines globalen Zuliefer- und Produktionsnetzwerkes*. 2005. Vortrag auf der Fachtagung: Zulieferlogistik in der Automobilindustrie – effektiv und zukunftsorientiert, 15.11.2005, Ludwigsburg.
- [Wen04] Wenzel, Sigrid: *Fabrikplanung: Von der Ablaufsimulation zur Digitalen Fabrik*. Dortmund, 2004. Fraunhofer IML. Logistik-Anwendungsforum SIMULATION Dortmund: Integrierte Nutzung der Logistiksimulation als Teil kundenspezifischer Planungsumgebungen.
- [Wen05] Wenz, Konrad: »Elektronik ist kein Selbstzweck«. *Automobil Industrie*, 50(6):86–87, 2005.
- [Wöh93] Wöhe, Günther: *Einführung in die Allgemeine Betriebswirtschaftslehre*. Verlag Franz Vahlen, 1993. 18. Auflage.
- [WH00] Wagenitz, Jürgen Wloka Axel and Bernd Hellingrath: *Alternative order management strategies in the automotive industry – analysis and evaluation of a new order management process by using simulation*. In *Proceedings from the 6th International Conference on Information Systems Analysis and Synthesis*, Orlando, Juli 2000.
- [WH03] Witthaut, Markus und Bernd Hellingrath: *Modellierungsrahmen für Dispositionsstrategien in GNL*. Technischer Bericht 03007, SFB 559 »Modellierung großer Netze in der Logistik«, Universität Dortmund, 2003.
- [Whi04a] White, Stephen A.: *BPMN Fundamentals*. BPMI.org, 2004. www.bpmi.org.
- [Whi04b] White, Stephen A.: *Business Process Modeling Notation (BPMN)*. BPMI.org, Mai 2004. Version 1.0, www.bpmn.org.
- [WHTM04] Wagenitz, Axel, Bernd Hellingrath, Michael Toth und Marco Motta: *Simpliere und herrsche*. *Beschaffung Aktuell*, (10/2004):32–33, 2004.
- [Wie03] Wienker, Reinhard Große: *Innovationsgrenze Komplexität überwinden*. Karlsruhe, 2003. 12. AIK-Symposium Herausforderung Komplexität-Komplexitätsmanagement und IT Karlsruhe, 17.10.2003.
- [Wil96] Wildemann, Horst: *Variantenmanagement*. TCW Transfer Centrum Verlag, 1996.
- [Wil04a] Wildemann, Horst: *Auftragsabwicklungsprozess*. TCW Transfer-Centrum GmbH Co. KG, 2004.

- [Wil04b] Will, Peter: *Kostenpotenziale in der Distribution – Analyse und Optimierung von Distributionsprozessen – Erfahrungen aus der Praxis*. 2004. Vortrag auf der Fachtagung: Heute bestellt, morgen geliefert: Zukünftige Anforderungen an die Order-to-Delivery-Prozesse der Automobilindustrie.
- [WLB+02] Windt, Katja, Stefan Lutz, Jan Wilhelm Breithaupt, Peter Nyhuis und Lothar Schulze: *Lenkung*. In: Arnold, Dieter, Heinz Isermann, Axel Kuhn und Horst Tempelmeier (Herausgeber): *Handbuch Logistik*. Springer Verlag, 2002. Kapitel B 3.3.
- [WN00] Wenzel, Sigrid und Bernd Noche: *Simulationsinstrumente in Produktion und Logistik – eine Marktübersicht*. In: Mertins, Kai und Markus Rabe (Herausgeber): *The New Simulation in Production and Logistics*, Berlin, März 2000. Fraunhofer Institut Produktionsanlagen und Konstruktionstechnik, Eigenverlag.
- [Wo195] Wolff, Stefan: *Zeitoptimierung in logistischen Ketten – Ein Instrumentarium zum Controlling von Liefer- und Durchlaufzeiten bei kundenspezifischer Serienproduktion*. Huss Verlag, 1995. Schriftenreihe der Bundesvereinigung Logistik (BVL) e.V, Bremen.
- [WQ97] Winz, Gerald und Michael Quint: *Prozeßkettenmanagement*. Verlag Praxiswissen, 1997.
- [WS01] Weyer, Matthias und Dieter Spath: *Das Produktionssteuerungskonzept »Perlenkette«*. Zeitschrift für wirtschaftlichen Fabrikbetrieb, 96(1-2):17–19, 2001.
- [WSS01] Werner Scholze-Stubenrecht, Birgit Eickhoff, Dieter Mang: *Duden Fremdwörterbuch*. Bibliographisches Institut, 2001.
- [WSWK06] Wagenitz, Axel, Stefan Spittank, Markus Witthaut, and Katja Klingebiel: *Simulation based evaluation of collaborative execution processes – d5.10*. Technical report, ILIPT Projekt, 2006. »Public Deliverable« des ILIPT Projektes.
- [WvdAV04] Weske, Mathias, Wil M. P. van der Aalst, and H. M. W. (Eric) Verbeek: *Advances in business process management*. Data Knowledge Engineering, 50(1):1–8, 2004.
- [WW00] Wagenitz, Axel und Jürgen Wloka: *Simulation komplexer Geschäftsprozesse über die gesamte Logistikkette von Unternehmen – eine Fallstudie*. In: Mertins, Kai und Markus Rabe (Herausgeber): *The New Simulation in Production and Logistics*, Berlin, März 2000. Fraunhofer Institut Produktionsanlagen und Konstruktionstechnik, Eigenverlag.
- [WWH00] Wagenitz, Axel, Jürgen Wloka, and Bernd Hellingrath: *Simulation of alternative order management strategies in the automotive industry*. In *Proceedings of the 16th International Conference on CAD/CAM*,

- Robotics and Factories of the Future*, Port of Spain, Trinidad and Tobago, Juni 2000.
- [WYS02] Worbs, Jochen, Kwok Wai Yu und Lothar Schulze: *Ressourcen*. In: Arnold, Dieter, Heinz Isermann, Axel Kuhn und Horst Tempelmeier (Herausgeber): *Handbuch Logistik*. Springer Verlag, 2002. Kapitel B 3.4.
- [Yee02] Yee, Shang Tae: *Establishment of product offering and production leveling principles via supply chain simulation under order-to-delivery environment*. In Yücesan, E., C. H. Chen, J. L. Snowdon, and J. M. Charnes (editors): *Proceedings of the Winter Simulation Conference (WSC 2002)*, pages 1260–1268, San Diego, Dezember 2002. Artikel im PDF-Format: <http://www.informs-sim.org/wsc02papers/169.pdf>.
- [Zei76] Zeigler, Bernard P.: *Theory of Modelling and Simulation*. John Wiley Sons, 1976.
- [Zei02] Zeier, Alexander: *Identifikation und Analyse branchenspezifischer Faktoren für den Einsatz von Supply-Chain-Management-Software – Teil I: Grundlagen, Methodik und Kernanforderungen*. Technischer Bericht FWN-2002-002, FORWIN – Bayerischer Forschungsverbund Wirtschaftsinformatik, 2002.
- [ZPK00] Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim: *Theory Of Modeling And Simulation: Integrating Discrete Event And Continuous Complex Dynamic Systems*. Academic Press, San Diego, San Francisco, New York, Boston, 2000. Second Edition.

A. Anhang – Entwicklungssystem

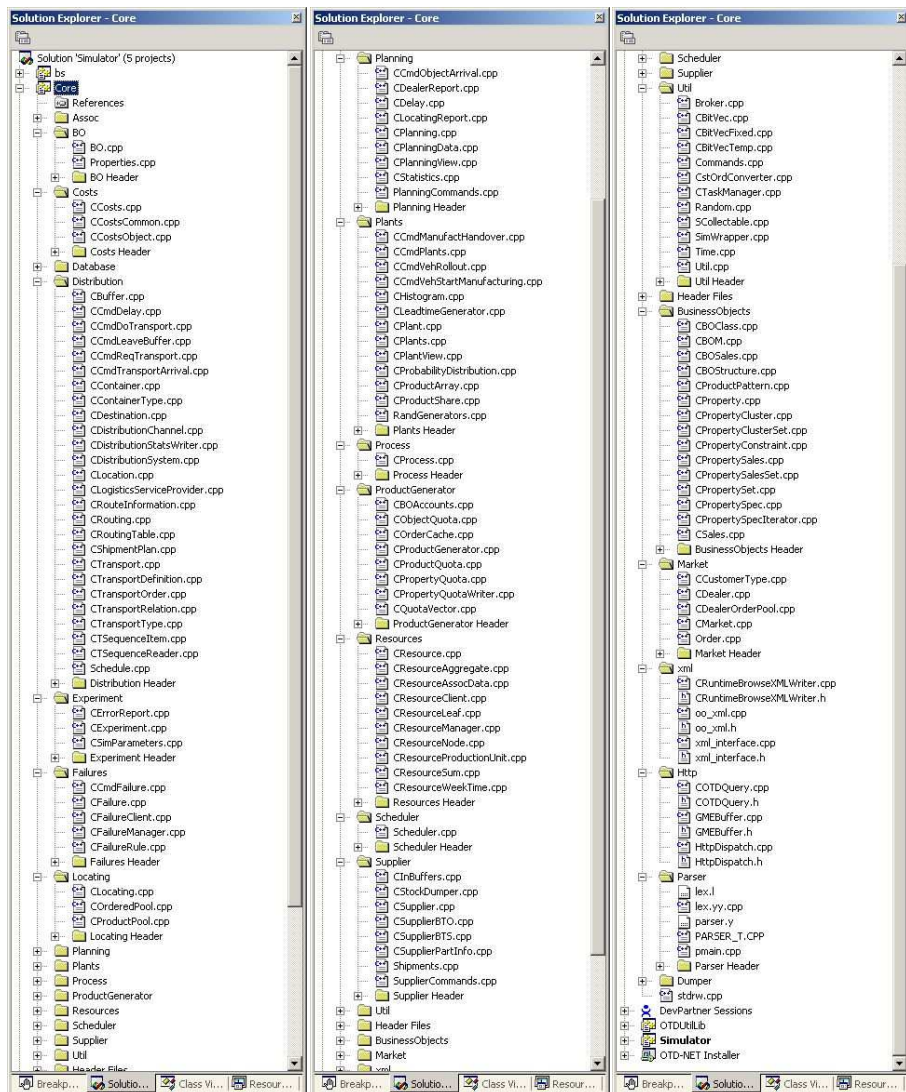


Abbildung A.1.: C++-Klassen im Entwicklungssystem Visual Studio 2005

B. Anhang – Teile des Beispielmmodells

Engine Modules (EM)	Front Modules (FM)	Door Modules (DM)
EM diesel automatic	FM left steering, halogen, elegance	DM sensor v1
EM diesel hybrid automatic	FM left steering, halogen, sports	DM sensor v2
EM diesel hybrid manual	FM left steering, xenon, elegance	DM sensor v3
EM diesel manual	FM left steering, xenon, sports	DM sensor v4
EM fuel cell automatic	FM right steering, halogen, elegance	
EM gasoline automatic	FM right steering, halogen, sports	
EM gasoline hybrid automatic	FM right steering, xenon, elegance	
EM gasoline hybrid manual	FM right steering, xenon, sports	
EM gasoline manual		

Tabelle B.1.: Eigenschaftenfamilien Beispielmmodell

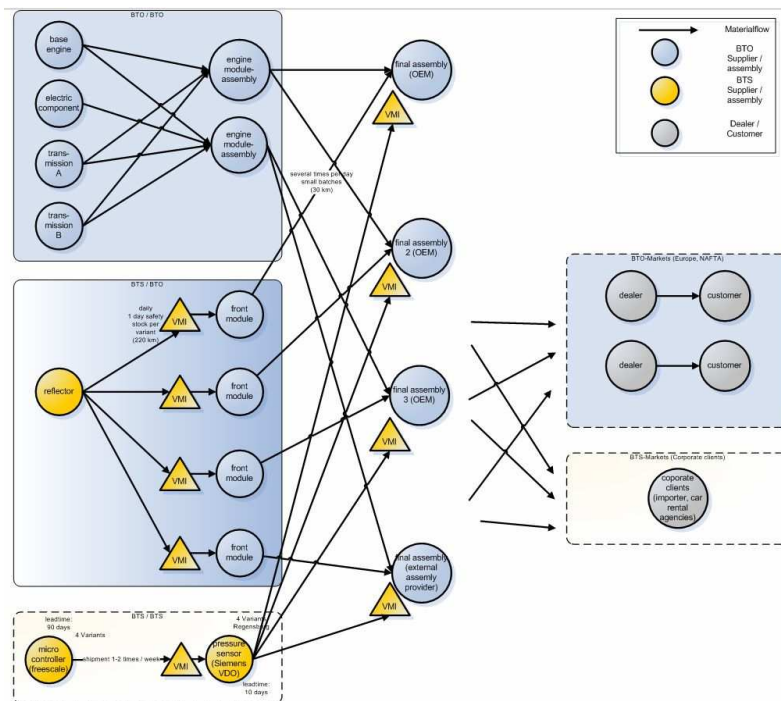


Abbildung B.1.: Modellstruktur



Abbildung B.2.: Montagestandorte

Fahrzeitzonen um die Werksstandorte (siehe Abbildung B.2.):

1. Bremen (blau)
2. Toulouse (rot)
3. Turin (grün)
4. Prag (gelb)

Durch Überlappung der Zonen ist auf dieser Abbildung nur die Ausdehnung der Fahrzeitzone des Werkes Turin in voller Ausdehnung zu sehen.

	Bremen	Toulouse	Turin	Prag
Market Austria	5808		360	808
Market Benelux	7344	5056	8488	1920
Market Corporate Clients	13952	13800	13872	5432

	Bremen	Toulouse	Turin	Prag
Market France	13640	30928	624	
Market Germany	47040	1976	9904	14400
Market Italy	3192	9632	33744	4240
Market NAFTA	16504	33728	37312	6568
Market Switzerland	496	1768	3520	248
Markets Czech Republic, Hungary, Slovakia, Slovenia	616	1264	1480	12896
Markets Denmark, Norway, Sweden, Finland	1472	464	496	280
Markets Estonia, Latvia, Lithuania, Poland	8856	2768	4384	3496
Markets Spain, Portugal	3288	20632	7800	2320
Markets UK, Ireland	27216	17312	16544	

Tabelle B.2.: Verteilung Marktvolumina auf Werke

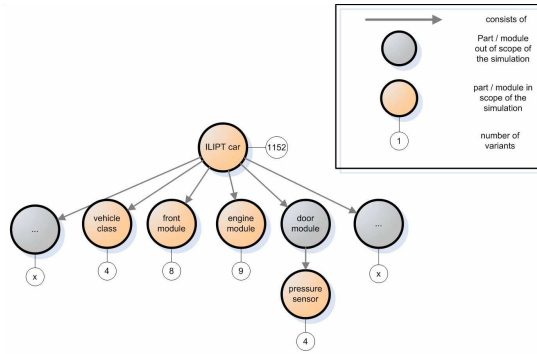


Abbildung B.3.: Modulstruktur Fahrzeug

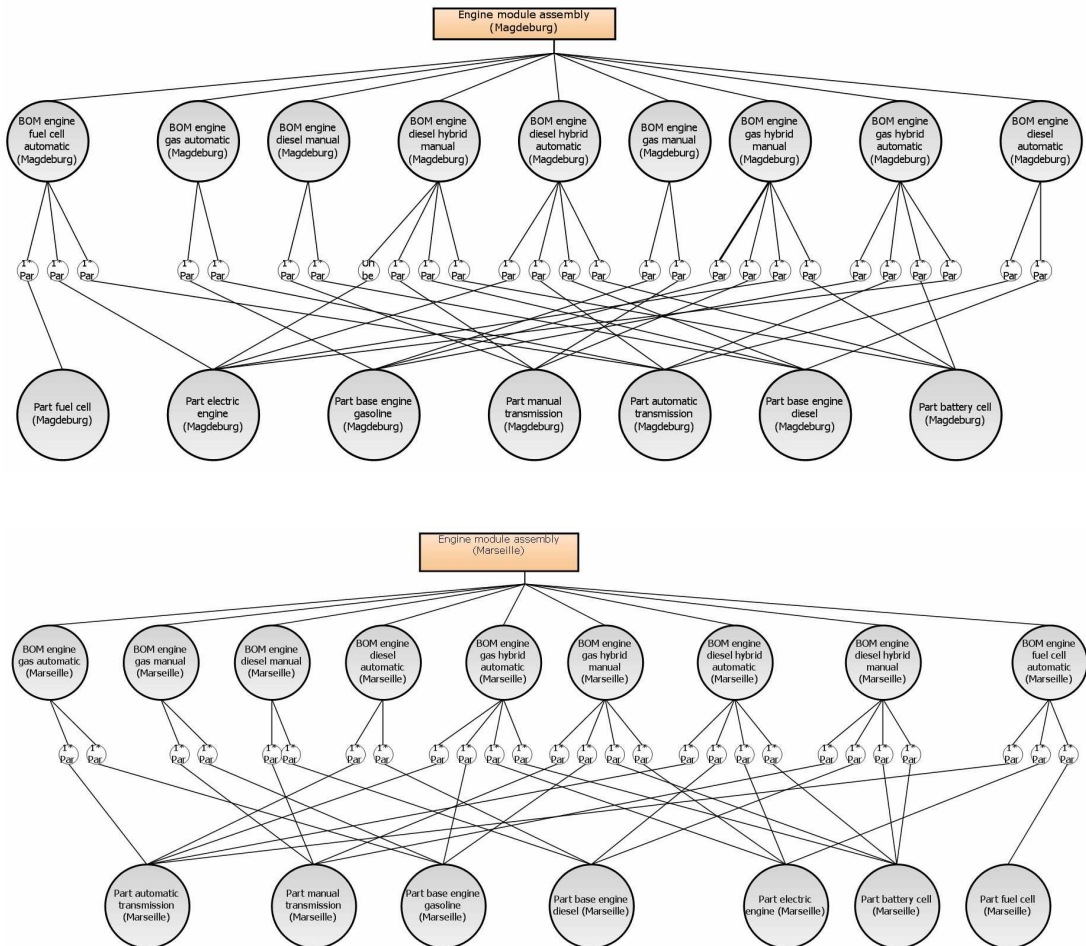


Abbildung B.4.: Stücklisten Motoren

Abbildung B.5.: Liefernetzstruktur Motoren

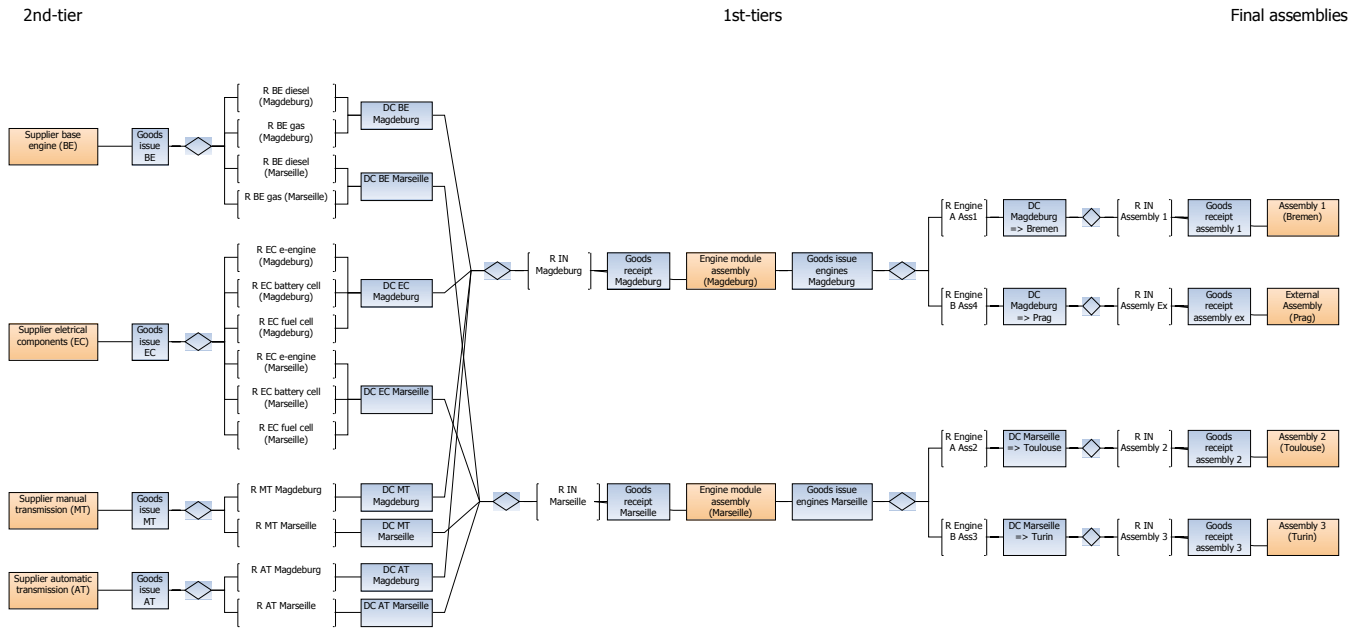
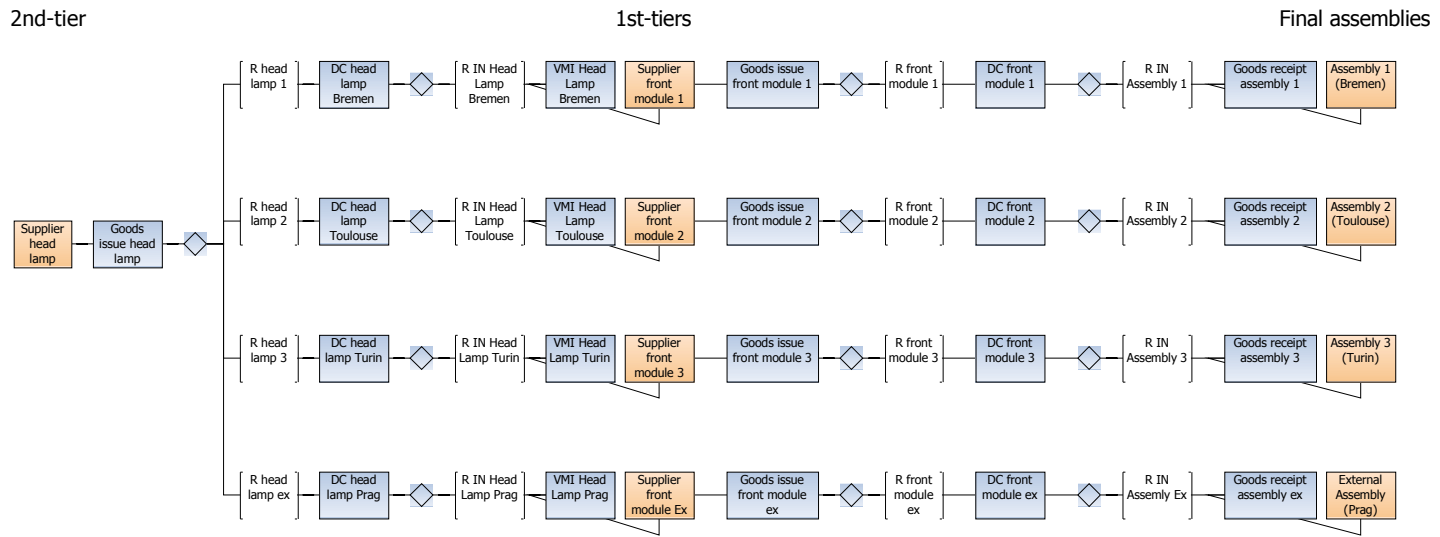


Abbildung B.6.: Liefernetzstruktur Front-Modul



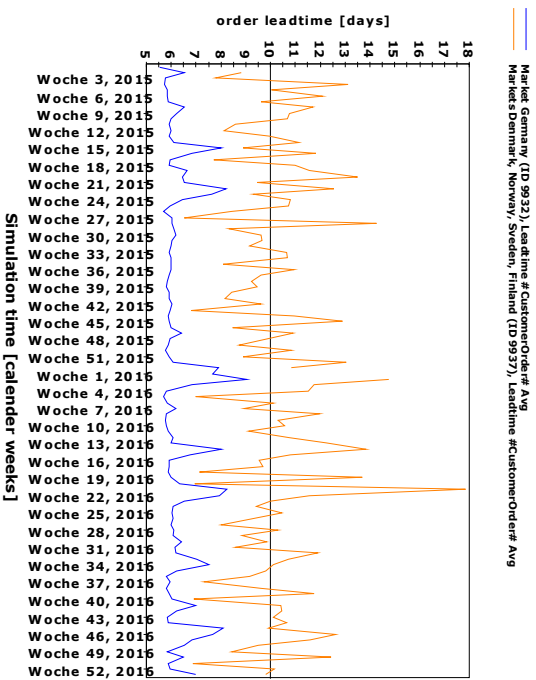


Abbildung B.7.: Durchschnittliche Lieferzeit Skandinavien und Deutschland

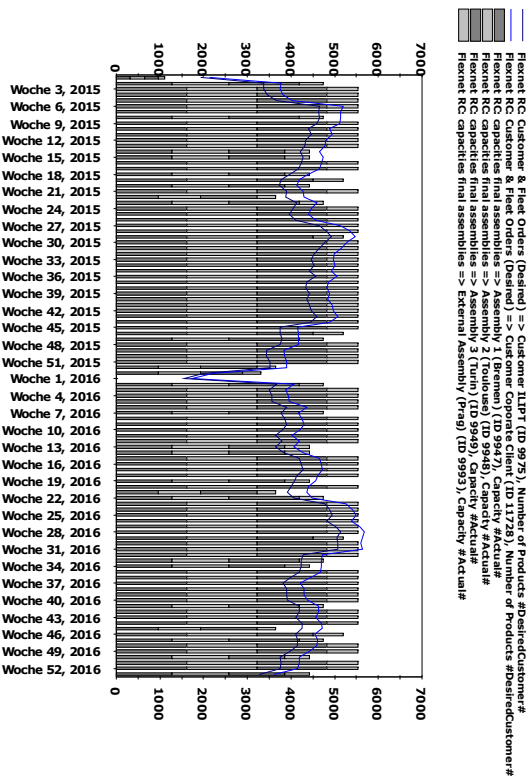


Abbildung B.8.: Kapazitäten der Werke und Kundennachfrage

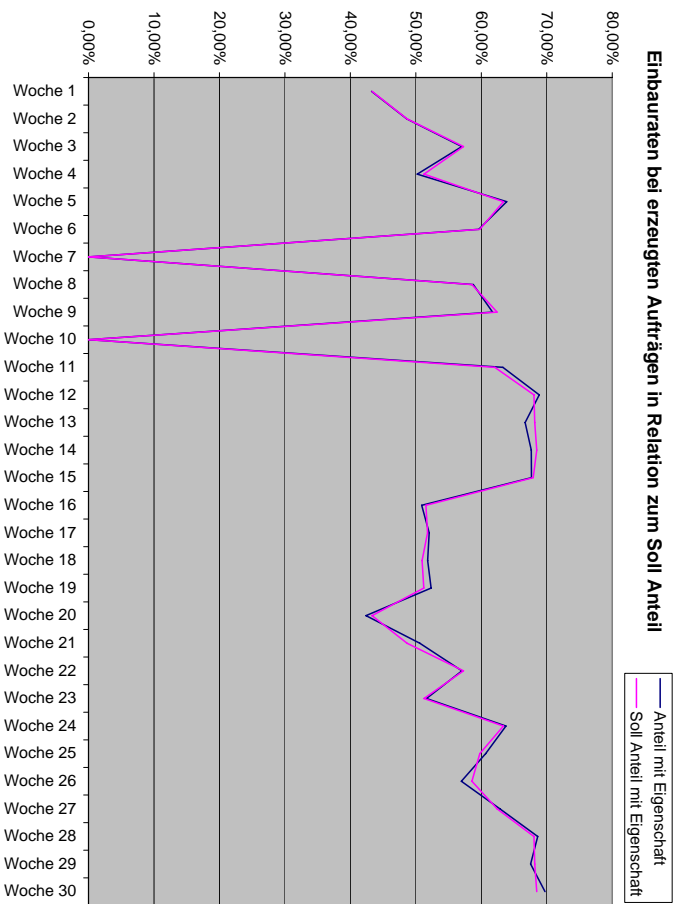


Abbildung B.9.: Abweichung Einbauraten Soll/Ist

C. Anhang – Beispiel Produktbeschreibung

Eigenschaftensfamilie	Code	Eigenschaft	Einbaurrate
Ausstattung	101	Basis	48 %
	102	Comfort	29 %
	103	Familiy	13 %
	104	Business	17 %
	105	Sport	3 %
Motor	201	75KW-Motor	50 %
	202	95KW-Motor	50 %
	203	110KW-Motor	50 %
	204	150KW-Motor	50 %
Getriebe	301	Schaltautomat 6-Gang	50 %
	302	6-Gang-Getriebe sequentiell	50 %
	303	6-Gang-Getriebe 350 NM	50 %
	304	6-Gang-Getriebe 250 NM	50 %
	305	5-Gang Getriebe	50 %
	306	5-Gang Getriebe Allrad	50 %
	307	6-Gang Getriebe Allrad	50 %
Navigation/Radio	401	Radio CD	50 %
	402	Soundpaket Plus	50 %
	403	Navigation Mono	50 %
	404	Navigation Color	50 %
	405	Navigation Color TV	50 %
Batterie	501	44Ah	50 %
	502	60Ah	50 %
	503	70Ah	50 %
	504	74Ah	50 %
	505	80Ah	50 %
	506	85Ah	50 %
	507	105Ah	50 %
Sitze	601	Stoff	50 %
	602	Leder rot	50 %
	603	Leder gelb	50 %
	604	Leder schwarz	50 %
Schaltknauf	701	SK Lederimitat	50 %
	702	SK Leder rot	50 %
	703	SK Leder gelb	50 %
	704	SK Leder schwarz	50 %
	705	SK Sport	50 %

Tabelle C.1.: Erweitertes Beispiel für Einbauraten, Zwänge und Verbote

Nr	Konstruktionsregeln
2	75KW-Motor <i>zwingt</i> 5-Gang-Getriebe

Tabelle C.2.: Aus Einbauraten abgeleitete Konstruktionsregel

D. Anhang – Ausgewählte UML Diagramme

UML-Klasse	Abb.	GME-Klasse	Abb.
CSimObject	D.1	SimObject	E.68
CExperiment	D.2	Experiment	E.23
CSimParameters	D.3	SimParameters	E.69
CBOStructure	D.5	Experiment	E.53
CBOClass	D.5	ProductClass	E.48
CPropertyCluster	D.6	PropertyCluster	E.59
CProperty	D.7	Property	E.54
CPropertySpec	D.7	PropertySpec	E.60
CBOM	D.8	BillOfMaterials	E.9
CComponent	D.8	Part	E.39
CComponentReference	D.8	PartReference	E.40
CDistributionSystem	D.10	Distribution	E.20
CDistributionChannel	D.11	DistributionChannel	E.21
CLocation	D.11	Location	E.30
CTransportRelation	D.12	Transportrelation	E.81
CRoutingTable	D.13	RoutingTable	E.65
CRouting	D.13	Routing	E.64
CLogisticsServiceProvider	D.14	LogisticsServiceProvider	E.31
CLSPRoute	D.14	LSPRoute	E.32
CTransport	D.15	Transporter	E.80
CTransportDefinition	D.15	TransportDefinition	E.79
CContainer	D.16	Container	E.13
CRouteInformation	D.17	RouteInformation	E.63
CTransportOrder	D.18	—	—
CResourceAggregate	D.24	AggregatingResource	E.1
CResourceAssocData	D.26	AssocDatResource	E.6
CResourceProductionUnit	D.24	ProductionUnitResource	E.49
CResourceSum	D.24	SumResource	E.70
CResourceWeekTime	D.27	WeekTimeResource	E.86

Tabelle D.1.: Gegenüberstellung UML Klassen / GME-Klassen

D.2. Klassenstruktur CExperiment

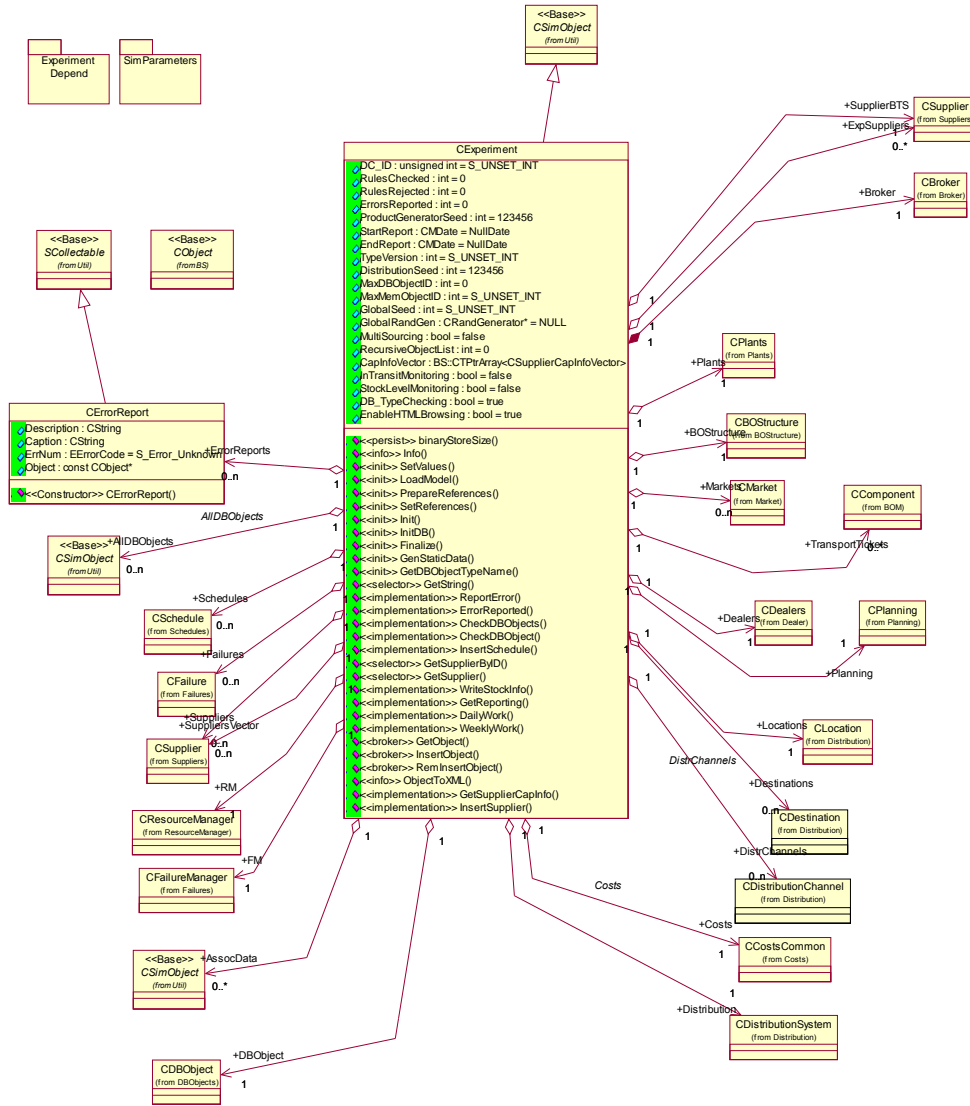


Abbildung D.2.: Klassenstruktur CExperiment

D.3. Klassenstruktur CSimParameters

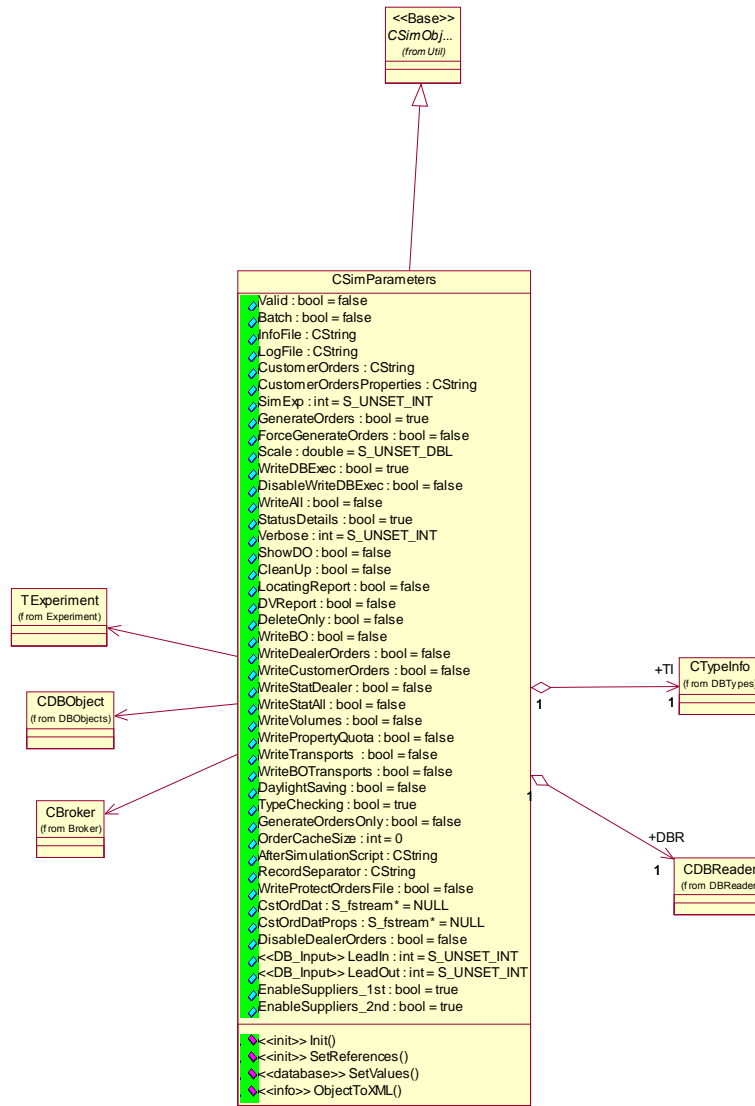


Abbildung D.3.: Klassenstruktur CSimParameters

D.4. Zentrale Klassen der Produktbeschreibung

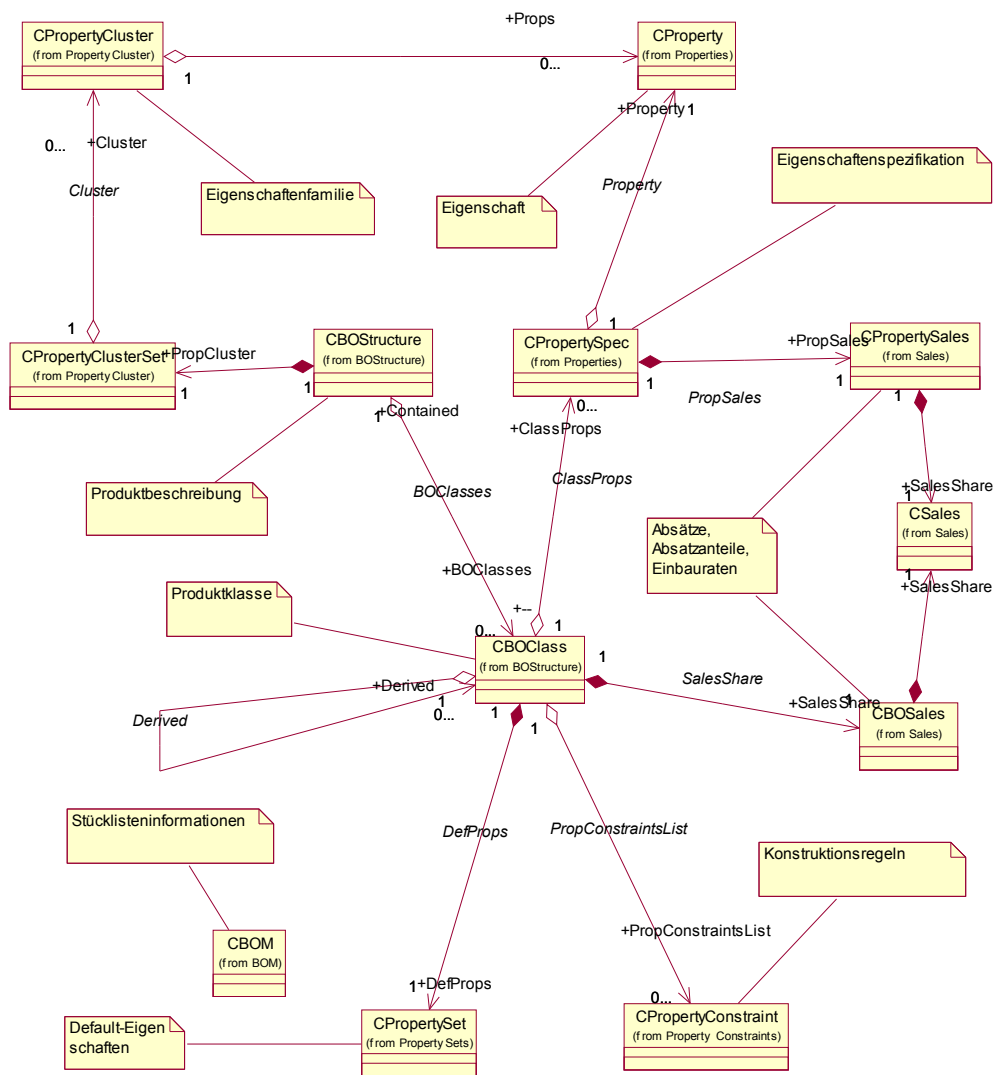


Abbildung D.4.: Zentrale Klassen der Produktbeschreibung

D.6. Klassenstruktur CPropertyCluster

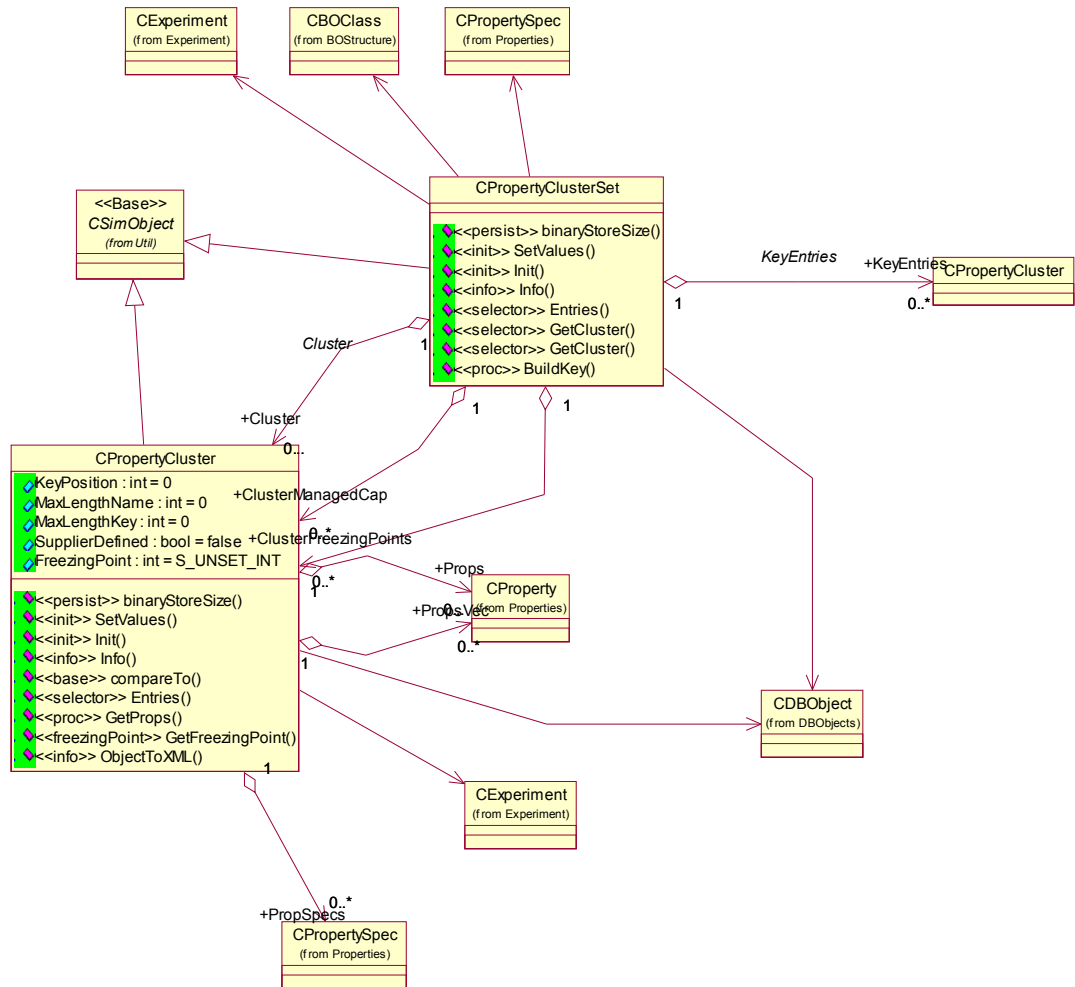


Abbildung D.6.: Klassenstruktur Eigenschaftenfamilien (CPropertyCluster)

D.7. Klassenstruktur CPropertySpec

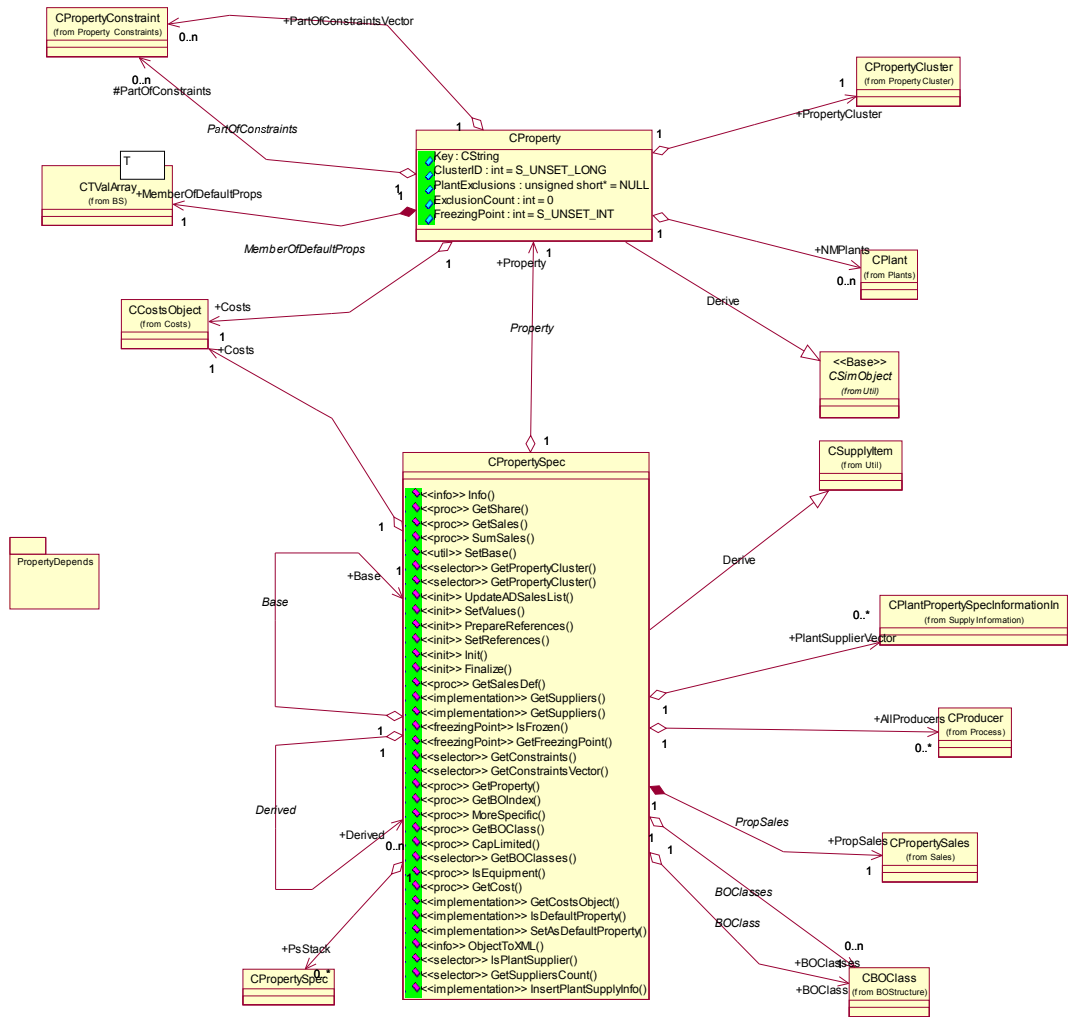


Abbildung D.7.: Klassenstruktur Eigenschaftenspezifikationen (CPropertySpec)

D.8. Klassenstruktur CBOM

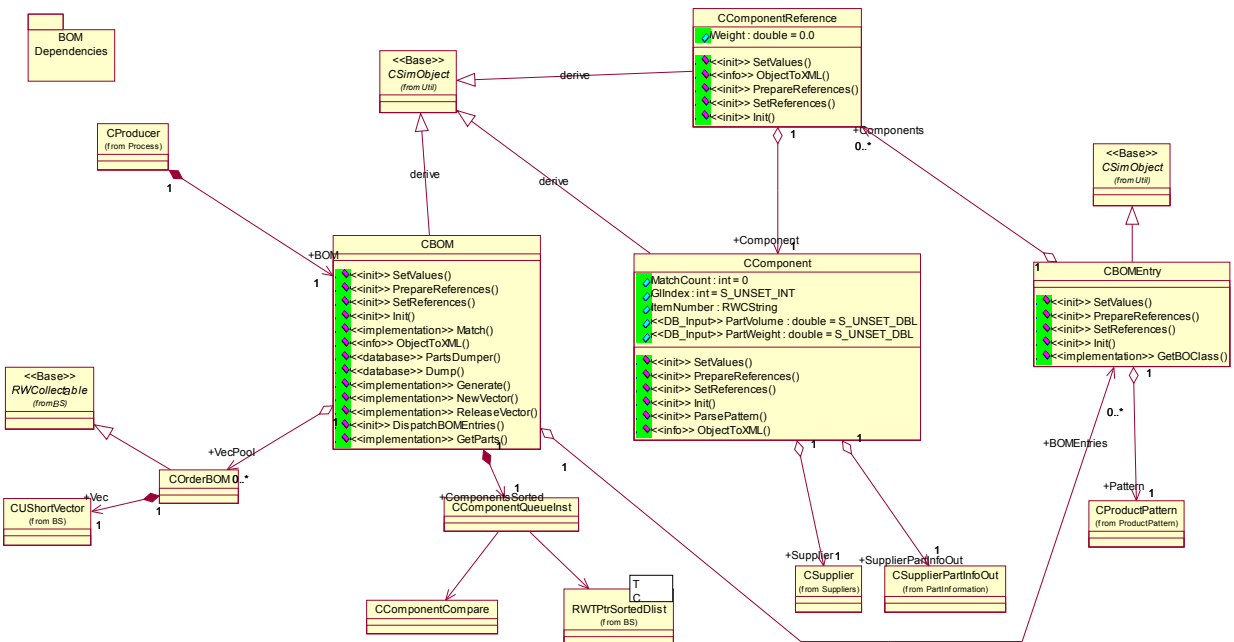


Abbildung D.8.: Klassenstruktur Stückliste (CBOM)

D.9. Klassenstruktur COrderModifier

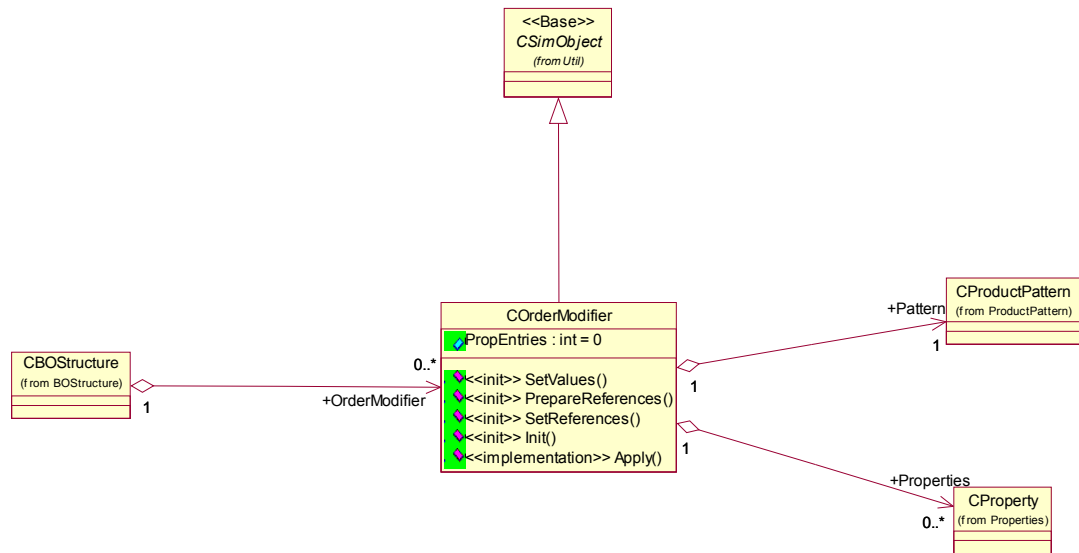


Abbildung D.9.: Klassenstruktur Auftragsanpassung (COrderModifier)

D.10. Klassenstruktur CDistributionSystem

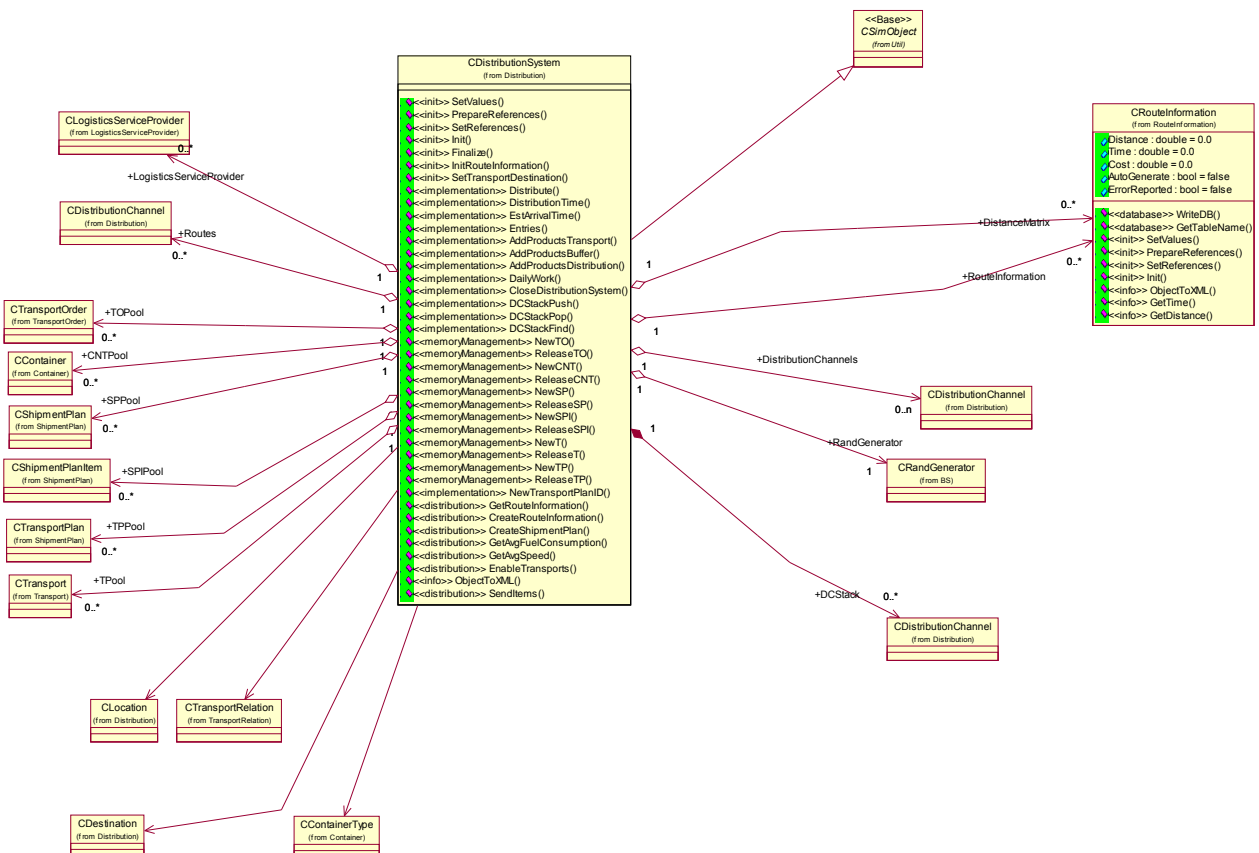


Abbildung D.10.: Klassenstruktur Distributionssystem (CDistributionSystem)

D.1.1. Klassenstruktur CDistributionChannel

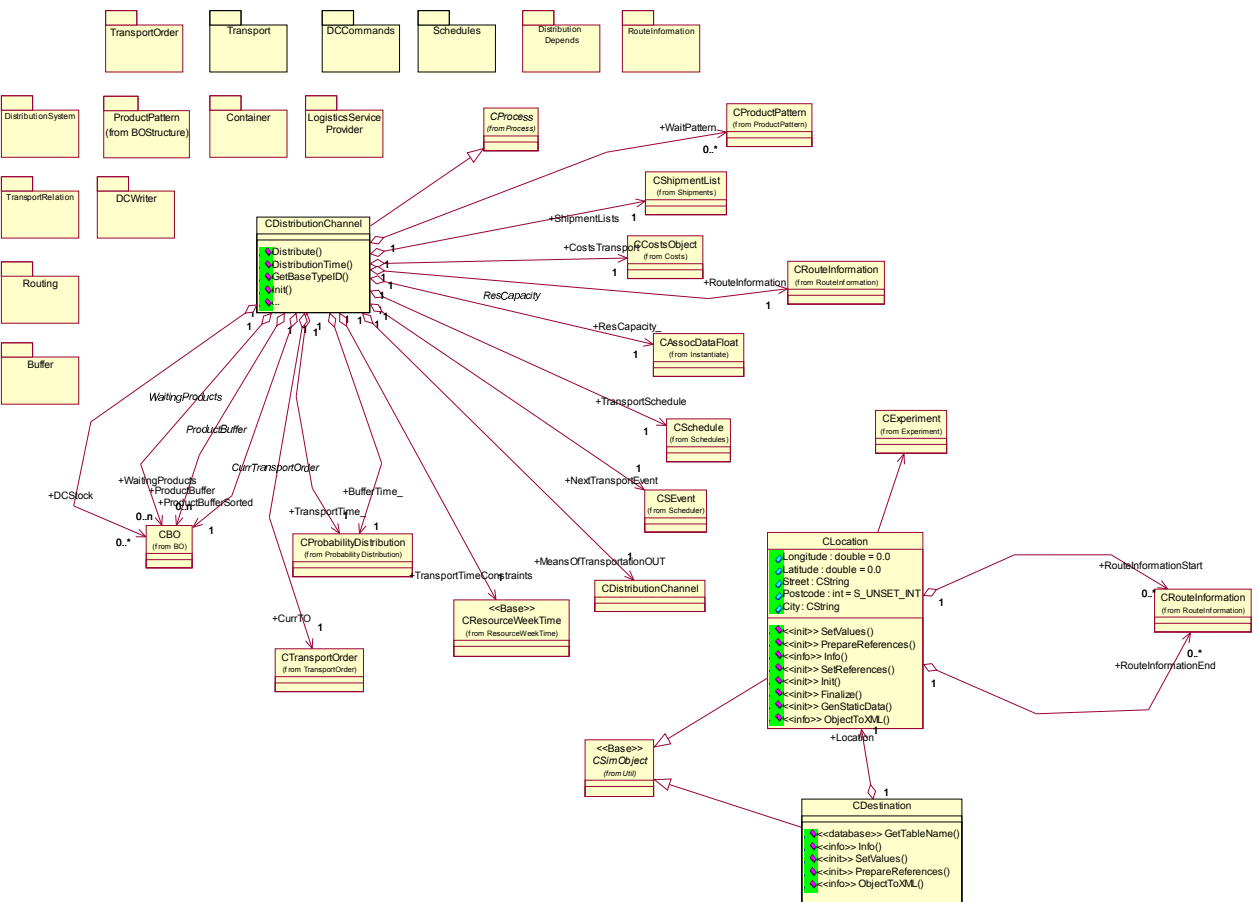


Abbildung D.1.1.: Klassenstruktur Distribution (CDistributionChannel)

D.12. Klassenstruktur CTransportRelation

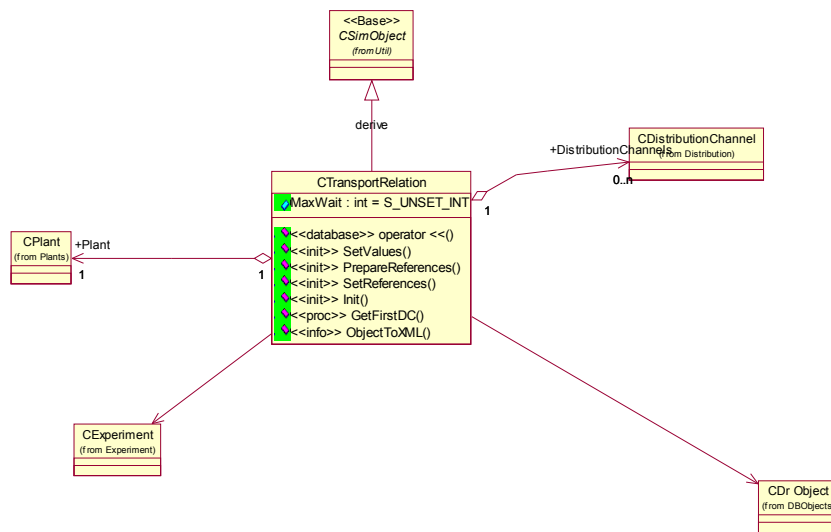


Abbildung D.12.: Klassenstruktur Transportrelation (CTransportRelation)

D.14. Klassenstruktur CLogisticsServiceProvider

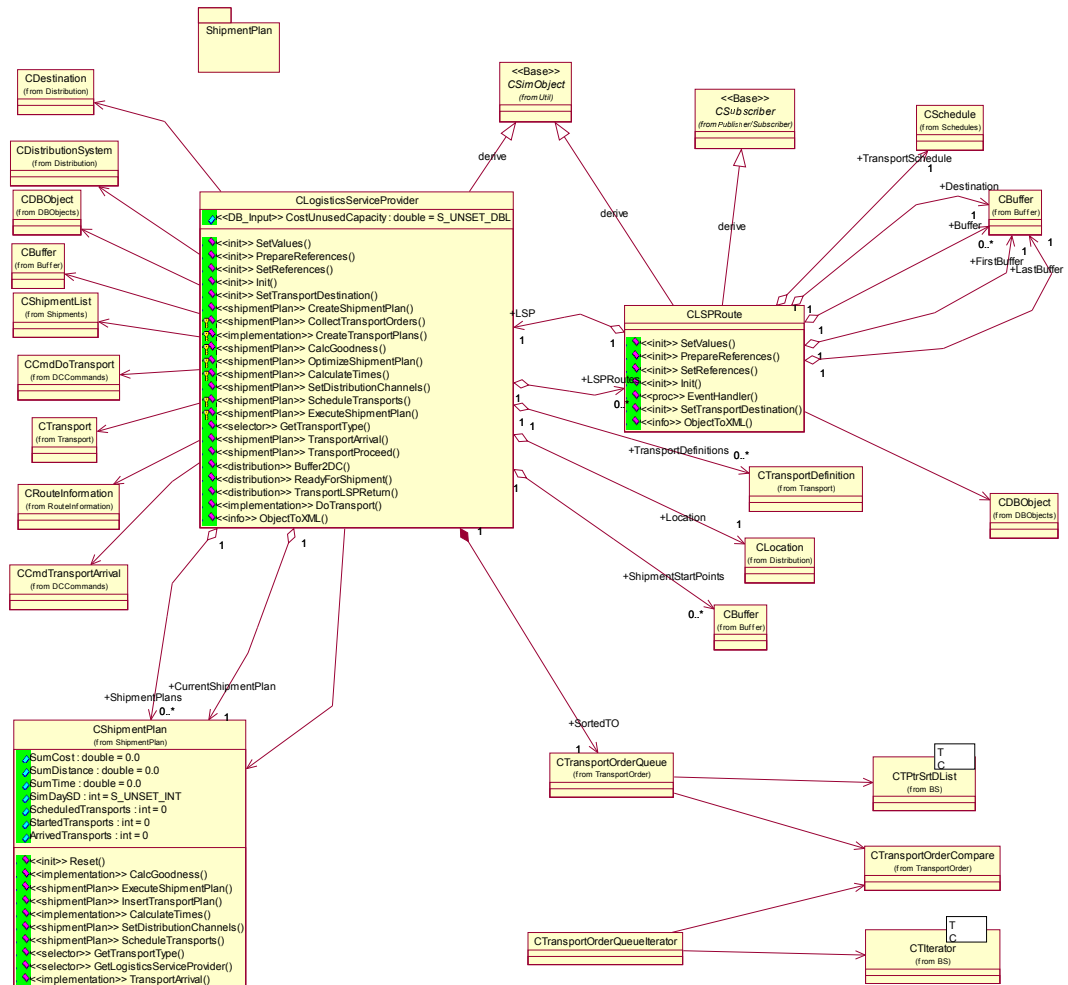


Abbildung D.14.: Klassenstruktur
(CLogisticsServiceProvider)

Logistikdienstleister

D.16. Klassenstruktur CContainer

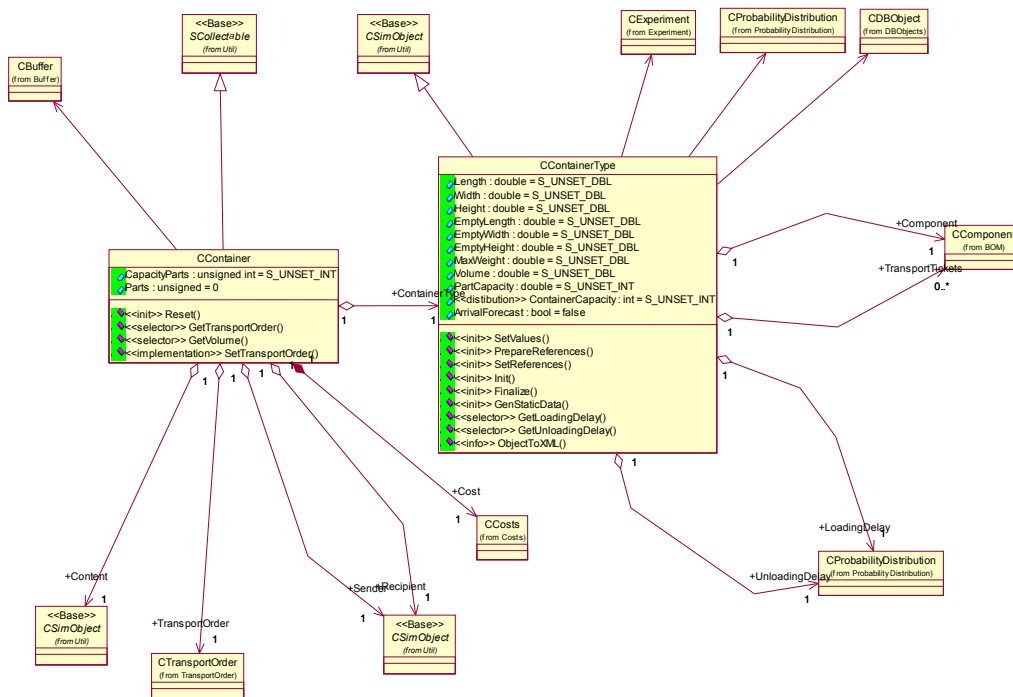


Abbildung D.16.: Klassenstruktur Behälter (CContainer)

D.17. Klassenstruktur CRouteInformation

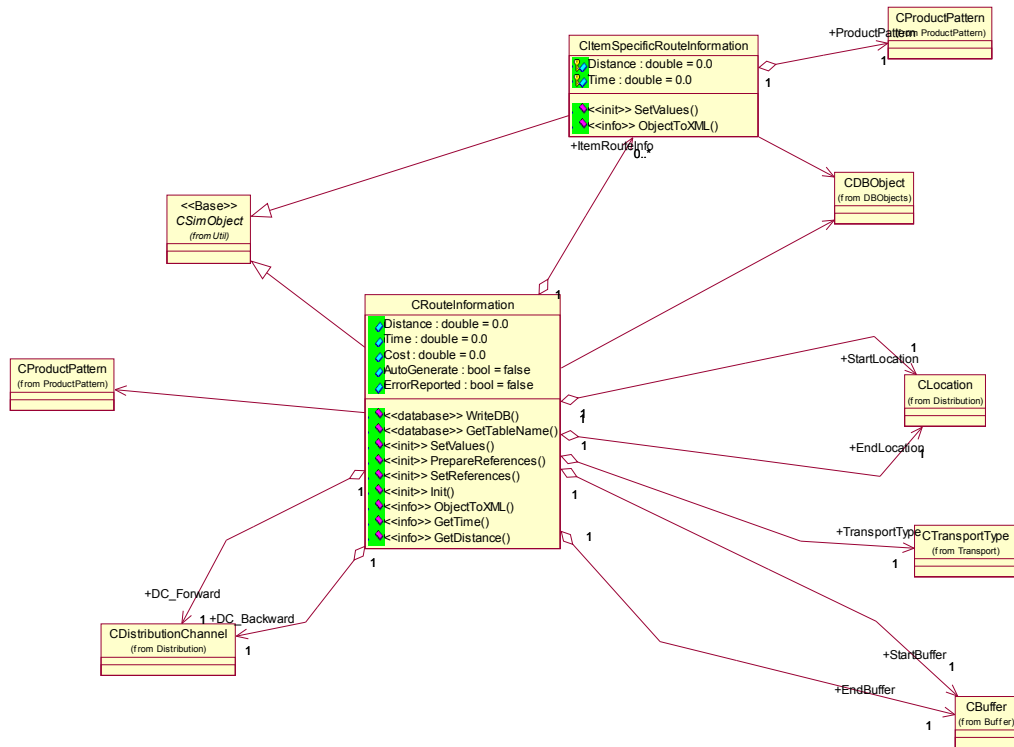


Abbildung D.17.: Klassenstruktur Distribution (CRouteInformation)

D.19. Klassenstruktur CMarket

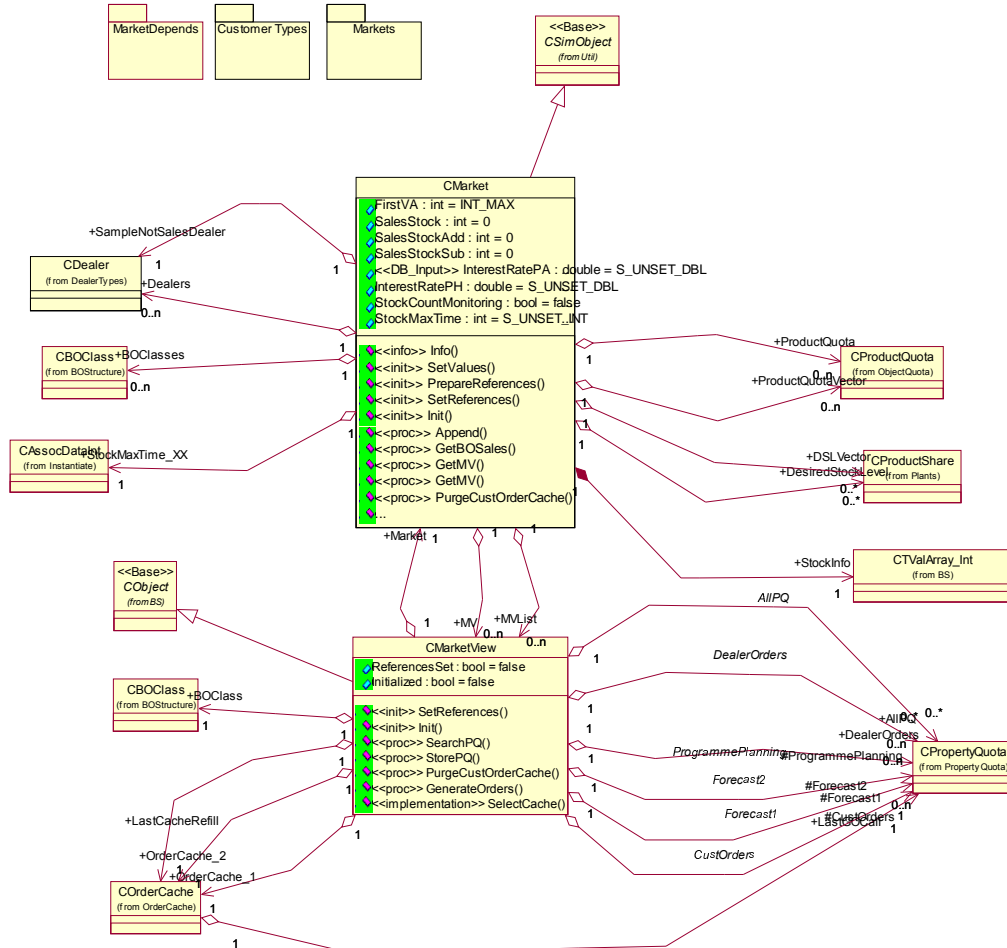


Abbildung D.19.: Klassenstruktur Markt (CMarket)

D.20. Klassenstruktur CDealer

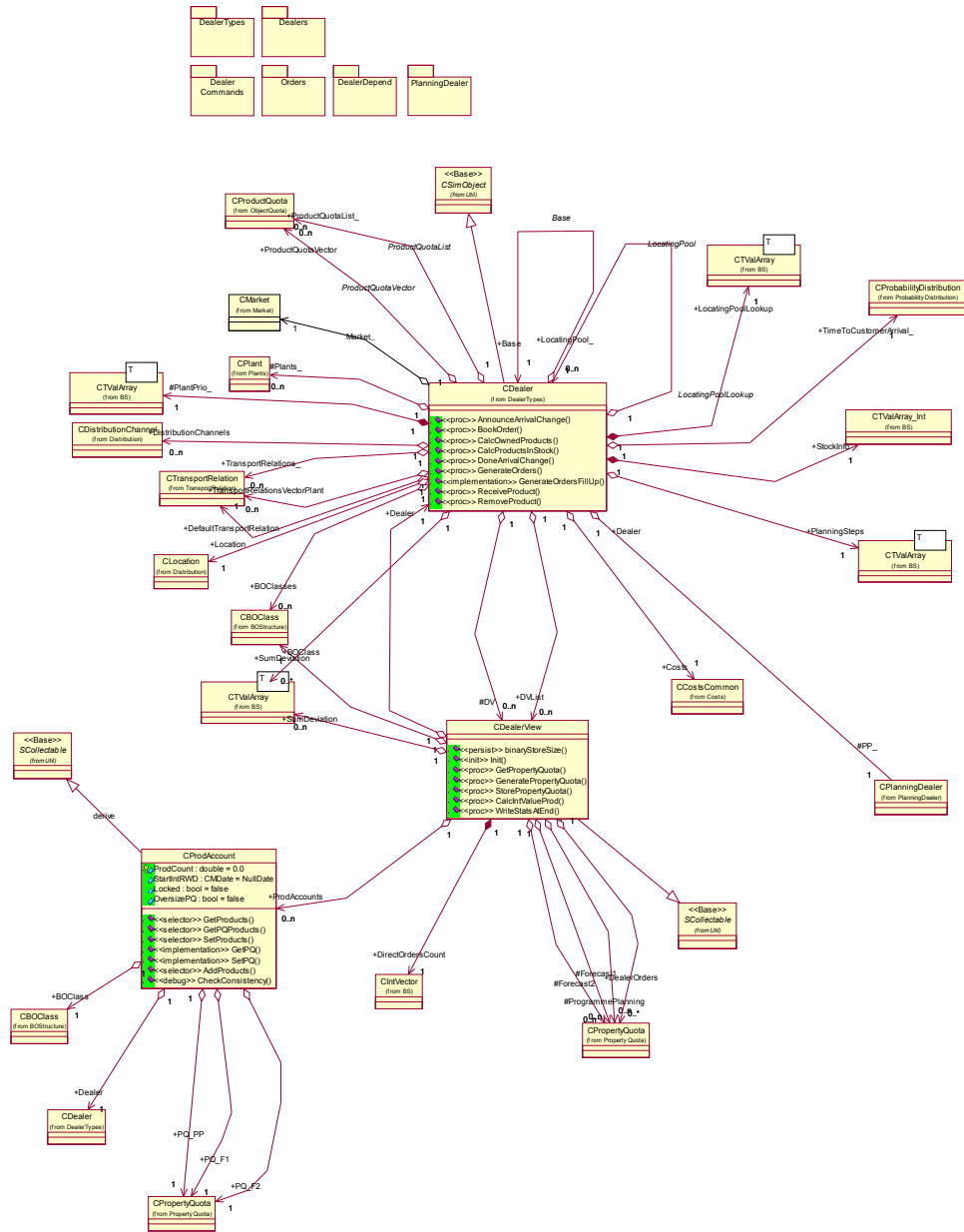


Abbildung D.20.: Klassenstruktur Händler (CDealer)

D.21. Klassenstruktur CDealer und CDealerOrderPool

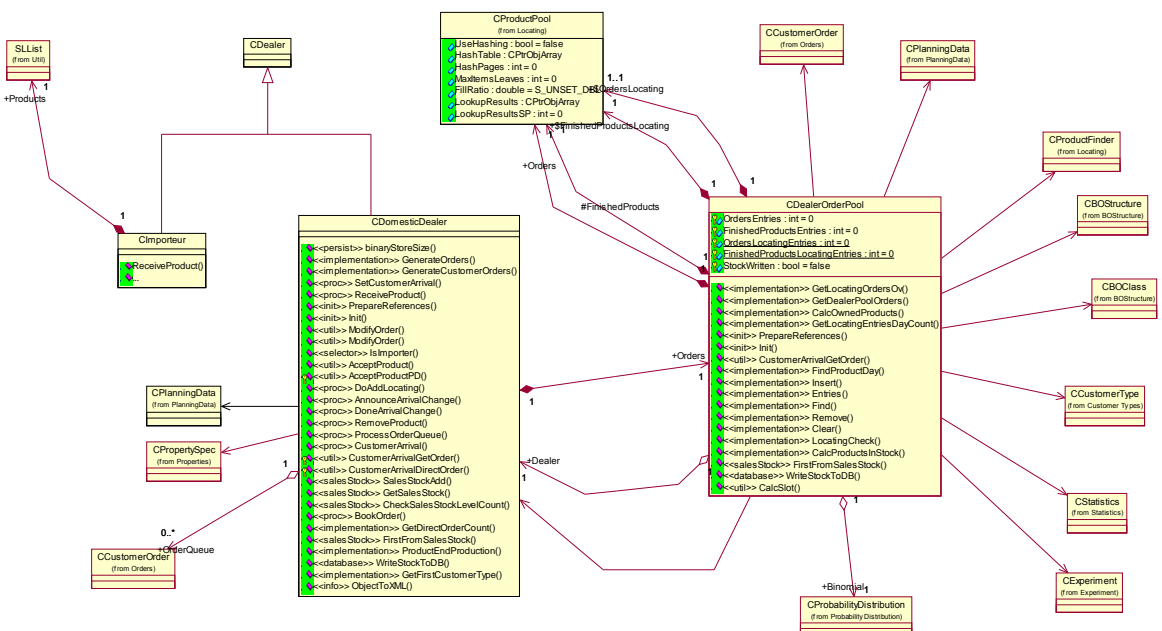


Abbildung D.21.: Klassenstruktur Händlertypen (CDealer, CDomesticDealer, CImporter)

D.22. Klassenstruktur CCustomerType

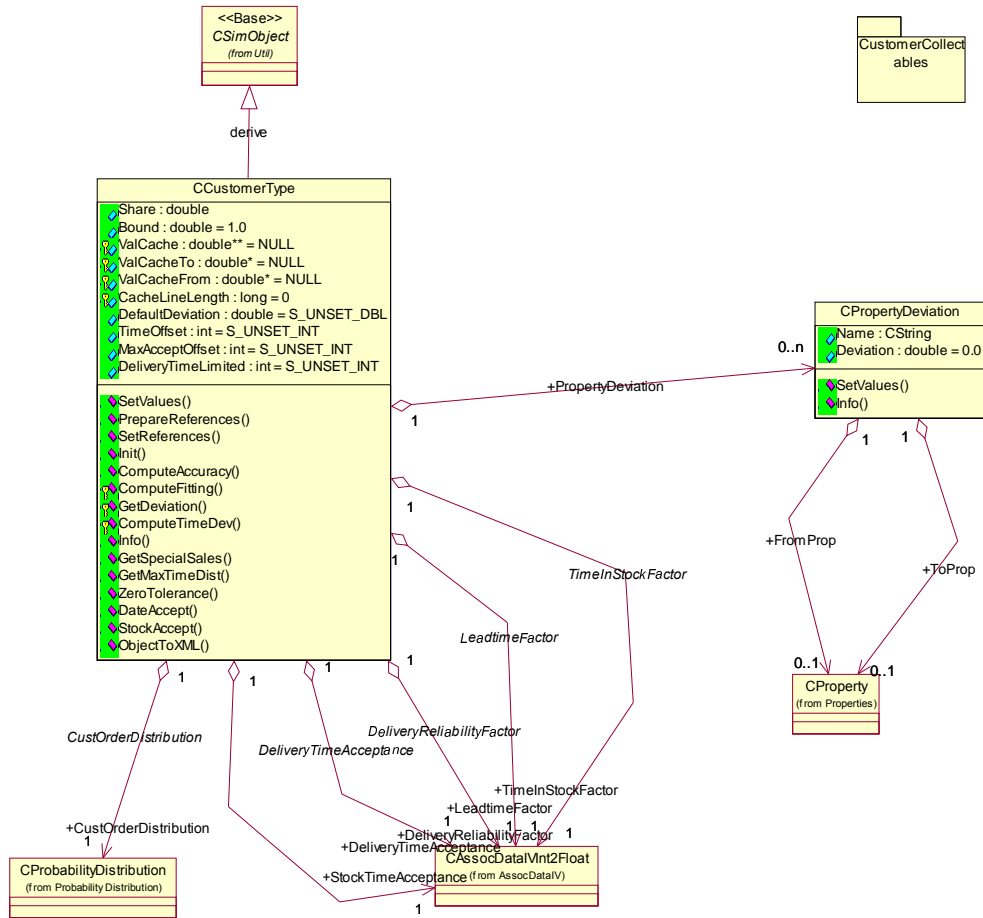


Abbildung D.22.: Klassenstruktur Kundentypen (CCustomerType)

D.23. Klassenstruktur CPlanningDealer

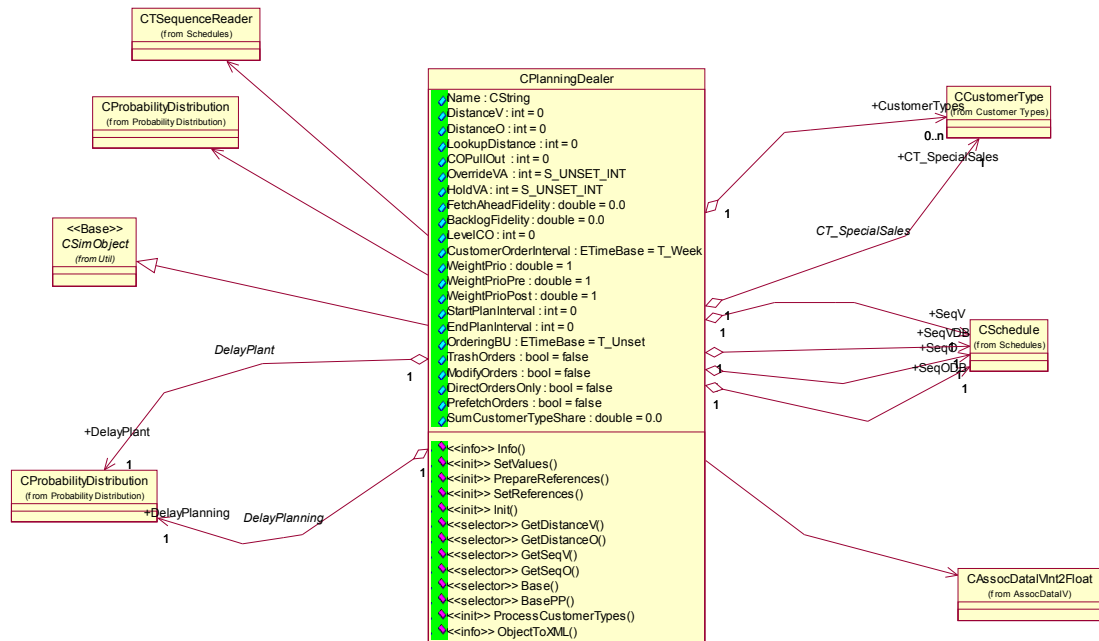


Abbildung D.23.: Klassenstruktur CPlanningDealer (CPlanningDealer) Planungsparameter Händler

D.24. Klassenstruktur CResource

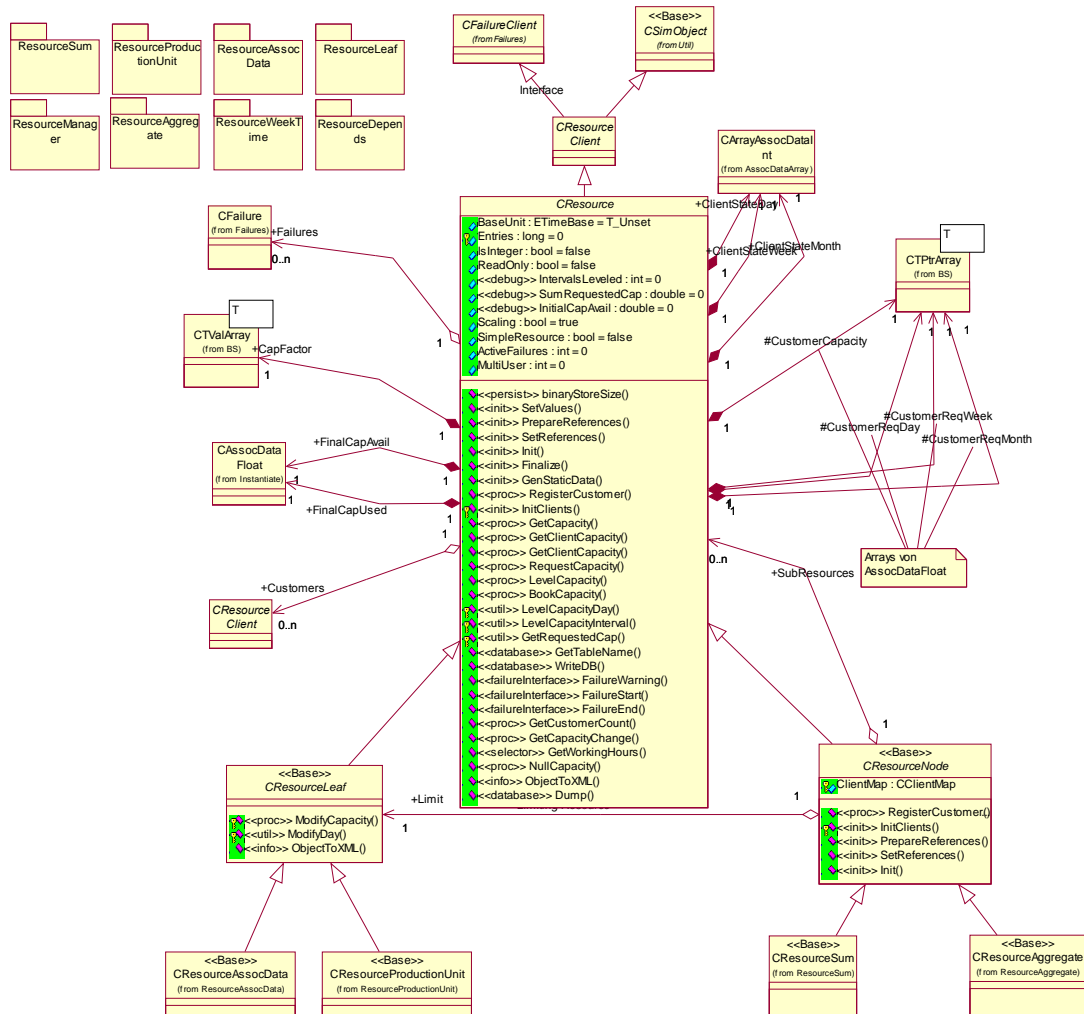


Abbildung D.24.: Klassenstruktur Ressourcen (CResource)

D.25. Klassenstruktur CResourceLeaf

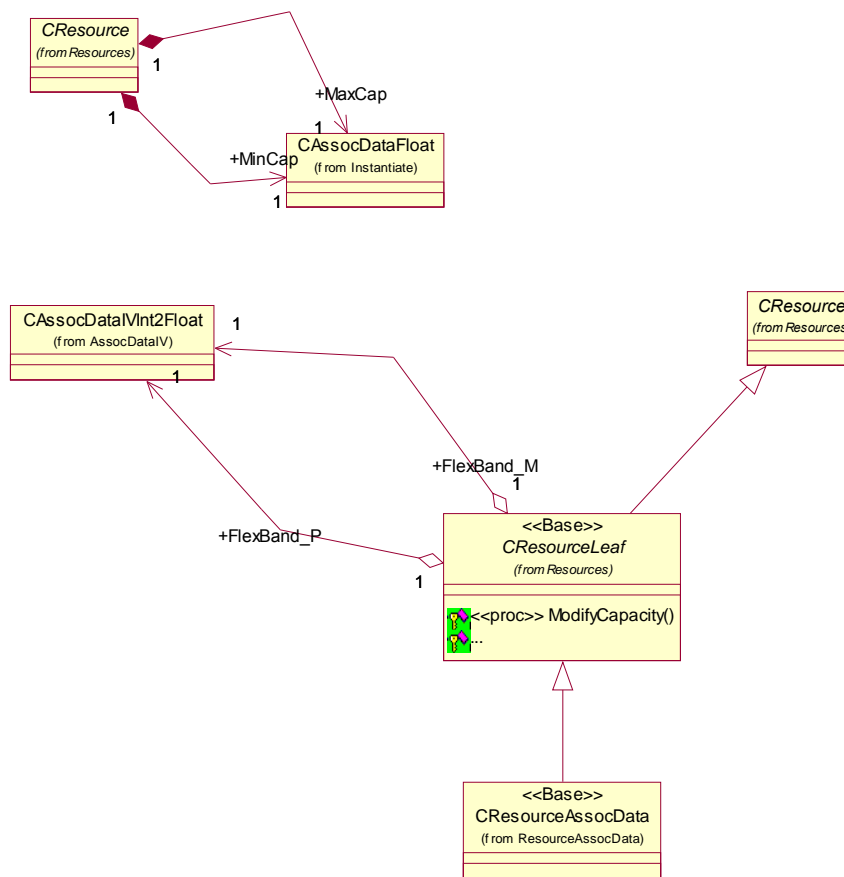


Abbildung D.25.: Klassenstruktur Ressourcen-Blatt (CResourceLeaf)

D.26. Klassenstruktur CResourceAssocData

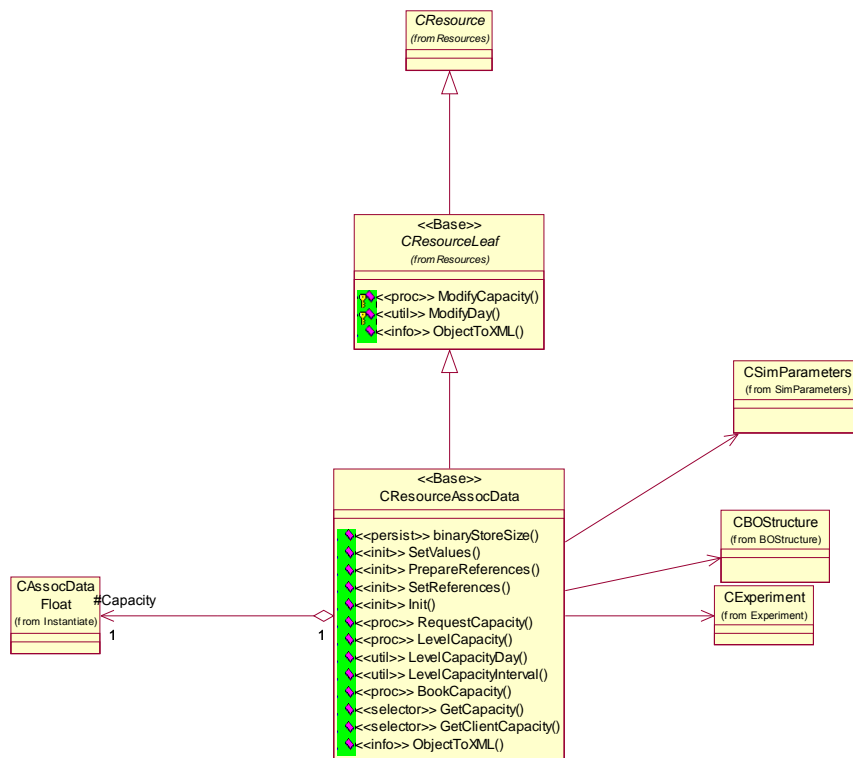


Abbildung D.26.: Klassenstruktur
(CResourceAssocData)

Wertverlauf-Ressourcen

D.27. Klassenstruktur CResourceWeekTime

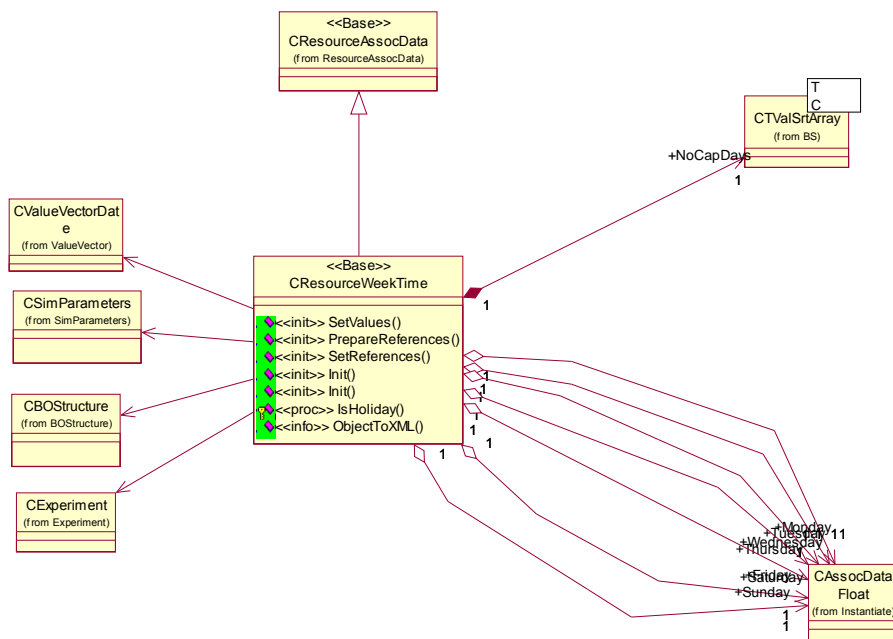


Abbildung D.27.: Klassenstruktur Wochenressource (CResourceWeekTime)

D.30. Klassenstruktur CFailure

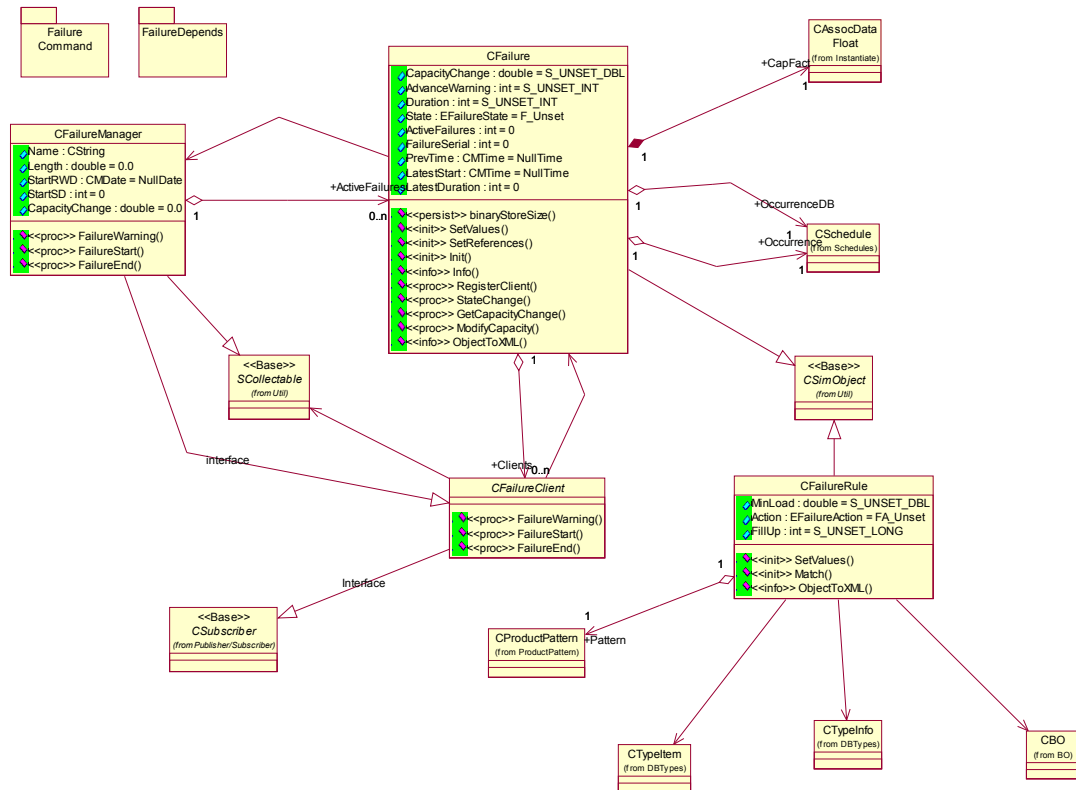


Abbildung D.30.: Klassenstruktur Störungen (CFailure)

D.31. Klassenstruktur CProductPattern

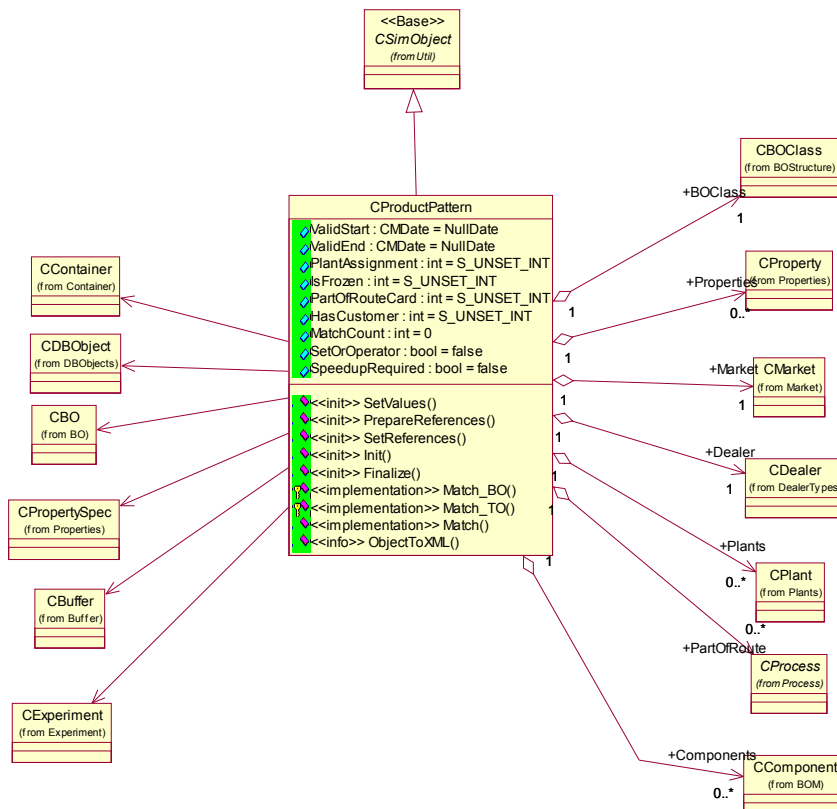


Abbildung D.31.: Klassenstruktur Suchmuster (CProductPattern)

D.32. Klassenstruktur CProbabilityDistribution

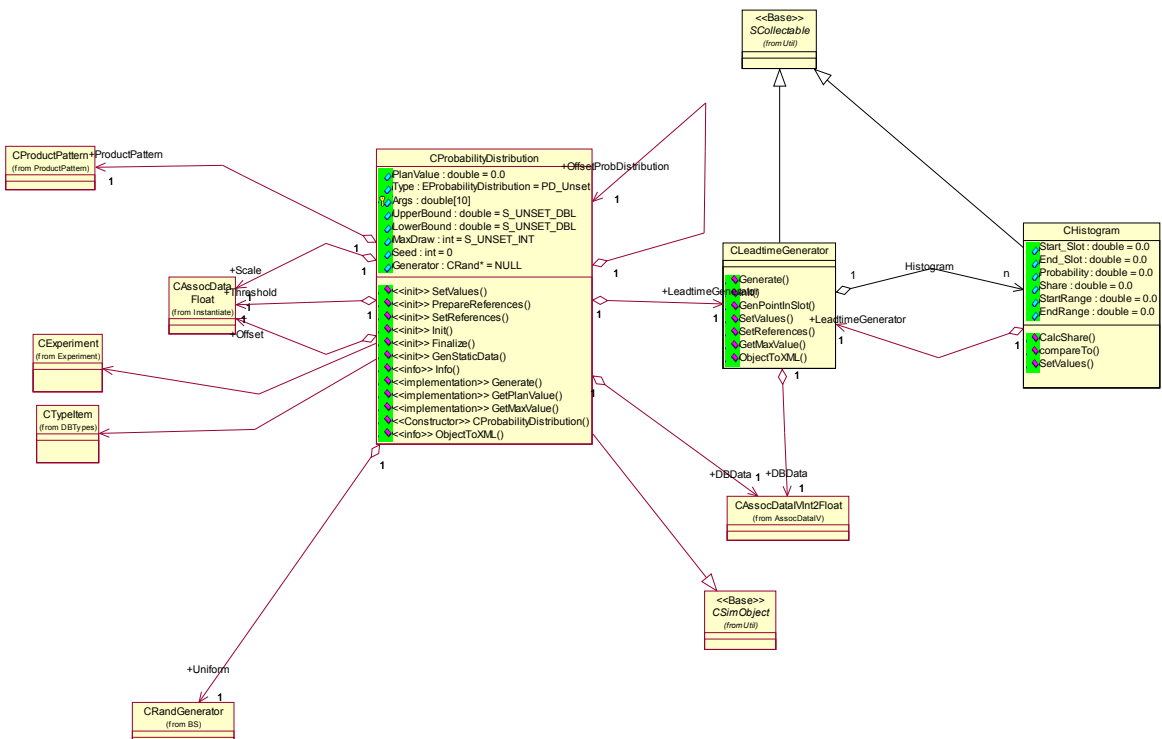


Abbildung D.32: Klassenstruktur Wahrscheinlichkeitsverteilung (CProbabilityDistribution)

D.33. Klassenstruktur COrderBank

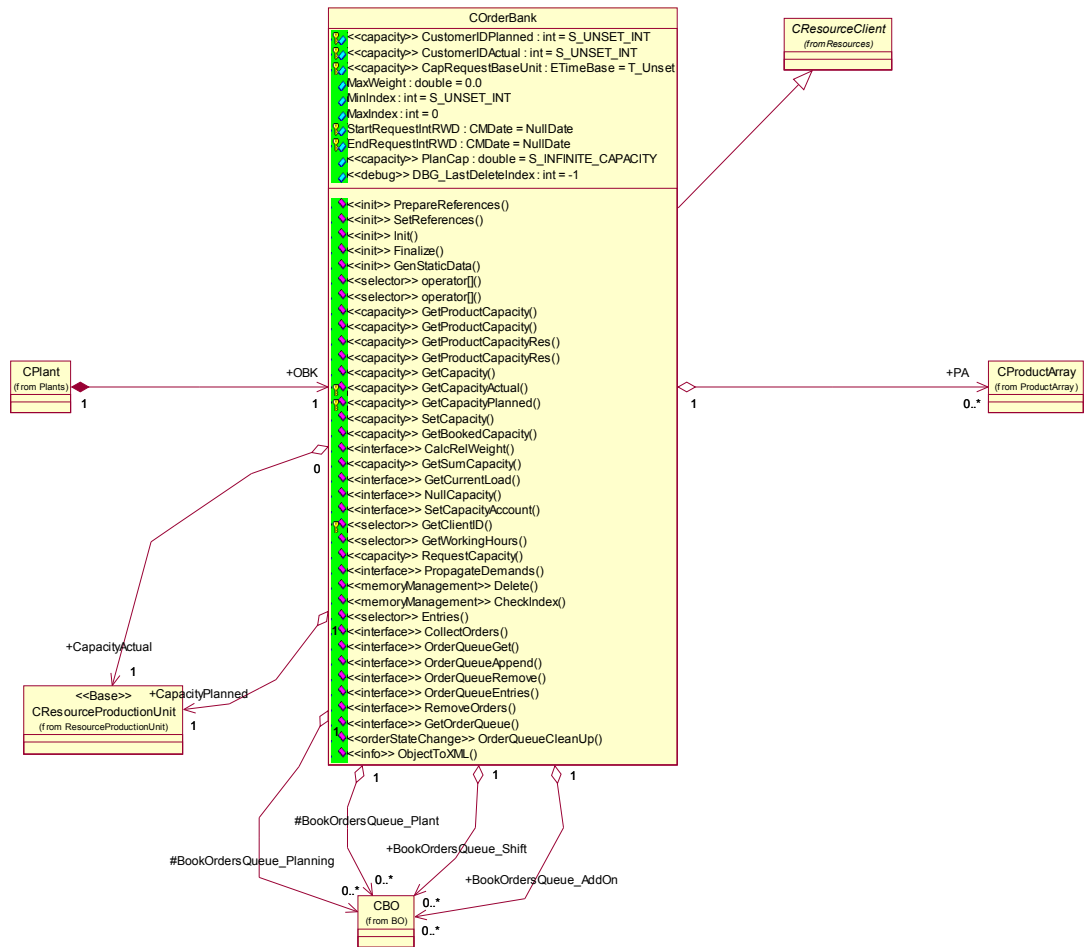


Abbildung D.33.: Klassenstruktur Auftragsverwaltung (COrderBank)

D.34. Klassenstruktur CPlanning

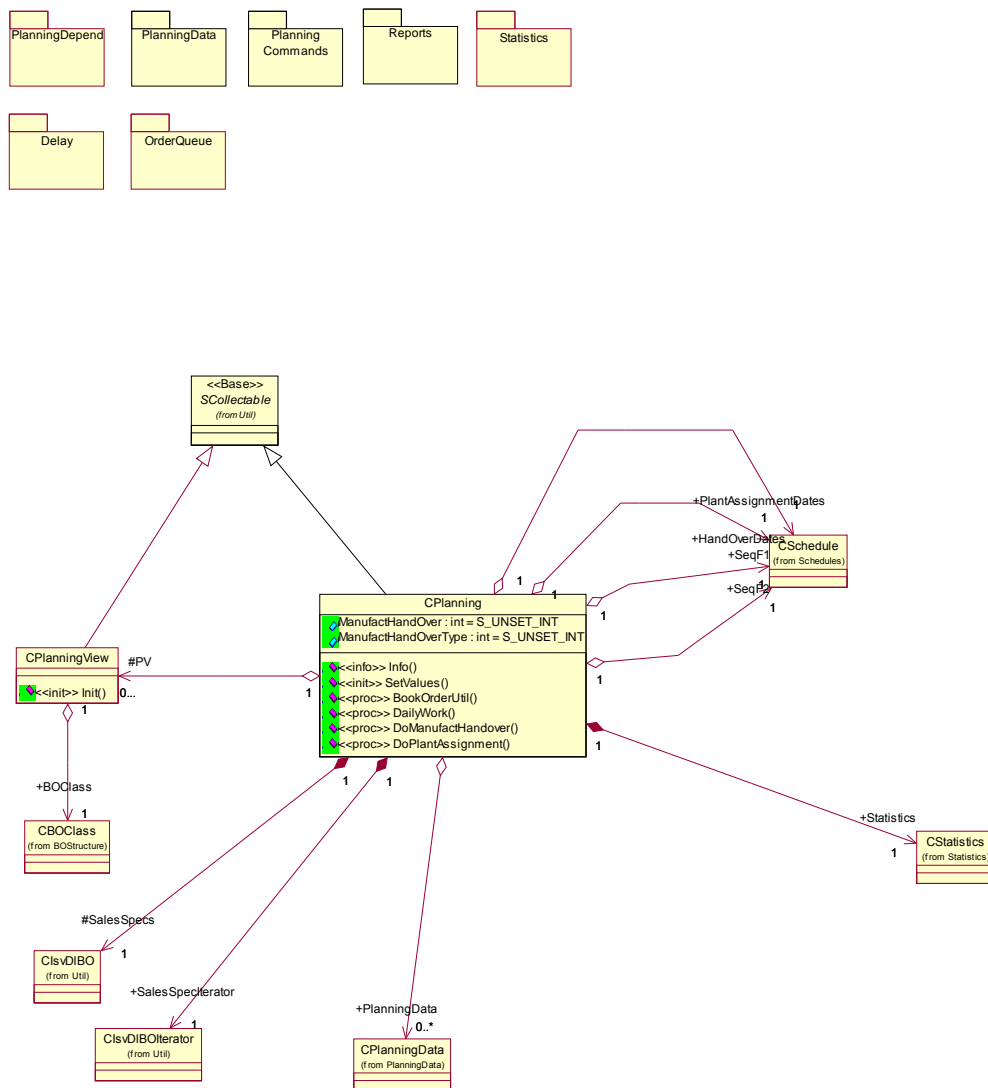


Abbildung D.34.: Klassenstruktur Prozesssteuerung (CPlanning)

D.35. Klassenstruktur CSupplier

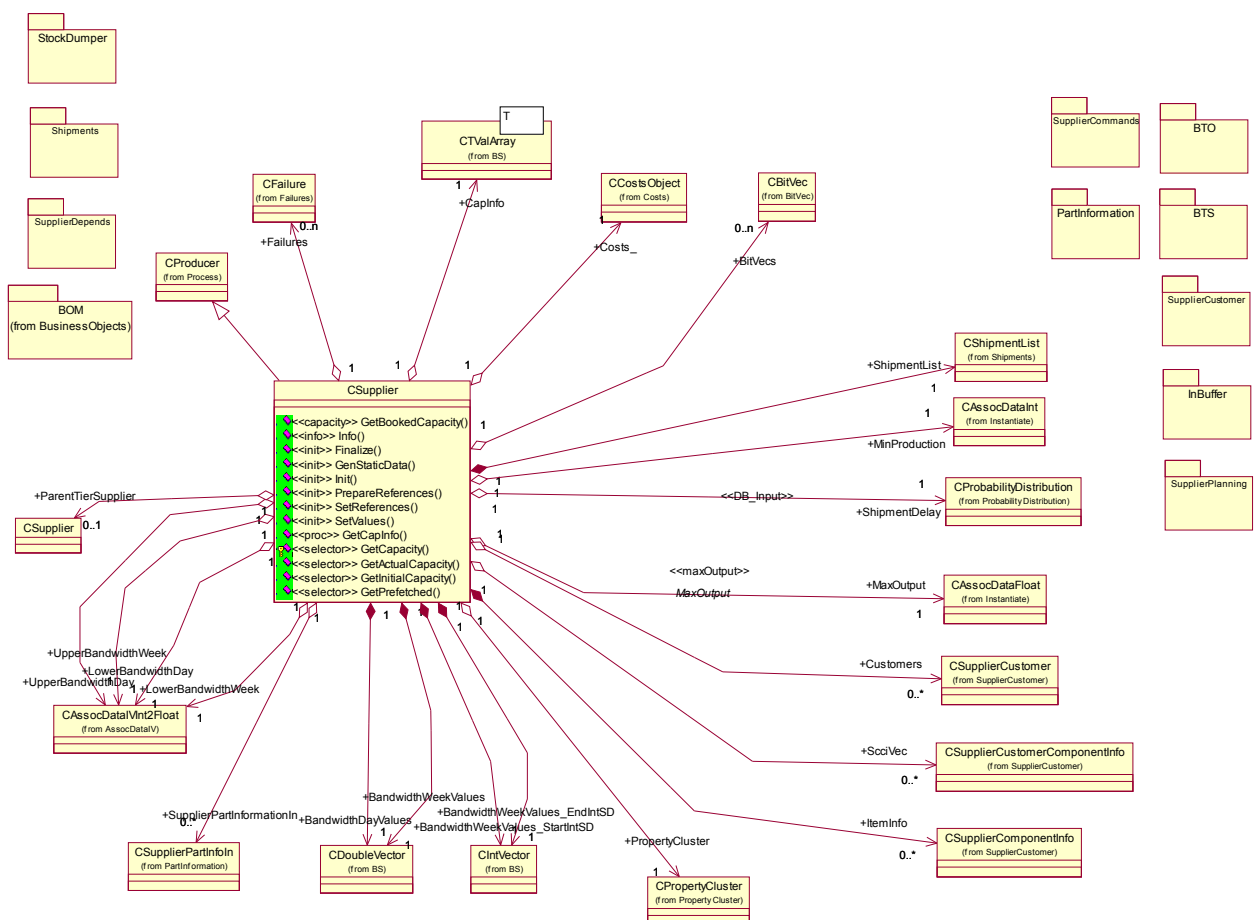


Abbildung D.35.: Klassenstruktur Zulieferer (Basis) (CSupplier)

D.36. Klassenstruktur CSupplierBTO

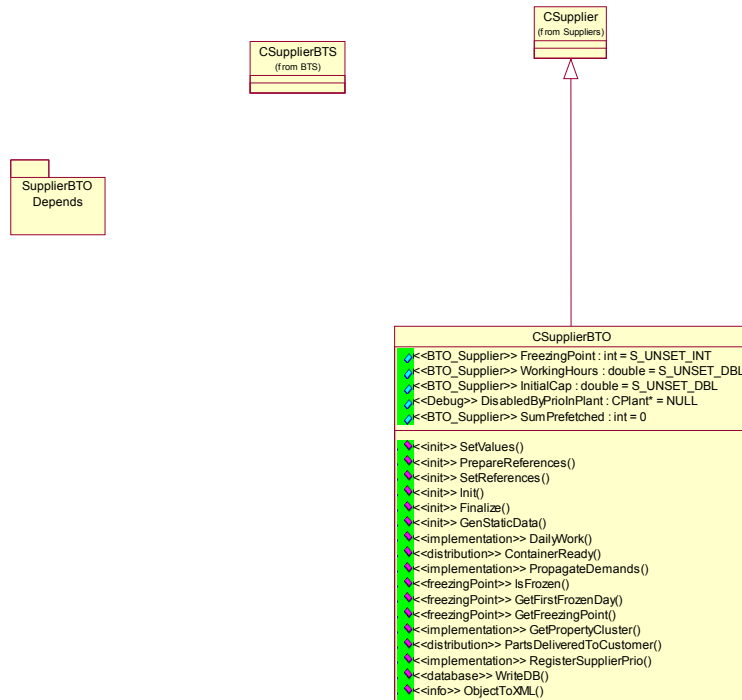


Abbildung D.36.: Klassenstruktur Zulieferer (BTO) (CSupplierBTO)

D.37. Klassenstruktur CSupplierBTS

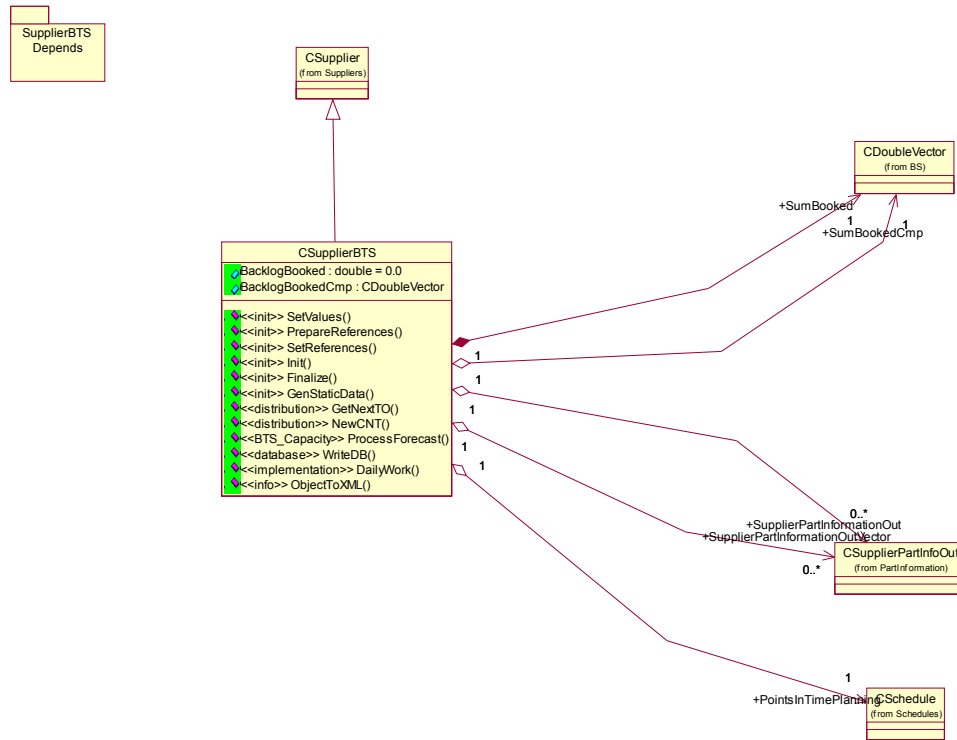


Abbildung D.37.: Klassenstruktur Zulieferer (BTS) (CSupplierBTS)

D.38. Klassenstruktur CSupplierCustomer

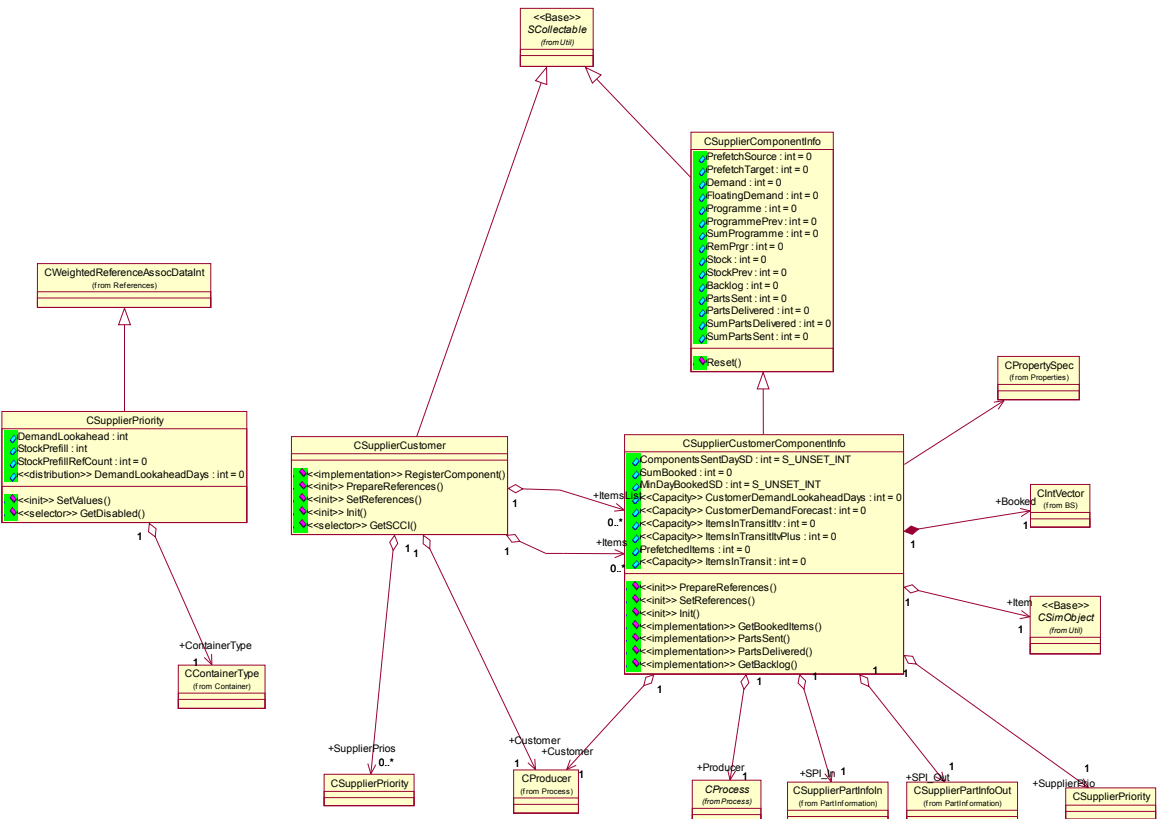


Abbildung D.38: Klassenstruktur zur Verwaltung von Abnehmern der Zulieferer (BTO) (CSupplierCustomer)

D.39. Klassenstruktur CAssocData

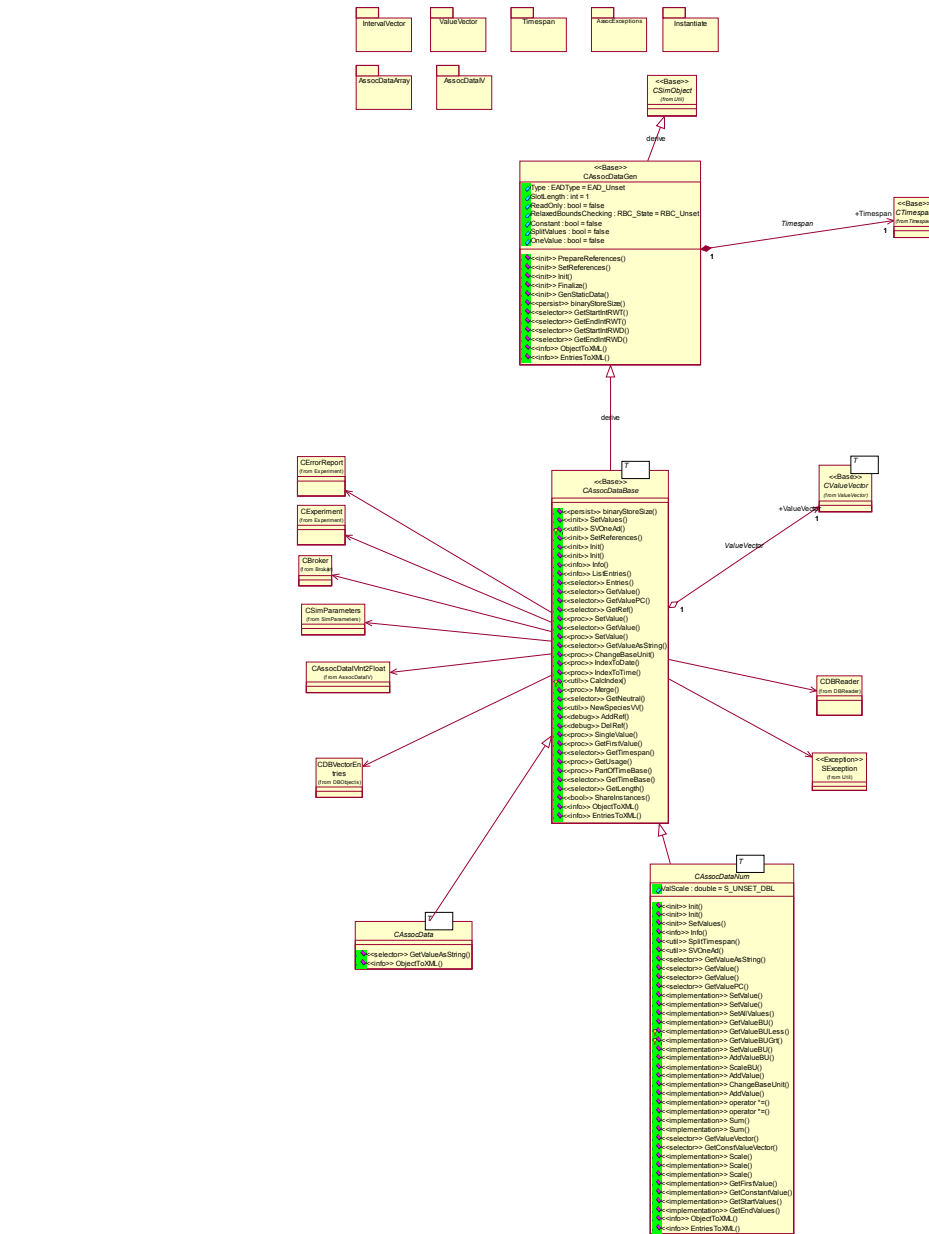


Abbildung D.39.: Klassenstruktur Wertverlauf (CAssocData)

D.40. Von CAssocData abgeleitete Klassen

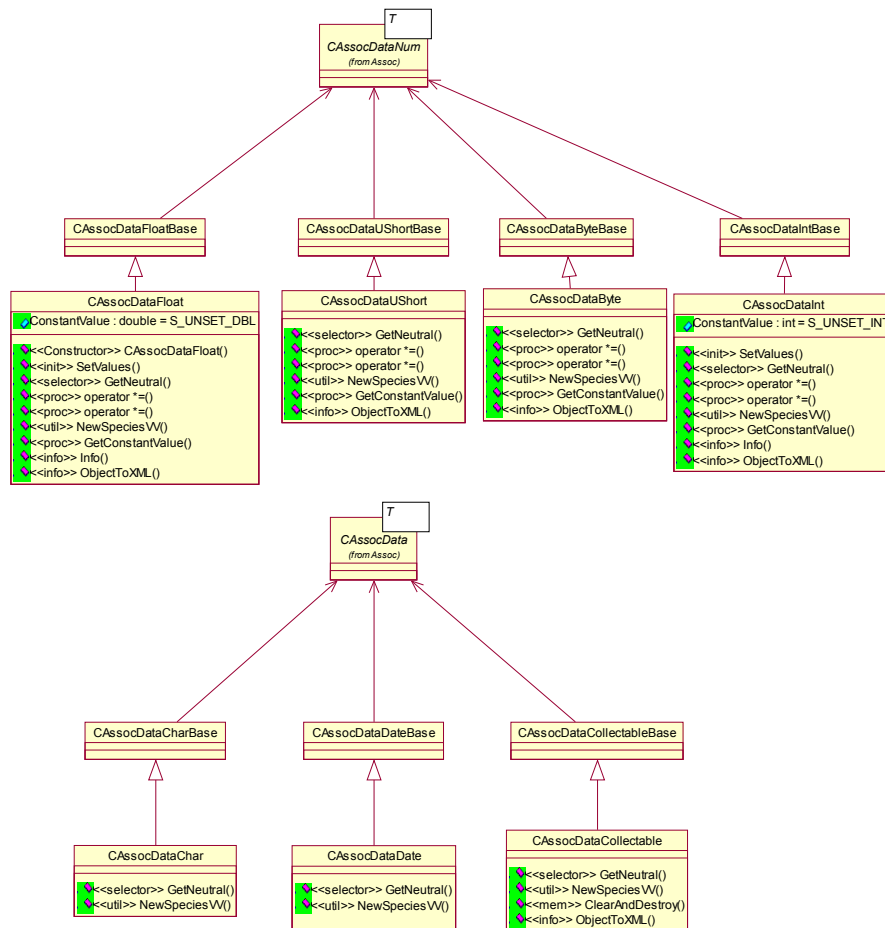


Abbildung D.40.: Klassenstrukturen weiterer Wertverläufe

D.41. Klassenstruktur CSchedule

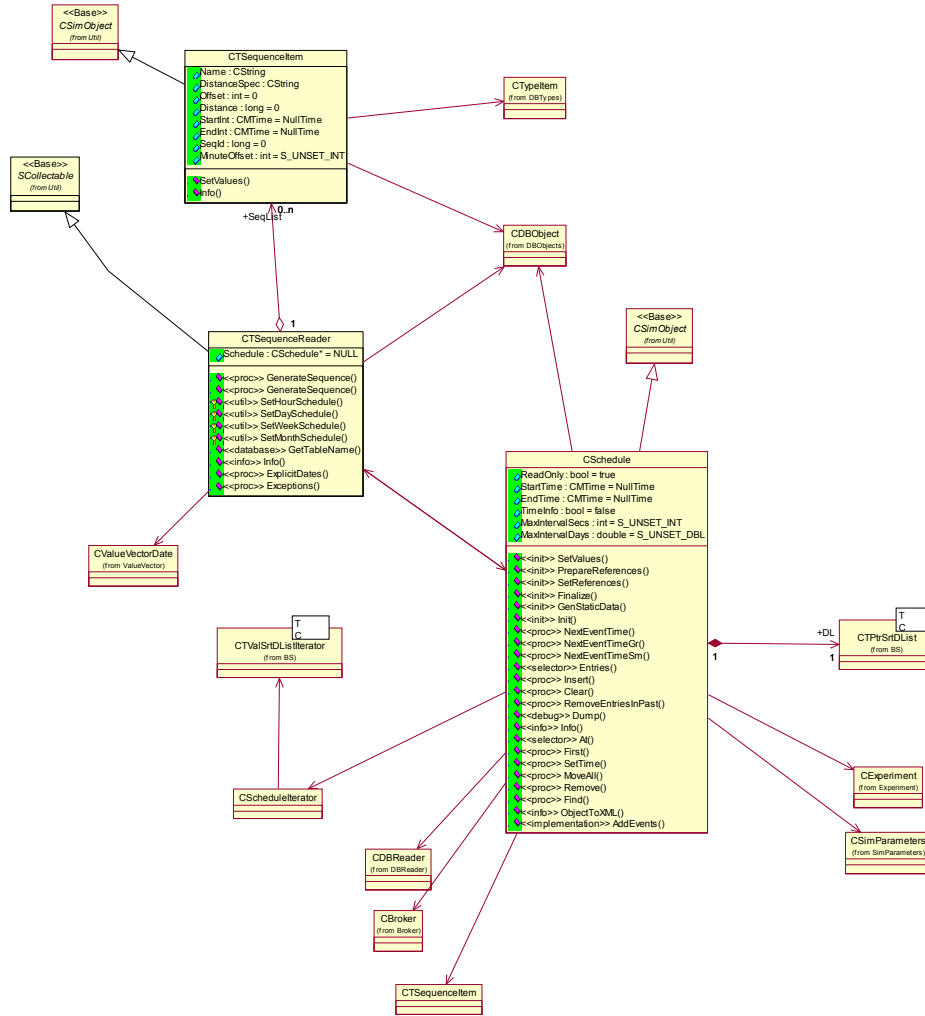


Abbildung D.41.: Klasse CSchedule

E. Anhang – Ausgewählte OTD-NET-Klassen

E.1. Klassenstruktur AGGREGATINGRESOURCE

Basisklasse: RESOURCE (siehe Tabelle E.62)

Attribut / Typ	Beschreibung
Kategorie: Kapazität	
.ISINTEGER <i>Bool</i>	Gibt an, ob für die Kapazität nur ganze Zahlen zulässig sind.
.SIMPLERESOURCE <i>Bool</i>	Ressource stellt Kapazität stets vollständig zur Verfügung.
.SUBRESOURCES RESOURCE 0..n →	Eine Liste mit untergeordneten Ressourcen. Die Position in der Liste gibt an, wieviele Einheiten die aktuelle Ressource von der untergeordneten Ressource zur Bereitstellung einer Einheit benötigt.
Kategorie: Störungen	
.FAULTS FAULT 0..n →	Liste aller Störungen, die diese Ressource betreffen.

Tabelle E.1.: AggregatingResource

E.2. Klassenstruktur ASS_DTA

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.CONSTANTVALUE <i>Float</i>	Ein konstanter Wert ersetzt die Wertreihe. Ein Intervallvektor sollte nicht spezifiziert werden und wird überlesen.
.VALSCALE <i>Float</i>	Alle Werte im Werteverlauf werden mit diesem Faktor multipliziert.

Tabelle E.2.: ASS_DTA

E.3. Klassenstruktur ASSOCDAT

Basisklasse: ASS_DTA (siehe Tabelle E.2)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.ONEVALUE <i>Bool</i>	Stehen in einer Wertreihe, in einem Wertverlauf lauter gleiche Werte, kann hier <<Ja>> spezifiziert werden. Es muss dann lediglich dieser eine Wert angegeben werden. Bsp. Für 24 Monate wird stets eine Kapazität von 100 angegeben. Nach dem Setzen dieser Variable auf <<Ja>> wird nur noch ein Wert erwartet (24 * 100 = 2400).

.SPLITVALUES <i>Bool</i>	Die spezifizierten Werte werden auf kleinere Einheiten verteilt, wenn <<Ja>>. Bsp.: Für den Monat April werden 3000 Einheiten spezifiziert und es wird ein Wert für einen Tag vom Simulator abgefragt. Hat diese Variable den Wert <<Wahr>>, wird ein Wert von 100 (3000 Einheiten / 30 Tage) berechnet. Hat die Variable den Wert <<Nein>>, wird der Wert 3000 berechnet.
-----------------------------	--

Tabelle E.3.: AssocDat

E.4. Klassenstruktur ASSOCDAT_TIMESPAN_FLOAT

Basisklasse: ASSOCDAT (siehe Tabelle E.3)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.4.: AssocDat_Timespan_Float

E.5. Klassenstruktur ASSOCDATCAPACITY

Basisklasse: ASSOCDAT_TIMESPAN_FLOAT (siehe Tabelle E.4)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.TIMESPANVECTOR TIMESPAN 0..n →	Intervalldefinition
.VALUEVECTOR VECTORCAPACITY 0..1 →	Wertreihe

Tabelle E.5.: AssocDatCapacity

E.6. Klassenstruktur ASSOCDATRESOURCE

Basisklasse: RESOURCE (siehe Tabelle E.62)

Attribut / Typ	Beschreibung
Kategorie: Kapazität	
.CAPACITY ASSOCDATCAPACITY 0..1 →	Kapazität
.FLEXBAND_M ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Bandbreite, innerhalb der die Kapazität nach unten verändert werden darf.
.FLEXBAND_P ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Bandbreite, innerhalb der die Kapazität nach oben verändert werden darf.
.ISINTEGER <i>Bool</i>	Gibt an, ob für die Kapazität nur ganze Zahlen zulässig sind.
.SIMPLERESOURCE <i>Bool</i>	Ressource stellt Kapazität stets vollständig zur Verfügung.
Kategorie: Störungen	
.FAULTS FAULT 0..n →	Liste aller Störungen, die diese Ressource betreffen.

Tabelle E.6.: AssocDatRessource

E.7. Klassenstruktur BASEDEALER

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.BASEOBJECT BASEDEALER 0..1 →	Objekt, von dem die Attributwerte übernommen werden, wenn in dem aktuellen Objekt keine eigenen Attributwerte definiert sind.
.PRODUCTQUOTA PRODUCTQUOTA 0..n →	Jede Fahrzeugbeschreibung, die auf dem Markt verkauft wird, muss beim Händler mit seinem Marktanteil definiert sein. Dieses Attribut ist eine Alternative zum Attribut "Marktanteil", um das Händlerverhalten weiter auszudifferenzieren.
.QUOTA Float	Wenn der Marktanteil gesetzt ist, wird das Attribut "Fahrzeugbeschreibungen und Anteile" ignoriert.
.REPORT Bool	Gibt an, ob für den Händler Statistiken in die Datenbank geschrieben werden sollen. Default: Ergebnisse werden in DB geschrieben.
Kategorie: Datenimport	
.ALIASNAMES String	Namen von Händlern, Importeuren o.ä., die beim Datenimport diesem Händler zugeordnet werden.
Kategorie: Distribution	
.DESTINATION DESTINATION 0..1 →	Der Zielbahnhof des Händlers
.TRANSPORTRELATIONS TRANSPORTRELATION 0..n →	Verbindungen zu Werken (Distributionskanäle) und händlerspezifischen Parametern.
Kategorie: Einplanung	
.DIRECTORDERING Bool	Stehen beim Händler keine Aufträge zur Verfügung, die einem Kunden zugeteilt werden können, warten die Kunden nicht auf den nächsten freien Auftrag; es werden Direktverbuchungen für diese Kunden durchgeführt.
.PLANNING DEALERPLANNING 0..1 →	Händler Planungsparameter
.PLANTS PLANT 0..1 →	Liste aller Werke und zugeordnete Priorität, die angibt, von welchem Werk der Händler beliefert werden will.
Kategorie: Kosten	
.COSTS COSTSCOMMON 0..1 →	Gemeinkosten Händler
Kategorie: Kundenverhalten	
.TIMETOCUSTOMERARRIVAL PROBABILITYDISTRIBUTION 0..1 →	Zeit von der Ankunft beim Händler/Importeur bis zur Übergabe an einen (evtl. anonymen) Kunden. Bei Anlieferung eines Fahrzeugs beim Händler wird eine Übergabezeit gezogen. Nach Ablauf der Übergabezeit wird das Fahrzeug dem Kunden übergeben. Wird das Fahrzeug erheblich vor dem Kundenwuschtermin dem Händler geliefert und der Wuschtermin liegt zeitlich hinter der Ankunftszeit des Fahrzeugs + Übergabezeit, dann wird das Fahrzeug zum Kundenwuschtermin übergeben.
Kategorie: Locating	
.LOCAGE Int	Anzahl Tage zwischen Bestellung eines Fahrzeugs und dem aktuellen Simulationstag. Zwischen Bestellung eines Fahrzeugs und dem aktuellen Simulationsdatum muss diese Anzahl Tage liegen, um das Fahrzeug für eine Überstellung in den Locating Pool in Frage kommen zu lassen.
.LOCATINGPOOL BASEDEALER 0..n →	Liste von Händlern, von denen Locatingfahrzeuge bezogen werden können.
.LOCATINGSEED Int	Seed Wert für Zufallsgenerator
.LOCDISTANCE Int	Räumliche Distanz, innerhalb derer der Händler andere Händler in seinen Locating Pool aufnimmt.
.LOCREADINESS Float	Die Bereitschaft (Wahrscheinlichkeit) eines Händlers Fahrzeuge bzw. Aufträge in den Locating Pool zu stellen. Kommt ein Fahrzeug in Frage in den Locating Pool gestellt zu werden, wird es mit dieser spezifizierten Wahrscheinlichkeit tatsächlich in den Locating Pool überstellt.
.LOCTIMEDISTANCE Int	Zeitlicher Abstand zwischen zwei Überprüfungen, welche Fahrzeuge in den Locating Pool gestellt werden. Ist dieser Wert z.B. 14, wird alle zwei Wochen für die Fahrzeuge im Händlerbestand geprüft, ob diese in den Locating Pool überstellt werden.

.LOCVEHICLENUMBER <i>Float</i>	Mindestbestand für Locating. Anzahl Fahrzeuge, die im Händlerbestand stehen müssen, bis Fahrzeuge in den Locating Pool gestellt werden. Die Anzahl Fahrzeuge wird als Anteil an der für den gegenwärtigen Monat prognostizierten Fahrzeuganzahl gemessen. Wurden in der Programmplanung für den aktuellen Monat in der Simulation z.B. 1.000 Fahrzeuge vorgesehen und das Attribut "Mindestbestand für Locating" ist mit 0,5 angegeben, werden erst dann Fahrzeuge in den Locating Pool abgegeben, wenn min. 500 Fahrzeuge im Auftrags- und Fahrzeugbestand des Händlers stehen.
-----------------------------------	--

Tabelle E.7.: BaseDealer

E.8. Klassenstruktur BASETYPETRANSFORMATION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Szenariendefinition	
.ATT_ID <i>Int</i>	ATT_ID

Tabelle E.8.: BaseTypeTransformation

E.9. Klassenstruktur BILLOFMATERIALRECORD

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.PARTS PARTREFERENCE 0..n →	Die Teile und ihre Gewichtungen, aus denen die Stückliste besteht.
.PATTERN PRODUCTPATTERN 0..1 →	Ein Suchmuster, das erfüllt sein muss, damit dieser Stücklisteneintrag angewendet wird.

Tabelle E.9.: BillOfMaterialRecord

E.10. Klassenstruktur BINOMIALDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.<N> <i>Float</i>	<n>
.<P> <i>Float</i>	<p>

Tabelle E.10.: BinomialDistribution

E.11. Klassenstruktur BOOLTRANSFORM

Basisklasse: BASETYPETRANSFORMATION (siehe Tabelle E.8)

Attribut / Typ	Beschreibung
Kategorie: Szenariendefinition	

.OLDVALUE <i>Bool</i>	oldValue
.VALUE <i>Bool</i>	Value

Tabelle E.11.: BoolTransform

E.12. Klassenstruktur BUFFER

Basisklasse: PROCESS (siehe Tabelle E.47)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISABLE <i>Bool</i>	Puffer ist deaktiviert.
Kategorie: Distribution	
.LOCATION DESTINATION 0..1 →	Ort
Kategorie: Kapazität	
.CAPACITY ASSOCDATCAPACITY 0..1 →	Pufferkapazität

Tabelle E.12.: Buffer

E.13. Klassenstruktur CONTAINER

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Abmessungen	
.EMPTYHEIGHT <i>Float</i>	Höhe (Leergut)
.EMPTYLENGTH <i>Float</i>	Länge (Leergut)
.EMPTYWIDTH <i>Float</i>	Breite (Leergut)
.HEIGHT <i>Float</i>	Höhe
.LENGTH <i>Float</i>	Länge
.WIDTH <i>Float</i>	Breite
Kategorie: Distribution	
.LOADINGDELAY PROBABILITYDISTRIBUTION 0..1 →	Zeit, einen Behälter dieses Typs auf ein Transportmittel zu laden.
.UNLOADINGDELAY PROBABILITYDISTRIBUTION 0..1 →	Zeit, einen Behälter dieses Typs von einem Transportmittel zu entladen.
Kategorie: Gewicht	
.MAXWEIGHT <i>Float</i>	Max. Gewicht
Kategorie: Kapazität	
.PARTCAPACITY <i>Int</i>	Kapazität Teile; Alternativ zur Angabe des Volumens.

Tabelle E.13.: Container

E.14. Klassenstruktur COSTSCOMMON

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Kosten	
.COSTADMIN ASSOCDATCOST 0..1 →	Verwaltungskosten. Bezugszeitraum: Monat
.COSTBASE ASSOCDATCOST 0..1 →	Basiskosten. Bezugszeitraum: Monat. Die Kosten fallen unabhängig von den Betriebsstunden des Objektes an.
.COSTDISTRIBUTION ASSOCDATCOST 0..1 →	Distributionskosten. Bezugszeitraum: Monat
.COSTINVEST ASSOCDATCOST 0..1 →	Investitionskosten. Bezugszeitraum: Monat
.COSTMATERIAL ASSOCDATCOST 0..1 →	Materialkosten. Bezugszeitraum: Monat
.COSTPRODUCTION ASSOCDATCOST 0..1 →	Produktionskosten. Bezugszeitraum: Betriebsstunden des Objektes. Enthalten sind auch Löhne/Gehälter.
.COSTSALES ASSOCDATCOST 0..1 →	Vertriebs- und Marketingkosten. Bezugszeitraum: Monat

Tabelle E.14.: CostsCommon

E.15. Klassenstruktur COSTSOBJECT

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Kosten	
.COSTDISTRIBUTION ASSOCDATCOST 0..1 →	Distributionskosten. Bezug: Pro Stück
.COSTMATERIAL ASSOCDATCOST 0..1 →	Materialkosten. Bezug: Pro Stück
.COSTPRODUCTION ASSOCDATCOST 0..1 →	Produktionskosten. Bezug: Pro Stück
.COSTSALES ASSOCDATCOST 0..1 →	Marketing- und Vertriebskosten. Bezug: Pro Stück
.REVENUE ASSOCDATCOST 0..1 →	Ertrag. Bezug: Pro Stück

Tabelle E.15.: CostsObject

E.16. Klassenstruktur CUSTOMERTYPE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Kosten	
.DELIVERYRELIABILITYFACTOR ASSOCDATKPIASSESSMENT 0..1 →	Bewertung Liefertreue
.LEADTIMEFACTOR ASSOCDATKPIASSESSMENT 0..1 →	Bewertung Lieferzeit
.TIMEINSTOCKFACTOR ASSOCDATKPIASSESSMENT 0..1 →	Bewertung Standzeit im Lager

Kategorie: Kundenverhalten	
.CUSTOMERORDERBANDWIDTH PROBABILITYDISTRIBUTION 0..1 →	Verteilung der Kundenwünsche
.DEFAULTDEVIATION <i>Float</i>	Wenn keine Abweichung durch den Parameter "Akzeptanz Abweichung" spezifiziert wurde, wird dieser Wert herangezogen.
.DELIVERYTIMEACCEPTANCE ASSOCDATDELIVERYTIMEACCEPTANCE 0..1 →	Stellt den Zusammenhang zwischen Lieferzeit und verlangter Übereinstimmung des Fahrzeugs mit dem Kundenwunsch dar.
.DELIVERYTIMELIMITED <i>Bool</i>	Wenn dieser Parameter aktiviert ist, springen Kunden ab, wenn die ihnen zugesagte (Plan-) Lieferzeit größer ist als die maximale Lieferzeit im Parameter "Akzeptanz Lieferzeit". Kunden, die abgesprungen sind, erkennt man in der Datenbank daran, dass im entsprechenden Eintrag in der Tabelle OUT_Customer_Order im Feld CapacityRestrictionN (N=1,...,4) der Vermerk LostSales eingetragen ist.
.DEVIATIONACCEPTANCE PROPERTYDEV 0..n →	Gibt die Abweichung des Auftrags vom Kundenwunsch an, wenn Eigenschaft X durch Eigenschaft Y ersetzt ist.
.SHARE <i>Float</i>	Anteil an den Kunden eines Händlers. Der Anteil muss Null sein (bzw. nicht spezifiziert), wenn der Kundentyp zur Abnahme von Lagerfahrzeugen vorgesehen ist.
.STOCKTIMEACCEPTANCE ASSOCDATDELIVERYTIMEACCEPTANCE 0..1 →	Modelliert die Wahrscheinlichkeit, mit der ein Kunde bereit ist, ein Lagerfahrzeug abzunehmen. Die Intervalle definieren die Zeitspanne in Tagen, seit der sich ein Fahrzeug im Verkaufsbestand befindet (d.h. ab der Ankunft beim Händler bzw. der Ankunft an einem Punkt in der Distribution, ab dem das Fahrzeug nur noch per Kundenauftrag weiter transportiert wird). Die Wertreihe muss Werte zwischen 0 und 1 enthalten. Diese Werte geben die Wahrscheinlichkeit an, mit der ein Kunde ein Lagerfahrzeug des jeweiligen "Alters" akzeptiert. Rabatte für solche Lagerfahrzeuge können über das Attribut "Bewertung Standzeit im Lager" definiert werden.
.TIMEOFFSET <i>Int</i>	Das Intervall "Akzeptanz Lieferzeit" wird um die spezifizierte Anzahl Minuten verschoben. Standzeiten eines Fahrzeugs beim Händler können so zu einer höheren Akzeptanz beim Kunden führen (Preisgestaltung).

Tabelle E.16.: CustomerType

E.17. Klassenstruktur DAILYRCT

Basisklasse: RECURRENTTASK (siehe Tabelle E.61)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISTANCE <i>Int</i>	Ereignis tritt jeweils alle n-Tage auf. Ein Wert von "8" führt zu einem Ereignis alle acht Tage.
.OFFSET <i>Int</i>	Ereignis tritt jeweils in der n-ten Stunde eines Tages ein. Ein Wert von 12 führt zu einem Ereignis um 12.00 an jedem Tag .

Tabelle E.17.: DailyRCT

E.18. Klassenstruktur DEALER

Basisklasse: BASEDEALER (siehe Tabelle E.7)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.18.: Dealer

E.19. Klassenstruktur DEALERPLANNING

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
----------------	--------------

Kategorie: Einplanung	
.AGREEMENTDEVIATION <i>Float</i>	Abweichung Volumenvereinbarung
.COPULLOUT <i>Int</i>	Anzahl Tage, die ein Fahrzeug vor dem Wunschtermin ankommen darf.
.FIRSTFORECASTDEVIATION <i>Float</i>	Abweichung 1. Prognose
.HOLDVA <i>Bool</i>	Der Händler reduziert sein Commitment für den nächsten Monat nicht, wenn für einzelne Kunden Direktbestellungen durchgeführt werden müssten.
.MODIFYORDERS <i>Bool</i>	Auftrag wird so verändert, dass er am gewünschten Termin gefertigt werden kann.
.ORDERS POINTSINTIME 0..1 →	Wann machen die Händler ihre Bestellungen?
.OVERRIDEVA <i>Bool</i>	Das Volumencomittment des Händlers darf überschritten werden, wenn Kapazität in den Werken und bei den Zulieferern vorhanden ist.
.PREFETCHORDERS <i>Bool</i>	Aufträge aus dem Auftragspool vor dem Werk (vor FU) heranziehen, um Tagesprogramme im Fall von Störungen aufzufüllen.
.SECONDFORECASTDEVIATION <i>Float</i>	Abweichung 2. Prognose
.TRASHORDERS <i>Bool</i>	Gibt an, ob die Bestellung gelöscht wird, wenn der Bestelltermin nicht exakt eingehalten werden kann!
Kategorie: Einplanungsfunktion Werk	
.BACKLOGFIDELITY <i>Float</i>	Gewichtung Termintreue (Rückstand) in der Formel für die Werksverbuchung
.ENDPLANINTERVAL <i>Int</i>	Ende Planintervall
.FETCHAHEADFIDELITY <i>Float</i>	Gewichtung Termintreue (Vorziehen) in der Formel für die Werksverbuchung
.STARTPLANINTERVAL <i>Int</i>	Beginn Planintervall
.WEIGHTPRIO <i>Float</i>	Gewichtung der Auslastung in der Formel für die Werksverbuchung
.WEIGHTPRIOPOST <i>Float</i>	Auslastungsfaktor Rückstände in der Formel für die Werksverbuchung
.WEIGHTPRIOPRE <i>Float</i>	Auslastungsfaktor Vorgriffe in der Formel für die Werksverbuchung
Kategorie: Kundenverhalten	
.CUSTOMERTYPES CUSTOMERTYPE 0..n →	Liste aller Kundentypen dieses Marktes
Kategorie: Verwirbelungen	
.DELAYPLANNING PROBABILITYDISTRIBUTION 0..1 →	Verwirbelung Aufträge (Planungsphase)
.DELAYPLANT PROBABILITYDISTRIBUTION 0..1 →	Verwirbelung Aufträge (Werk)
Kategorie: Zeit	
.DIRECTORDERSONLY <i>Bool</i>	Der Händler führt keine Auffüllbestellungen durch.
.DISTANCEO <i>Int</i>	Für wann machen die Händler ihre Bestellungen?
.DISTANCEVA <i>Int</i>	Für wann werden die Volumenvereinbarungen gemacht?
.LOOKUPDISTANCE <i>Int</i>	Länge des Intervalls, das der Händler abwechselnd in seinen Bestellungen und im Locating durchsucht, um ein passendes Fahrzeug für den Kunden zu finden.
.VOLUMEAGREEMENT POINTSINTIME 0..1 →	Wann werden die Volumenvereinbarungen gemacht?

Tabelle E.19.: DealerPlanning

E.20. Klassenstruktur DISTRIBUTION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.AVGSPEEDBACKROAD <i>Float</i>	Mittlere Geschwindigkeit Nebenstrasse
.AVGSPEEDEXPRESSWAY <i>Float</i>	Mittlere Geschwindigkeit auf der Autobahn
.AVGSPEEDHIGHWAY <i>Float</i>	Mittlere Geschwindigkeit Schnellstrasse
.AVGSPEEDMAINROAD <i>Float</i>	Mittlere Geschwindigkeit Bundesstrasse
.AVGSPEEDROAD <i>Float</i>	Mittlere Geschwindigkeit Landstrasse
.DISTANCEMATRIX ROUTEINFORMATION 0..n →	Informationen über Entfernungen zwischen zwei Punkten. Sind diese Informationen für zwei Punkte nicht verfügbar, wird die Entfernung als Luftlinie bestimmt .
.FUELCONSUMPTIONCITY <i>Float</i>	Treibstoffverbrauch Stadt
.FUELCONSUMPTIONEXPRESSWAY <i>Float</i>	Treibstoffverbrauch Autobahn
.FUELCOST <i>Float</i>	Kraftstoffkosten
.LOGISTICSERVICEPROVIDER LOGISTICSERVICEPROVIDER 0..n →	Logistikdienstleister
.MAXHOPS <i>Int</i>	Maximale Anzahl Distributionskanäle bzw. Puffer, durch die ein Fahrzeug geleitet werden darf. Wird dieser Wert überschritten, erzeugt der Simulator eine Fehlermeldung. Dient der Erkennung von Zyklen in Simulationsmodellen.
.ROUTES DISTRIBUTIONCHANNEL 0..n →	Wege, auf denen Transportmittel verkehren darf. Wenn nicht gesetzt, können alle Distributionskanäle genutzt werden.
.TOLLEXPRESSWAY <i>Float</i>	Maut Autobahn

Tabelle E.20.: Distribution

E.21. Klassenstruktur DISTRIBUTIONCHANNEL

Basisklasse: PROCESS (siehe Tabelle E.47)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISABLE <i>Bool</i>	Distributionskanal ist nicht aktiv. Fahrzeuge werden über alternative Wege geroutet.
Kategorie: Bedingungen	

.SORTCONDITION <i>String</i>	<p>Wenn dieses Attribut gesetzt ist, werden die Aufträge bzw. Produkte im Eingangspuffer des Distributionskanals sortiert. Das Attribut selbst ist ein Skript, welches die Sortierbedingung definiert. Syntax: if(<Bedingung> ; <dann> ; <sonst>); Das Skript wird für jedes Paar von Aufträgen aufgerufen und muss eine 1 oder eine 0 zurückgeben. 1 = Auftrag [0] ist "größer" 0 = Auftrag [1] ist "größer" Auf die Eigenschaften der zu prüfenden Aufträge kann über folgende Notation zugegriffen werden.</p> <pre>@[0].<Name der Eigenschaft> // Eigenschaft von Auftrag[0] @[1].<Name der Eigenschaft> // Eigenschaft von Auftrag[1]</pre> <p>Beispiel: Sortierung der Aufträge nach dem geplanten Produktionsende</p> <pre>if (@[0].PlannedEOP < @[1].PlannedEOP ; 0; 1);</pre> <p>Beispiel2:</p> <pre>prod1Day = @[0].Production_Start_Day + @[0].Delay / 1440; prod2Day = @[1].Production_Start_Day + @[1].Delay / 1440; hasCustomer1 = @[0].HasCustomer; hasCustomer2 = @[1].HasCustomer; repl1 = @[0].Replanned; repl2 = @[1].Replanned;</pre> <pre>if (prod1Day < prod2Day ; 1 ; if (prod1Day > prod2Day ; 0 ; if (repl1 > repl2 ; 1 ; if (hasCustomer1 and not hasCustomer2 ; 1 ; if (hasCustomer2 and not hasCustomer1 ; 0 ; if (@[0].SerialNo < @[1].SerialNo ; 1 ; 0)))))) ;</pre>
Kategorie: Distribution	
.ENDBUFFER BUFFER 0..1 →	Abladestelle für transportierte Güter
.STARTBUFFER BUFFER 0..1 →	Übernahmestelle für zu transportierende Güter
.SUBCHANNELS PROCESS 0..n →	Unterdistributionskanäle [as1]
Kategorie: Geographische Informationen	
.DESTINATION DESTINATION 0..1 →	Zielpunkt
.SHIPPINGAREA <i>String</i>	Die Versandzone des Zielpunkts
.SOURCE DESTINATION 0..1 →	Ausgangspunkt
Kategorie: Kapazität	
.CAPACITY ASSOCDATCAPACITY 0..1 →	Maximale Transportkapazität. Sollte min. dem Produkt aus <<Kapazität einer Transporteinheit>> und <<Min. Transporteinheiten>> entsprechen.
Kategorie: Kosten	
.COSTSTRANSPORT COSTSOBJECT 0..1 →	Einzelkosten, die jeweils für einen Transport anfallen, unabhängig von der Anzahl der transportierten Fahrzeuge.
Kategorie: Transport auf Abruf	
.MINUNITS <i>Int</i>	Anzahl Einheiten (z.B. Waggons), aus denen ein Transport min. bestehen muss, um gestartet werden zu können.
.ONREQUEST <i>Bool</i>	Transport auf Abruf? Wenn der Parameter nicht gesetzt ist, muss ein Fahrplan definiert sein!
.REQUESTDELAY <i>Int</i>	Minimale Zeitspanne zwischen der Anforderung eines Transports und dem Transportbeginn
.REQUESTMINSTOCK <i>Int</i>	Lagerbestand, der min. vorhanden sein muss, um eine Transportanforderung auszulösen.

.UNITSIZE <i>Int</i>	Anzahl Fahrzeuge, die auf einer Transporteinheit bewegt werden können. Die Transporteinheiten werden zu Transporten zusammengestellt. Bsp. <<Kapazität einer Transporteinheit>> (Waggon) ist acht, 20 Waggons (Parameter <<Min. Transporteinheiten>>) werden zu einem Zug zusammengestellt.
Kategorie: Transportbedingungen	
.WAITPATTERN PRODUCTPATTERN 0..n →	Solange diese Bedingung erfüllt ist, bleibt ein Fahrzeug im Transportpuffer des Distributionskanals stehen. Ersetzt bzw. erweitert das Attribut "Transport mit Kundenauftrag", welches früher an der Transportrelation existierte.
Kategorie: Zeit	
.TIMETABLE POINTSINTIME 0..1 →	Fahrplan
.TIMETOTRANSPORT PROBABILITYDISTRIBUTION 0..1 →	Eine Wahrscheinlichkeitsverteilung, die die Verweildauer im Puffer angibt.
.TRANSPORTTIME PROBABILITYDISTRIBUTION 0..1 →	Fahrzeit

Tabelle E.21.: DistributionChannel

E.22. Klassenstruktur DT_AssocDAT

Basisklasse: ASS_DTA (siehe Tabelle E.2)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.22.: DT_AssocDat

E.23. Klassenstruktur EXPERIMENT

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Datenbankausgaben	
.ENDREPORT <i>DateTime</i>	Ende Datenbankausgaben
.STARTREPORT <i>DateTime</i>	Start Datenbankausgaben
Kategorie: Distribution	
.DISTRIBUTION DISTRIBUTION 0..1 →	Distribution
Kategorie: Einplanung	
.PLANNING PLANNING 0..1 →	Globale Planungsparameter
Kategorie: Fahrzeugabsätze und Einbauraten	
.SALES SALES 0..1 →	Die globalen Fahrzeugabsätze für den Simulationlauf
Kategorie: Fertigung	
.PLANTS PLANT 0..n →	Liste aller Werke des Experiments
.SUPPLIERS SUPPLIER 0..n →	Liste aller Zulieferer des Experiments
Kategorie: Händler / Kundenbestellungen	
.MARKETS MARKET 0..n →	Liste aller Märkte des Experiments
Kategorie: Kosten	
.COSTS COSTSCOMMON 0..1 →	Gemeinkosten Konzern

Kategorie: Produktbeschreibung	
.PRODUCTSTRUCTURE PRODUCTSTRUCTURE 0..1 →	Die modellierten Eigenschaften, Familien und Fahrzeugbeschreibungen
Kategorie: Zeit	
.SIMEND DateTime	Ende der Simulation. Die Simulation sollte noch 3 Monate nach dem Ende des Absatzverlaufs fortgeführt werden.
.SIMSTART DateTime	Beginn der Simulation. Der Simulationsstart sollte 3 Monate vor Beginn des Absatzverlaufs liegen.
Kategorie: Zufallszahlen	
.DISTRIBUTIONSEED Int	Seed für Zufallszahlen in der Distribution
.GLOBALSEED Int	Ist dieser Wert gesetzt, wird der Seed-Wert für alle Zufallszahlen in dem Simulationsmodell festgelegt. Der globale Seed-Wert überschreibt dann alle anderen Seed-Werte.
.PRODUCTGENERATORSEED Int	Seed-Erzeugung Produktspezifikationen

Tabelle E.23.: Experiment

E.24. Klassenstruktur EXPONENTIALDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.LAMBDA Float	Lambda (>0)

Tabelle E.24.: ExponentialDistribution

E.25. Klassenstruktur FAILURERULE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Störungen	
.FILLUP Int	Gibt an, ob die Ausbringung im Störfall durch Vorziehen (um max. n-Tage) von Aufträgen gehalten (wenn möglich) oder gesenkt wird.
.MINLOAD Float	Die minimale Auslastung des Werkes an einem Tag, die zur Aktivierung der Regel führt. Ist der Parameter nicht gesetzt, ist die Regel immer aktiv.
.PATTERN PRODUCTPATTERN 0..1 →	Beschreibung der Fahrzeuge, die von dieser Störungsregel betroffen sind.

Tabelle E.25.: FailureRule

E.26. Klassenstruktur FAULT

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Störungen	
.ADVANCEWARNING Int	Zeit vor Beginn der Störung, zu dem diese Störung bereits bekannt ist.
.CAPACITYCHANGE Float	Ein Faktor, mit dem die ursprüngliche Kapazität multipliziert wird.
.DURATION Int	Dauer einer Störung

.OCCURRENCE POINTSINTIME 0..1 →	Zeitpunkte, an denen die Störung auftritt.
---------------------------------------	--

Tabelle E.26.: Fault

E.27. Klassenstruktur HISTOGRAMDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.HISTOGRAM ASSOCDATHISTOGRAM 0..1 →	Histogramm

Tabelle E.27.: HistogramDistribution

E.28. Klassenstruktur HOURLYRCT

Basisklasse: RECURRENTTASK (siehe Tabelle E.61)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISTANCE <i>Int</i>	Bezeichnet den Abstand der Ereignisse in Stunden. Ein Wert von 3 führt zu einem Ereignis alle drei Stunden.
.OFFSET <i>Int</i>	Bezeichnet die n-te Minute einer Stunde. Ein Wert von z.B. 3 führt zu einer Terminserie: 0.03, 1.03, 2.03 etc..

Tabelle E.28.: HourlyRCT

E.29. Klassenstruktur IMPORTER

Basisklasse: BASEDEALER (siehe Tabelle E.7)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.29.: Importer

E.30. Klassenstruktur LOCATION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Geographische Informationen	
.CITY <i>String</i>	Stadt
.LATITUDE <i>Float</i>	geographische Breite in Dezimaldarstellung
.LONGITUDE <i>Float</i>	geographische Länge in Dezimaldarstellung
.POSTCODE <i>Int</i>	Postleitzahl
.STREET <i>String</i>	Strasse und Hausnummer

Tabelle E.30.: Location

E.31. Klassenstruktur LOGISTICSERVICEPROVIDER

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Bewertung Planung	
.COSTUNUSEDCAPACITY <i>Float</i>	Kosten ungenutzter Kapazität
Kategorie: Distribution	
.LSPROUTES LSPROUTE 0..n →	Definition von Kunden bzw. Abholstellen
.TRANSPORTDEFINITION TRANSPORTDEFINITION 0..n →	Festlegung der Arbeitsparameter für die Transportmittel des Logistikdienstleisters
Kategorie: Geographische Informationen	
.LOCATION LOCATION 0..1 →	Ort

Tabelle E.31.: LogisticsServiceProvider

E.32. Klassenstruktur LSPROUTE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.BUFFER BUFFER 0..n →	Abholstellen
.TRANSPORTSCHEDULE POINTSINTIME 0..1 →	Fahrplan Transporte

Tabelle E.32.: LSPRoute

E.33. Klassenstruktur MARKET

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Fahrzeugabsätze und Einbauratzen	
.PRODUCTQUOTA PRODUCTQUOTA 0..n →	Fahrzeuge, die auf diesem Markt verkauft werden und Anteile hierfür.
Kategorie: Händler / Kundenbestellungen	
.DEALERLIST BASEDEALER 0..n →	Liste aller Händler des Marktes
Kategorie: Kosten	
.COSTS COSTSCOMMON 0..1 →	Gemeinkosten Markt
.INTEREST <i>Float</i>	Zinssatz auf diesem Markt
Kategorie: Sonderverkäufe	

.DESIREDSTOCKLEVEL PRODUCTSHARE 0..n →	Wird der gewünschte Lagerbestand überschritten, können Produkte durch hierfür konfigurierte Kundentypen (kein "Anteil") abgenommen werden. Die Bestandshöhen können spezifisch für einzelne Fahrzeugbeschr. parametrisiert werden.
.STOCKMAXTIME Int	Wenn sich ein Fahrzeug länger als die in diesem Attribut definierte Zeit im Verkaufsbestand befindet, versucht der Händler es über einen Sonderverkauf an den Kunden zu bringen. Diese Sonderverkäufe werden durch einen speziellen Kundentyp am Markt abgenommen, der dadurch definiert ist, dass kein Anteil definiert ist (analog zum Attribut "Gewünschter Lagerbestand").

Tabelle E.33.: Market

E.34. Klassenstruktur MONTHLYRCT

Basisklasse: RECURRENTTASK (siehe Tabelle E.61)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISTANCE Int	Bezeichnet den Abstand der Ereignisse in Monaten. Ein Wert von "5" führt zu einem Ereignis alle fünf Monate.
.OFFSET Int	Ein Ereignis tritt an jedem n-ten Tag im Monat ein. Ein Wert von "7" führt zu einem Ereignis am 7.1., am 7.2., am 7.3. usw..

Tabelle E.34.: MonthlyRCT

E.35. Klassenstruktur MONTHLYTIMESPAN

Basisklasse: TIMESPAN (siehe Tabelle E.78)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.35.: MonthlyTimespan

E.36. Klassenstruktur NORMALDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.MEAN Float	Erwartungswert
.VARIANCE Float	Varianz

Tabelle E.36.: NormalDistribution

E.37. Klassenstruktur OBJECTQUOTA

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Händler / Kundenbestellungen	
.CUSTOMERORDERS ASSOCIATINSTRATE 0..1 →	Kundenbestellungen

.DEALERORDERS ASSOCDATINSTRATE 0..1 →	Händlerbestellungen
Kategorie: Prognosen	
.FORECAST1 ASSOCDATINSTRATE 0..1 →	1. Prognose des Handels
.FORECAST2 ASSOCDATINSTRATE 0..1 →	2. Prognose des Handels
.LONGTERMFORECAST ASSOCDATINSTRATE 0..1 →	Langfristiger Absatzplan
.PROGRAMMEPLANNING ASSOCDATINSTRATE 0..1 →	Programmplanung

Tabelle E.37.: ObjectQuota

E.38. Klassenstruktur ORDERMODIFIER

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.PATTERN PRODUCTPATTERN 0..1 →	Suchmuster
.PROPERTY PROPERTY 0..n →	Eigenschaft, die im Auftrag gesetzt wird, wenn Suchmuster "passt".

Tabelle E.38.: OrderModifier

E.39. Klassenstruktur PART

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.PARTVOLUME <i>Float</i>	Volumen eines Teils, das vom Zulieferer gefertigt wird.
.PARTWEIGHT <i>Float</i>	Gewicht Teil
Kategorie: Produktbeschreibung	
.ITEMNUMBER <i>String</i>	Artikelnummer
.PARTSUPPLIER SUPPLIER 0..1 →	Zulieferer

Tabelle E.39.: Part

E.40. Klassenstruktur PARTREFERENCE

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.PART PART 0..1 →	Teil

.WEIGHT <i>Float</i>	Gewichtung
-------------------------	------------

Tabelle E.40.: PartReference

E.41. Klassenstruktur PLANNING

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Einplanung	
.HANDOVERDATES POINTSINTIME 0..1 →	Terminserie mit Terminen, an denen das Einfrieren der Aufträge vorgenommen wird.
.LOADDIFFPRIO <i>Float</i>	Priorität gleichmäßige Auslastung Werke in der Formel für die Werksverbuchung
.MANUFACTHANDOVER <i>Int</i>	Anzahl Tage (Arbeits-/Wochentage) vor Fertigungsende, an dem die Fixierung der Aufträge (Einfrieren) stattfindet.
.MANUFACTHANDOVERTYPE <i>Bool</i>	Angabe des Zeitpunkts zum Fixieren (Einfrieren) der Aufträge in Kalendertagen (Ja) oder Arbeitstagen (Nein) vor Fertigungsende.
.PLANTASSIGNMENT <i>Int</i>	Anzahl Tage (Arbeits-/Wochentage) vor Fertigungsende, an dem die Werkszuordnung stattfindet.
.PLANTASSIGNMENTDATES POINTSINTIME 0..1 →	Terminserie mit Terminen, an denen die Werkszuordnung vorgenommen wird.
.PLANTASSIGNMENTTYPE <i>Bool</i>	Angabe des Zeitpunkts der Werkszuordnung in Kalendertagen (Ja) oder Arbeitstagen (Nein) vor Fertigungsende.
.PREFETCHORDERS <i>Bool</i>	Aufträge aus dem Auftragspool vor dem Werk heranziehen, um Tagesprogramme im Fall von Störungen aufzufüllen.
.RANDOMIZEDEALERORDERS <i>Bool</i>	Reihenfolge Händlerbestellungen zufällig
Kategorie: Prognosen	
.DISTANCEF1 <i>Int</i>	Distanz 1. Prognose
.DISTANCEF2 <i>Int</i>	Distanz 2. Prognose
.FIRSTFORECAST POINTSINTIME 0..1 →	Termine 1. Prognose
.SECONDFORECAST POINTSINTIME 0..1 →	Termine 2. Prognose
Kategorie: Sequenzbildung	
.ORDERFUNCTIONPLANNING <i>String</i>	Skript, in dem eine Funktion definiert werden kann, die zur Sortierung des Auftragspools vor dem Werk dient. Diese Funktion muss zwingend eine Totalordnung auf den Aufträgen implizieren.

Tabelle E.41.: Planning

E.42. Klassenstruktur PLANT

Basisklasse: PROCESS (siehe Tabelle E.47)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.DISTRIBUTIONSTART PROCESS 0..1 →	Startpunkt Distribution
.INBUFFER BUFFER 0..1 →	Der Wareneingangspuffer muss definiert werden, wenn ein Inbound-Distributionssystem modelliert wurde. Vom Zulieferer befüllte Behälter werden in diesem Puffer vom Logistikkdienstleister abgestellt.
Kategorie: Fertigung	

E.43. KLASSENSTRUKTUR PLANTPROPERTYSPECINFORMATIONIN286

.FILLUP <i>Bool</i>	Ja: fehlende Teile werden im Wareneingangspuffer ergänzt.
.LABORTIME <i>Bool</i>	Aus Durchlaufzeitverteilung gezogene Durchlaufzeit eines Auftrags wird als Arbeitszeit interpretiert.
.PROPERTIES PROPERTY 0..n →	Liste aller Eigenschaften, die nicht im Werk verbaut werden.
.SUBPLANTS PLANT 0..n →	Liste der Fertigungsabschnitte
Kategorie: Geographische Informationen	
.DESTINATION DESTINATION 0..1 →	Zielbahnhof des Werks (WA - Lager)
Kategorie: Kapazität	
.CAPACITYACTUAL PRODUCTIONUNITRESSOURCE 0..1 →	Ist-Kapazität des Werkes
.CAPACITYPLANNED PRODUCTIONUNITRESSOURCE 0..1 →	Die für die Planungsaktivitäten genutzte Kapazität des Werkes. Ist dieser Parameter nicht gesetzt, wird die parametrisierte Ist-Ausbringung genutzt.
.CAPLIMITPP ASSOCDATHISTOGRAM 0..1 →	Das Werk fertigt max. die im Produktionsprogramm festgelegte Anzahl Aufträge multipliziert mit diesem Faktor und nutzt maximal jedoch die zur Verfügung stehende Kapazität.
.CAPLIMITTC ASSOCDATHISTOGRAM 0..1 →	Anteil der technischen Kapazität, der für Umplanungen bzw. Reservierungen im Fall von Engpässen genutzt werden kann.
Kategorie: Produktbeschreibung	
.BILLOFMATERIALPLANT BILLOFMATERIALRECORD 0..n →	Eine Liste mit allen Stücklisteninträgen für das Werk
.PRODUCTSHARES PRODUCTSHARE 0..n →	Alle Fahrzeugbeschreibungen, die in diesem Werk gebaut werden, mit ihrem Anteil am Gesamtaufkommen.
Kategorie: Sequenzbildung	
.ORDFUNCPLTASSIGN <i>String</i>	Skript, in dem eine Funktion definiert werden kann, die zur Sortierung des Auftragspools vor dem Werk dient. Diese Funktion muss zwingend eine Totalordnung auf den Aufträgen implizieren.
Kategorie: Sourcing	
.SUPPLIERS PLANTPROPERTYSPECINFORMATIONIN 0..n →	Priorisierte Referenzen auf alle Zulieferer, die dieses Werk beliefern. Existieren für eine Eigenschaft mehrere alternative Zulieferer, wird die Auswahl über die Priorisierung getroffen.
Kategorie: Störungen	
.FAILURESTRATEGIES FAILURERULE 0..n →	Steuerungsregeln, die angewandt werden, wenn eine Störung auftritt.
.RESCHEDULEORDERS <i>Bool</i>	Betroffene Aufträge werden nach einer Kapazitätsveränderung neu eingeplant. Ein neuer Zeitpunkt für den Produktionsstart wird bestimmt.
.RESERVEDISTANCE <i>Int</i>	Distanz Tage vor Fertigungsbeginn. Frühestens hinter diesem Zeitpunkt kann Kapazität reserviert werden.
Kategorie: Zeit	
.DELAYAVG <i>Int</i>	Mittlere Durchlaufzeit. Wird genutzt zur Rückwärtsterminierung, wenn kein aktueller Auftrag verfügbar ist.

Tabelle E.42.: Plant

E.43. Klassenstruktur PLANTPROPERTYSPECINFORMATIONIN

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Sourcing	
.PROPERTYSPEC PROPERTYSPEC 0..1 →	Eigenschaftenspezifikation
.SUPPLIERPRIORITISATION SUPPLIERPRIORITISATION 0..n →	Zuliefererpriorisierung

Tabelle E.43.: PlantPropertySpecInformationIn

E.44. Klassenstruktur POINTSINTIME

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DATESEQUENCE VECTOREXPLICITDATES 0..1 →	Liste mit expliziten Ereigniszeitpunkten
.EXCEPTIONS VECTOREXPLICITDATES 0..1 →	Zeitpunkte, die aus der Liste entfernt werden.
.RECURRENTTASKS RECURRENTTASK 0..n →	Liste mit Terminserien, die Ereigniszeitpunkte generieren.

Tabelle E.44.: PointsInTime

E.45. Klassenstruktur POISSONDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.LAMBDA <i>Float</i>	Lambda

Tabelle E.45.: PoissonDistribution

E.46. Klassenstruktur PROBABILITYDISTRIBUTION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.LOWERBOUND <i>Float</i>	Untere Grenze für Werte aus der Verteilungsfunktion. Liegt ein gezogener Wert darunter, wird ein neuer Wert gezogen.
.MAXDRAW <i>Int</i>	Liegt der gezogene Wert ausserhalb der unteren und oberen Grenzen, wird solange gezogen, bis der Wert innerhalb liegt oder die Anzahl Versuche überschritten wurde. Ansonsten wird der Planwert zurückgegeben.
.OFFSET ASSOCDATDISTRIBUTIONOFFSET 0..1 →	Versatz der Verteilung über die Zeit
.PLANVALUE <i>Float</i>	Planwert wird bei der Berechnung von voraussichtlichen Durchlaufzeiten genutzt.
.SCALE ASSOCDATDISTRIBUTIONSCALE 0..1 →	Skalierung der Verteilung über die Zeit
.UPPERBOUND <i>Float</i>	Untere Schranke des Intervalls, aus dem Werte gezogen werden.
Kategorie: Bedingungen	
.PRODUCTPATTERN PRODUCTPATTERN 0..1 →	Bedingungen, die erfüllt sein müssen, um die Wahrscheinlichkeitsverteilung auszuwählen bzw. zu aktivieren.
Kategorie: Zufallszahlen	
.OFFSETPROBDISTRIBUTION PROBABILITYDISTRIBUTION 0..1 →	Eine weitere Wahrscheinlichkeitsverteilung. Der aus dieser Verteilung gezogene Wert wird addiert. Zirkuläre Strukturen sind nicht zulässig.

.SEED <i>Int</i>	Startwert für die Generierung der Zufallszahlen
---------------------	---

Tabelle E.46.: ProbabilityDistribution

E.47. Klassenstruktur PROCESS

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Datenbankausgaben	
.WRITEPROCESSSTATS <i>Bool</i>	Prozessstatistiken in die Datenbank schreiben. Wenn für einen Prozess die Statistiken nicht benötigt werden, kann die Deaktivierung der Statistiken die Laufzeit der Simulation deutlich verbessern.
Kategorie: Distribution	
.ANNOUNCEETA <i>Bool</i>	Melde erwartete Ankunftszeit
.OUT PROCESS 0..1 →	Nachfolgender Prozess eines Prozesses
.ROUTINGTABLE ROUTINGTABLE 0..1 →	Enthält Liste alternativer Routen, auf denen ein Fahrzeug bewegt werden kann.
Kategorie: Kosten	
.COSTSCOMMON COSTSCOMMON 0..1 →	Gemeinkosten
.COSTSOBJECT COSTSOBJECT 0..1 →	Einzelkosten
.COSTSOBJECTDELAY COSTSOBJECT 0..1 →	Kosten Verweildauer im Objekt. Für jeden Tag bzw. Teil eines Tages wird ein entsprechender Kostensatz addiert.
Kategorie: Zeit	
.DELAY PROBABILITYDISTRIBUTION 0..n →	Durchlaufzeit bzw. Wartezeit vor Verlassen des Objektes

Tabelle E.47.: Process

E.48. Klassenstruktur PRODUCTCLASS

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Datenimport	
.ALIASNAMES <i>String</i>	Namen von Fahrzeugklassen, Aktionsfahrzeugen o.ä., die beim Datenimport dieser Fahrzeugbeschreibung zugeordnet werden.
Kategorie: Einplanung	
.WEIGHT <i>Float</i>	Gewichtung der Fahrzeugbeschreibung im Tagesprogramm
Kategorie: Fahrzeugabsätze und Einbauraten	
.ACTUALRATE ASSOCDATINSTRATE 0..1 →	Der anteilige Absatz bezieht sich immer auf die übergeordnete Fahrzeugbeschreibung, bzw. in der Wurzel auf die globalen Absätze.
.EXPECTEDRATE ASSOCDATINSTRATE 0..1 →	Der anteilige Absatz bezieht sich immer auf die übergeordnete Fahrzeugbeschreibung, bzw. in der Wurzel auf die globalen Absätze.
.LONGTERMEST ASSOCDATINSTRATE 0..1 →	Langfristiger Absatzplan
Kategorie: Kapazität	
.CHANGEINTO PRODUCTCLASS 0..n →	Produktklassen, in die Auftrag im Rahmen der Bestellveränderung geändert werden kann. Umfasst auch abgeleitete Beschreibungen.

.RESERVECAPACITY <i>Bool</i>	Kapazität auf Basis LAP reservieren. Für Fahrzeugbeschreibungen, die für Auffüllbestellungen verwendet werden, auf 0 setzen.
Kategorie: Kosten	
.COSTS COSTSOBJECT 0..1 →	Einzelkosten für Fahrzeug
Kategorie: Kundenverhalten	
.EQUIVALENCE PRODUCTCLASS 0..n →	Fahrzeugklassen im Modell, die das gleiche Produkt modellieren und von einem Kunden auch akzeptiert werden.
Kategorie: Produktbeschreibung	
.CLASSPROPERTIES PROPERTYSPEC 0..n →	Eigenschaften mit Einbaurationen für diese Fahrzeugbeschreibung
.DEFAULTPROPERTIES PROPERTY 0..n →	Standardsetzungen dieser Fahrzeugbeschreibung (z.B. Links- oder Rechtslenker für ein Länderkennzeichen)
.DERIVEDCLASSES PRODUCTCLASS 0..n →	Untergeordnete Fahrzeugbeschreibungen sind z.B. Aktionsfahrzeuge unterhalb einer Fahrzeugklasse.
.ENFORCEDPROPERTIES PROPERTY 0..n →	Eigenschaften, die durch die Fahrzeugbeschreibung gezwungen werden. Diese Eigenschaften sind in jedem Auftrag dieser Fahrzeugbeschreibung enthalten und sind nicht durch Regeln oder Einbaurationen überschreibbar. Beispiel: Die Fahrzeugbeschreibung "VW Golf 1,6L 3 Türer" legt die Eigenschaft Motor eindeutig auf den 1,6l Motor fest. Kein anderer Motor kann in dieser Fahrzeugbeschreibung eingebaut werden. Spezifikationen anderer Motoren mit Einbaurationen sind an dieser Fahrzeugbeschreibung ungültig. Regeln, die andere Motoren hineinzwängen, sind ungültig.
.ISABSTRACT <i>Bool</i>	Abstrakte Fahrzeugbeschreibungen werden in der Simulation nicht gebaut. Stattdessen werden die Einbaurationen der untergeordneten Fahrzeugbeschreibungen auf 100 % normiert (wodurch nur diese gebaut werden).
.PROPERTYCONSTRAINTS PROPERTYCONSTRAINTBASE 0..n →	Gruppen von Eigenschaften mit gemeinsamen Einbaurationen
Kategorie: Visualisierung	
.IMAGE <i>OLEObject</i>	Bild des Produkts als *.bmp, *.gif, oder *.jpg Datei. Wird z.B. zur Visualisierung im OTD-Analyser verwendet.

Tabelle E.48.: ProductClass

E.49. Klassenstruktur PRODUCTIONUNITRESOURCE

Basisklasse: RESOURCE (siehe Tabelle E.62)

Attribut / Typ	Beschreibung
Kategorie: Kapazität	
.CAPACITYDEVIATION ASSOCDATCAPACITY 0..1 →	Abweichung von der spezifizierten Kapazität als Faktor
.ISINTEGER <i>Bool</i>	Gibt an, ob für die Kapazität nur ganze Zahlen zulässig sind.
.OUTPUT ASSOCDATCAPACITY 0..1 →	Anzahl Fahrzeuge pro Stunde, die gefertigt werden können. Multipliziert mit der Arbeitszeit an einem Tag ergibt sich die tatsächliche Kapazität (an diesem Tag).
.SIMPLERESOURCE <i>Bool</i>	Ressource stellt Kapazität stets vollständig zur Verfügung.
.WORKINGHOURS WEEKTIMERESSOURCE 0..1 →	Wochenressource
Kategorie: Störungen	
.FAULTS FAULT 0..n →	Liste aller Störungen, die diese Ressource betreffen.

Tabelle E.49.: ProductionUnitResource

E.50. Klassenstruktur PRODUCTPATTERN

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Bedingungen	
.DEALER BASEDEALER 0..1 →	Händler, bei dem das Fahrzeug erworben werden soll, bzw. der das Fahrzeug anbietet.
.HASCUSTOMER <i>Bool</i>	Überprüft, ob für ein Fahrzeug (bzw. einen Auftrag) eine Kundenbestellung existiert.
.ISFROZEN <i>Bool</i>	Auftrag eingefroren
.MARKET MARKET 0..1 →	Markt, auf dem das Fahrzeug gekauft werden soll bzw. angeboten wird.
.PARTS PART 0..n →	Teile
.PLANT PLANT 0..n →	Werk, dem der Auftrag zugeordnet ist.
.PLANTASSIGNMENT <i>Bool</i>	Auftrag wurde bereits einem Werk zugeordnet.
.PRODUCTCLASS PRODUCTCLASS 0..1 →	Fahrzeugtyp, nach dem gesucht wird. Muster passt auch auf alle weiter unten (abgeleiteten) im Beschreibungsbaum stehenden Fahrzeugtypen.
.PROPERTIES PROPERTY 0..n →	Liste von Eigenschaften, die sich mindestens im Fahrzeug befinden muß.
.SCRIPT <i>String</i>	Script, in dem weitere Bedingungen kodiert werden, die erfüllt sein müssen, damit das Suchmuster "passt".
.SETOperator <i>Bool</i>	Elemente gleichen Typs mit ODER verknüpfen (Default ist UND)
.SPEEDUPREQUIRED <i>Bool</i>	Gibt an, ob es sich bei einem Routing um ein beschleunigtes Routing handelt. Attribut wird von der Transportsteuerung verwendet.
Kategorie: Historie Prozessschritte	
.PARTOFROUTE PROCESS 0..1 →	Bestandteil der Route, über den das Fahrzeug bisher gelaufen ist.
.PARTOFROUTECARD <i>Int</i>	Wie oft muss das Fahrzeug über den Prozess aus Parameter <<Routen-Element>> gelaufen sein, damit die Bedingung erfüllt ist.
Kategorie: Zeit	
.VALIDEND <i>DateTime</i>	Endtermin des Gültigkeitszeitraums
.VALIDSTART <i>DateTime</i>	Starttermin des Gültigkeitszeitraums

Tabelle E.50.: ProductPattern

E.51. Klassenstruktur PRODUCTQUOTA

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Fahrzeugabsätze und Einbauraten	
.PROPAGATEGROUP <i>Bool</i>	Gruppe von Einbauraten abschließen. In die Fahrzeugquoten-Objekte der Liste "Gruppe von Einbauraten" wird ihrerseits diese Liste vermerkt.
.QUOTA OBJECTQUOTA 0..1 →	Einbauraten
.QUOTAGROUP PRODUCTQUOTA 0..n →	Gruppe von Einbauraten, für die eine gemeinsame Summe vereinbart werden soll (Attribut "Summe Einbauraten").
.QUOTATARGET ASSOCDATINSTRATE 0..1 →	Summe der spezifizierten Einbauraten in "Gruppe von Einbauraten" nach Normierung. Wird eine solche Gruppe ohne eine Summe für die Normierung - also ohne dieses Attribut - spezifiziert, wird auf 1.0 normiert.

Kategorie: Produktbeschreibung	
.PRODUCTCLASS PRODUCTCLASS 0..1 →	Fahrzeugbeschreibung

Tabelle E.51.: ProductQuota

E.52. Klassenstruktur PRODUCTSHARE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.PROPERTYCLASS PRODUCTCLASS 0..1 →	Fahrzeugbeschreibung
.SHARE ASSOCDATPLANTDISTRIBUTION 0..1 →	Anteil bzw. Anzahl Fahrzeuge

Tabelle E.52.: ProductShare

E.53. Klassenstruktur PRODUCTSTRUCTURE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.ORDERMODIFIER ORDERMODIFIER 0..n →	Auftragsanpassung
.PROPERTYCLASSES PRODUCTCLASS 0..n →	Liste aller Fahrzeugbeschreibungen des Experiments
.PROPERTYGROUPS PROPERTYGROUP 0..n →	Liste aller Eigenschaftsfamilien des Experiments

Tabelle E.53.: ProductStructure

E.54. Klassenstruktur PROPERTY

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Einplanung	
.FREEZINGPOINT <i>Int</i>	Der Einfrierzeitpunkt bestimmt, wie lange eine Eigenschaft (Ausstattung) eines Auftrags noch änderbar ist. Der Einfrierzeitpunkt gibt den Tag vor Rohbauauflage an, ab dem diese Eigenschaft eingefroren (nicht mehr änderbar) ist. Einfrierzeitpunkte können für Zulieferer von Eigenschaftenspezifikationen angegeben werden und beziehen sich dann auf die von diesem Zulieferer gelieferte Ausstattung eines Auftrags oder Fahrzeuges. Alternativ können Einfrierzeitpunkte für Eigenschaftenspezifikationen, Eigenschaftsfamilien angegeben werden. Diese beziehen sich dann auf alle Ausstattungen eines Auftrags oder Fahrzeuges, die durch diese Spezifikation gegeben sind bzw. diese Eigenschaft beinhalten bzw. eine Eigenschaft dieser Familie beinhalten. Es gilt jeweils der spezifischste Einfrierzeitpunkt: Einfrierzeitpunkte an Eigenschaftsfamilien überschreiben Einfrierzeitpunkte an Eigenschaftenspezifikationen überschreiben Einfrierzeitpunkte an Eigenschaftsfamilien. Einfrierzeitpunkte an Eigenschaftenspezifikationen überschreiben Einfrierzeitpunkte an Eigenschaftsfamilien. Einfrierzeitpunkte an Zulieferern überschreiben Einfrierzeitpunkte an Eigenschaftenspezifikationen.
Kategorie: Kosten	

.COSTS COSTSOBJECT 0..1 →	Einzelkosten Eigenschaft. Werden Kosten für die jeweils im Produkt vertretenen Eigenschaftenspezifikationen gesetzt, werden die an den Eigenschaften definierten Kosten ignoriert.
Kategorie: Produktbeschreibung	
.KEY String	Kürzel der Eigenschaft, mit der sie im Bestellschlüssel markiert ist.

Tabelle E.54.: Property

E.55. Klassenstruktur PROPERTYCONSTRAINT

Basisklasse: PROPERTYCONSTRAINTBASE (siehe Tabelle E.56)

Attribut / Typ	Beschreibung
Kategorie: Fahrzeugabsätze und Einbauraten	
.ACTUALRATES ASSOCDATINSTRATE 0..1 →	Liste mit Ist-Einbauraten für die Eigenschaft / Eigenschaftengruppe am zugehörigen Markt
.EXPECTEDRATES ASSOCDATINSTRATE 0..1 →	Liste mit prognostizierten Einbauraten für die Eigenschaft / Gruppe und einem zugehörigen Markt
Kategorie: Produktbeschreibung	
.PROPERTYSET PROPERTY 0..n →	Liste der Eigenschaften dieser Eigenschaften Gruppe

Tabelle E.55.: PropertyConstraint

E.56. Klassenstruktur PROPERTYCONSTRAINTBASE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.56.: PropertyConstraintBase

E.57. Klassenstruktur PROPERTYCONSTRAINTRULE

Basisklasse: PROPERTYCONSTRAINTBASE (siehe Tabelle E.56)

Attribut / Typ	Beschreibung
Kategorie: Bedingungen	
.PATTERN String	Boolescher Ausdruck, der die Eigenschaftenkombination beschreibt, die in einem Auftrag vorliegen muss, um die Regel für diesen Auftrag relevant zu machen.
Kategorie: Produktbeschreibung	
.CONSTRAINT String	Eigenschaften, die in einem Auftrag vorhanden sein müssen (Zwang) oder nicht enthalten sein dürfen (Verbot).
.OPERATOR Int	“Zwang“ oder “Verbot“
Kategorie: Zeit	
.VALIDFROM DateTime	Beginn Gültigkeit
.VALIDTO DateTime	Ende Gültigkeit

Tabelle E.57.: PropertyConstraintRule

E.58. Klassenstruktur PROPERTYDEV

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Kundenverhalten	
.DESTINATIONPROPERTY PROPERTY 0..1 →	nach Eigenschaft
.DEVIATION <i>Float</i>	Gibt an, wie stark ein Fahrzeug von seiner Spezifikation abweicht, wenn die erste Eigenschaft durch die zweite ersetzt ist.
.SOURCEPROPERTY PROPERTY 0..1 →	von Eigenschaft

Tabelle E.58.: PropertyDev

E.59. Klassenstruktur PROPERTYGROUP

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Datenbankausgaben	
.WRITE2DB <i>Bool</i>	Gibt an, ob der Simulator in den Simulationsergebnissen Informationen über verbaute Eigenschaften dieser Familie ausgibt.
Kategorie: Einplanung	
.FREEZINGPOINT <i>Int</i>	Der Einfrierzeitpunkt bestimmt, wie lange eine Eigenschaft (Ausstattung) eines Auftrags noch änderbar ist. Der Einfrierzeitpunkt gibt den Tag vor Rohbauauflage an, ab dem diese Eigenschaft eingefroren (nicht mehr änderbar) ist. Einfrierzeitpunkte können für Zulieferer von Eigenschaftenspezifikationen angegeben werden und beziehen sich dann auf die von diesem Zulieferer gelieferte Ausstattung eines Auftrags oder Fahrzeuges. Alternativ können Einfrierzeitpunkte für Eigenschaftenspezifikationen, Eigenschaften oder Eigenschaftenfamilien angegeben werden. Diese beziehen sich dann auf alle Ausstattungen eines Auftrags oder Fahrzeuges, die durch diese Spezifikation gegeben sind bzw. diese Eigenschaft beinhalten bzw. eine Eigenschaft dieser Familie beinhalten. Es gilt jeweils der spezifischste Einfrierzeitpunkt: Einfrierzeitpunkte an Eigenschaften überschreiben Einfrierzeitpunkte an Eigenschaftenfamilien. Einfrierzeitpunkte an Eigenschaftenspezifikationen überschreiben Einfrierzeitpunkte an Eigenschaften. Einfrierzeitpunkte an Zulieferern überschreiben Einfrierzeitpunkte an Eigenschaftenspezifikationen.
Kategorie: Produktbeschreibung	
.KEYPOSITION <i>Int</i>	Gibt an, ob diese Familie zum Schlüssel gehört, und wenn ja, an welche Position.
.PROPERTIES PROPERTY 0..n →	Liste der Eigenschaften dieser Eigenschaftenfamilie

Tabelle E.59.: PropertyGroup

E.60. Klassenstruktur PROPERTYSPEC

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Einplanung	

.FREEZINGPOINT <i>Int</i>	Der Einfrierzeitpunkt bestimmt, wie lange eine Eigenschaft (Ausstattung) eines Auftrags noch änderbar ist. Der Einfrierzeitpunkt gibt den Tag vor Rohbauauflage an, ab dem diese Eigenschaft eingefroren (nicht mehr änderbar) ist. Einfrierzeitpunkte können für Zulieferer von Eigenschaftenspezifikationen angegeben werden und beziehen sich dann auf die von diesem Zulieferer gelieferte Ausstattung eines Auftrags oder Fahrzeuges. Alternativ können Einfrierzeitpunkte für Eigenschaftenspezifikationen, Eigenschaften oder Eigenschaftenfamilien angegeben werden. Diese beziehen sich dann auf alle Ausstattungen eines Auftrags oder Fahrzeuges, die durch diese Spezifikation gegeben sind bzw. diese Eigenschaft beinhalten bzw. eine Eigenschaft dieser Familie beinhalten. Es gilt jeweils der spezifischste Einfrierzeitpunkt: Einfrierzeitpunkte an Eigenschaften überschreiben Einfrierzeitpunkte an Eigenschaftenfamilien. Einfrierzeitpunkte an Eigenschaftenspezifikationen überschreiben Einfrierzeitpunkte an Eigenschaften. Einfrierzeitpunkte an Zulieferern überschreiben Einfrierzeitpunkte an Eigenschaftenspezifikationen.
Kategorie: Fahrzeugabsätze und Einbauraten	
.ACTUALRATES ASSOCDATINSTRATE 0..1 →	Liste mit Ist-Einbauraten für die Eigenschaft / Eigenschaftengruppe am zugehörigen Markt
.EXPECTEDRATES ASSOCDATINSTRATE 0..1 →	Liste mit prognostizierten Einbauraten für die Eigenschaft / Gruppe und einem zugehörigen Markt
.LONGTERMEST ASSOCDATINSTRATE 0..1 →	Langfristiger Absatzplan
Kategorie: Kosten	
.COSTS COSTSOBJECT 0..1 →	Einzelkosten Eigenschaftenspezifikation
Kategorie: Produktbeschreibung	
.ISABSTRACT <i>Bool</i>	Abstrakte Eigenschaftenspezifikationen werden in der Simulation nicht gebaut. Stattdessen werden die Einbauraten der abgeleiteten Eigenschaft Spezifikationen auf 100 % normiert (womit nur diese gebaut werden).
.PROPERTY PROPERTY 0..1 →	Eigenschaft

Tabelle E.60.: PropertySpec

E.61. Klassenstruktur RECURRENTTASK

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.END <i>DateTime</i>	Datum, an dem die Terminserie endet.
.START <i>DateTime</i>	Datum, an dem die Terminserie beginnt.

Tabelle E.61.: RecurrentTask

E.62. Klassenstruktur RESOURCE

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.62.: Ressource

E.63. Klassenstruktur ROUTEINFORMATION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.DISTANCE <i>Float</i>	Distanz zwischen zwei Punkten
.TIME <i>Float</i>	Zeit, die benötigt wird, um vom Startpunkt zum Endpunkt zu gelangen.
Kategorie: Geographische Informationen	
.ENDPOINT LOCATION 0..1 →	Endpunkt
.STARTPOINT LOCATION 0..1 →	Startpunkt
Kategorie: Kosten	
.COST <i>Float</i>	Kosten, die für die Nutzung des Wegestücks erhoben werden.

Tabelle E.63.: RouteInformation

E.64. Klassenstruktur ROUTING

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.BASEGOODNESS <i>Int</i>	Basiswert, der in die Gütefunktion eingeht. So lassen sich zwei Routen gegeneinander priorisieren.
Kategorie: Bedingungen	
.BUFFERLEVEL <i>Int</i>	Pufferfüllung in Prozent. Ein Fahrzeug wird nur dann aus dem Puffer bewegt, wenn der eingestellte Füllgrad überschritten wird.
.PATTERN PRODUCTPATTERN 0..1 →	Eigenschaften, die ein Auftrag haben muss, um ausgewählt zu werden.
Kategorie: Distribution	
.OUT PROCESS 0..1 →	Nachfolgender Prozess eines Routings
.REFERENCE <i>Int</i>	Prozess, auf den sich die Bedingungen in der Routing Tabelle beziehen. Die Werte 0 und 1 bzw. ein nicht spezifizierter Wert führen zur Betrachtung der unmittelbaren Nachfolger. Ein Wert von 2 bezieht die Bedingungen auf den übernächsten Prozess usw..
.SUCCESSOR PROCESS 0..1 →	Das nächste Wegelement, auf dem der Transport eines Fahrzeugs fortgesetzt wird.
Kategorie: Zufallszahlen	
.PROBABILITY ASSOCDATPROBABILITY 0..1 →	Spezifiziert die Wahrscheinlichkeit, mit der die Route ausgewählt wird.

Tabelle E.64.: Routing

E.65. Klassenstruktur ROUTINGTABLE

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Bedingungen	

.CAPACITYLIMIT <i>Float</i>	Maximale Auslastung eines Distributionskanals, um weitere Fahrzeuge aufnehmen zu können. Gemessen als Faktor multipliziert mit der max. Kapazität des nächsten Transportes. Bsp.: Der nächste Transport ist ein Zug mit max. 20 Waggons, auf die jeweils 10 Einheiten passen. Hat der Parameter <<Kapazitätsgrenze>> den aktuellen Wert 2.0, so wird versucht, bei einer Pufferfüllung von 400 Einheiten auf ein anderes Transportmittel auszuweichen. Ist in den angehängten Routenobjekten der Parameter <<Referenz>> gesetzt, wird im dort spezifizierten nachgelagerten Objekt die Kapazität untersucht.
.CAPACITYLIMITRATING <i>Int</i>	Summand in der Gütefunktion für die Routenplanung. Default = -1
.PREFERCHEDULED <i>Bool</i>	Bevorzugt Route mit Transporten nach Fahrplan auswählen. Die parametrisierte Zahl fließt als Summand in die Gütefunktion zur Routenauswahl ein. Ist in den angehängten Routenobjekten der Parameter <<Referenz>> gesetzt, wird im dort spezifizierten nachgelagerten Objekt nach einem Fahrplan gesucht.
.PREFERCHEDULEDRATING <i>Int</i>	Summand in der Gütefunktion für die Routenplanung. Default = 1
.PRIORAND <i>Bool</i>	Die Auswahl eines Transportweges wird auf Basis von Wahrscheinlichkeitsverteilungen vorgenommen, die in Routingobjekten spezifiziert sind (siehe dort).
Kategorie: Distribution	
.ROUTING ROUTING 0..n →	Liste von alternativen Routen

Tabelle E.65.: RoutingTable

E.66. Klassenstruktur SALES

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Fahrzeugabsätze und Einbauraten	
.ACTUALSALES ASSOCDATSALES 0..1 →	Anzahl Auslieferungswünsche
.EXPECTEDSALES ASSOCDATSALES 0..1 →	Fahrzeugabsätze Soll
.LONGTERMEST ASSOCDATSALES 0..1 →	Langfristiger Absatzplan

Tabelle E.66.: Sales

E.67. Klassenstruktur SCENARIO

Attribut / Typ	Beschreibung
Kategorie: Szenariendefinition	
.CHANGESAPPLIED <i>Bool</i>	Gibt an, ob die Änderungen des Szenarios aktuell in der Datenbank umgesetzt sind.
.REFERRED EXPERIMENT EXPERIMENT 0..1 →	Ein Verweis auf das Experiment, das zu dem Szenario gehört.
.TRANSFORMATIONOBJECTS ABSTRANSFORMATIONCLASS 0..n →	Eine Liste mit allen Unterschieden zwischen dem Basisexperiment und dem Szenario

Tabelle E.67.: Scenario

E.68. Klassenstruktur SIMOBJECT

Attribut / Typ	Beschreibung
----------------	--------------

Kategorie: Allgemeine Attribute	
.COMMENT <i>String</i>	Kommentar
.NAME <i>String</i>	Der Name des Objekts
.SHORTNAME <i>String</i>	Kurzname
Kategorie: Datenimport	
.ALIASNAMES <i>String</i>	Ein Objekt kann unter allen in der Liste aufgeführten Namen identifiziert werden. Die verwendeten Namen müssen innerhalb der Klasse eindeutig sein.

Tabelle E.68.: SimObject

E.69. Klassenstruktur SIMPARAMETERS

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.ACTSCENARIO SCENARIO 0..1 →	Aktives Szenario
.BATCH <i>Bool</i>	Batch
.CACHESIZE <i>Int</i>	CacheSize
.DAYLIGHTSAVING <i>Bool</i>	Wenn aktiviert, wird in der Simulation eine Sommer-/Winterzeit Umstellung durchgeführt. Dies zieht eine Verlängerung der Simulationszeit nach sich.
.LEADIN <i>Int</i>	Anzahl Monate, die Simulation vor dem "eigentlichen" Experimentzeitraum beginnt.
.LEADOUT <i>Int</i>	Anzahl Monate, um die die Simulation nach dem "eigentlichen" Simulationszeitraum verlängert wird.
.LOGFILE <i>String</i>	Pfad zur LOG-Datei
.ORDERCACHE SIZE <i>Int</i>	Anzahl Aufträge, die auf einmal erzeugt werden. Eine größere Anzahl (z.B. 1000) ermöglicht die Erzeugung von Aufträgen mit Eigenschaften mit geringen Einbauraten.
.SCALE <i>Float</i>	Zu Testzwecken ist es möglich, Experimente skaliert zu simulieren. Alle Absätze und Kapazitäten werden mit dem Skalierungsfaktor multipliziert. Da durch die Skalierung Rundungseffekte auftreten können, sollten skalierte Experimente nur für Testzwecke berechnet werden.
.SIMEXP EXPERIMENT 0..1 →	Das ausgewählte Experiment
.TYPECHECKING <i>Bool</i>	Typprüfung für Wertverläufe. Durch Deaktivierung können Wertverläufe flexibler eingesetzt werden.
Kategorie: Datenbankausgaben	
.AFTERSIMULATIONSCRIPT <i>String</i>	Skript nach Simulationsende
.BACKGROUNDWRITES <i>Bool</i>	Wenn aktiviert, werden Daten über einen zweiten "Thread" in die Datenbank geschrieben.
.BULKCOPY <i>Bool</i>	Ausgaben über "Bulk Copy" Funktionen schreiben.
.CREATEINDEX <i>Bool</i>	Wenn aktiviert, werden in den Ausgabetafeln Indizes auf häufig genutzten Attributen erzeugt.
.WRITEBO <i>Bool</i>	Sollen die in der Simulation erzeugten Fahrzeuge in die Datenbank geschrieben werden?
.WRITEBOTRANSPTS <i>Bool</i>	Transportwege in die DB schreiben.
.WRITECUSTOMERORDERS <i>Bool</i>	Sollen die Kundenbestellungen in die Datenbank geschrieben werden?
.WRITEDBEXEC <i>Bool</i>	DB Ausgaben aktivieren.
.WRITEDEALERORDERS <i>Bool</i>	Sollen die Händlerbestellungen in die Datenbank geschrieben werden?

<code>.WRITEPROPERTIESID</code> <i>Bool</i>	Fahrzeugeigenschaften als IDs schreiben. Ermöglicht Auswertungen auf Ebene der Fahrzeugausstattungen.
<code>.WRITEPROPERTIESSTRING</code> <i>Bool</i>	Fahrzeugeigenschaften als Strings schreiben.
<code>.WRITEPROPERTYQUOTAS</code> <i>Bool</i>	Sollen prognostizierte, vereinbarte und bestellte Einbauraten in die Datenbank geschrieben werden?
<code>.WRITESTATALL</code> <i>Bool</i>	Sollen die Treuwerte und Bestände in die Datenbank geschrieben werden?
<code>.WRITESTATDEALER</code> <i>Bool</i>	Sollen die Händlerbestände in die Datenbank geschrieben werden?
<code>.WRITETRANSPORTS</code> <i>Bool</i>	Sollen die simulierten Transporte in die Datenbank geschrieben werden?
<code>.WRITEVOLUMES</code> <i>Bool</i>	Sollen die Händlerprognosen und Volumenvereinbarungen in die Datenbank geschrieben werden?
Kategorie: Händler / Kundenbestellungen	
<code>.CUSTOMERORDERS</code> <i>String</i>	Pfad zur Datei mit Kundenbestellungen
<code>.CUSTOMERORDERSPROPERTIES</code> <i>String</i>	Datei für Eigenschaften von Kundenbestellungen. Wird nur benötigt, wenn ausschließlich Kundenbestellungen erzeugt werden (es wird nicht simuliert).
<code>.DISABLEDEALERORDERS</code> <i>Bool</i>	Händlerbestellungen werden vom Simulator nicht mehr erzeugt. Nur noch eingelesene Händlerbestellungen werden durchgeführt.
<code>.GENERATECUSTOMERORDERS</code> <i>Bool</i>	Gibt an, ob die Kundenbestellungen erzeugt oder eingelesen werden sollen.
<code>.GENERATEORDERSONLY</code> <i>Bool</i>	Der Simulator erzeugt lediglich Aufträge auf Basis der Ist-Einbauraten bzw. Fahrzeuganzahlen. Eine vollständige Simulation wird nicht durchgeführt.
<code>.RECORDSEPARATOR</code> <i>String</i>	Satztrennzeichen für Auftragsdateien. Default ist ','
Kategorie: Netzwerk	
<code>.ENABLESUPPLIERS_1ST</code> <i>Bool</i>	Ist dieses Attribut auf "nein" gesetzt, werden bei der Simulation alle Zulieferer ignoriert. Simuliert wird lediglich das Netzwerk von Werken über Märkte, Händler bis zum Kunden.
<code>.ENABLESUPPLIERS_2ND</code> <i>Bool</i>	2nd-Tier Zulieferer aktivieren.

Tabelle E.69.: SimParameters

E.70. Klassenstruktur SUMRESOURCE

Basisklasse: RESOURCE (siehe Tabelle E.62)

Attribut / Typ	Beschreibung
Kategorie: Kapazität	
<code>.ISINTEGER</code> <i>Bool</i>	Gibt an, ob für die Kapazität nur ganze Zahlen zulässig sind.
<code>.SIMPLERESOURCE</code> <i>Bool</i>	Ressource stellt Kapazität stets vollständig zur Verfügung.
<code>.SUBRESSOURCES</code> RESOURCE 0..n →	Eine Liste mit untergeordneten Ressourcen. Die Position in der Liste gibt an, wieviele Prozent des Bedarfs der aktuellen Ressource von der untergeordneten Ressource gedeckt werden muß.
Kategorie: Sequenzbildung	
<code>.FAULTS</code> FAULT 0..n →	Liste aller Störungen, die diese Ressource betreffen.

Tabelle E.70.: SumRessource

E.71. Klassenstruktur SUPPLIER

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
----------------	--------------

Kategorie: Allgemeine Attribute	
.DISABLE <i>Bool</i>	Zulieferer ist deaktiviert.
Kategorie: Distribution	
.INBUFFERSUPPLIER BUFFER 0..1 →	Der Wareneingangspuffer muss definiert werden, wenn ein Inbound-Distributionssystem für den Zulieferer modelliert wurde. Vom Zulieferer des Zulieferers befüllte Behälter werden in diesem Puffer (vom Logistikdienstleister) abgestellt.
.OUTBUFFER BUFFER 0..1 →	Im Warenausgang werden Güter für den Abtransport bereitgestellt.
.SHIPMENTDELAY PROBABILITYDISTRIBUTION 0..1 →	Zeit zwischen Verfügbarkeit aller Teile zur Befüllung eines Behälters und der Transportbereitschaft des Behälters
Kategorie: Einplanung	
.SUPPLIERPARTINFORMATIONIN SUPPLIERPARTINFORMATIONIN 0..n →	Zuliefererinformationen für beschaffte Teile
Kategorie: Fertigung	
.FILLUP <i>Bool</i>	Ja: fehlende Teile werden im Wareneingangspuffer ergänzt.
Kategorie: Geographische Informationen	
.LOCATION DESTINATION 0..1 →	Ort, an dem der Zulieferer tätig ist.
Kategorie: Kapazität	
.CAPACITYACTUAL PRODUCTIONUNITRESOURCE 0..1 →	Ist-Kapazität des Zulieferers
.CAPACITYPLANNED PRODUCTIONUNITRESOURCE 0..1 →	Plankapazität des Zulieferers
Kategorie: Kosten	
.COSTS COSTSOBJECT 0..1 →	Einzelkosten für Teile, die von diesem Zulieferer hergestellt werden.
Kategorie: Produktbeschreibung	
.BILLOFMATERIALSUPPLIER BILLOFMATERIALRECORD 0..n →	Eine Liste mit allen Stücklisteninträgen für den Zulieferer

Tabelle E.71.: Supplier

E.72. Klassenstruktur SUPPLIERBTO

Basisklasse: SUPPLIER (siehe Tabelle E.71)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.LEADTIMEMULTIPLIER <i>Float</i>	Durchlaufzeitfaktor
.PLANDISTRIBUTIONTIME <i>Int</i>	Distributionszeit (Plan)
Kategorie: Bewertung Planung	
.COSTLATEARRIVAL <i>Float</i>	Bewertung verspäteter Ankunft
.COSTLATESHIPMENT <i>Float</i>	Bewertung Versandverzögerung
Kategorie: Distribution	
.CONTAINER CONTAINER 0..1 →	Behälter, der die Teile aufnimmt, die der Zulieferer produziert.
Kategorie: Einplanung	

.FREEZINGPOINT <i>Int</i>	Der Einfrierzeitpunkt bestimmt, wie lange eine Eigenschaft (Ausstattung) eines Auftrags noch änderbar ist. Der Einfrierzeitpunkt gibt den Tag vor Rohbauauflage an, ab dem diese Eigenschaft eingefroren (nicht mehr änderbar) ist. Einfrierzeitpunkte können für Zulieferer von Eigenschaftenspezifikationen angegeben werden und beziehen sich dann auf die von diesem Zulieferer gelieferte Ausstattung eines Auftrags oder Fahrzeuges. Alternativ können Einfrierzeitpunkte für Eigenschaftenspezifikationen, Eigenschaften oder Eigenschaftensfamilien angegeben werden. Diese beziehen sich dann auf alle Ausstattungen eines Auftrags oder Fahrzeuges, die durch diese Spezifikation gegeben sind bzw. diese Eigenschaft beinhalten bzw. eine Eigenschaft dieser Familie beinhalten. Es gilt jeweils der spezifischste Einfrierzeitpunkt: Einfrierzeitpunkte an Eigenschaften überschreiben Einfrierzeitpunkte an Eigenschaftensfamilien. Einfrierzeitpunkte an Eigenschaftenspezifikationen überschreiben Einfrierzeitpunkte an Eigenschaften. Einfrierzeitpunkte an Zulieferern überschreiben Einfrierzeitpunkte an Eigenschaftenspezifikationen.
Kategorie: Kapazität	
.BANDWIDTHREFDAY <i>Int</i>	Wochentag, an dem die gebuchten Kapazitäten als neue Referenzgrößen für die Bandbreitenbeschränkung (Woche) gespeichert werden. Bsp.: Bei einem Wert von "2" werden jeden Mittwoch die gebuchten Kapazitäten gespeichert. Gegen diese gespeicherte Kapazität werden Änderungswünsche gemessen. Ist eine Bandbreite nach oben von 5 % gesetzt und eine gebuchte Kapazität von 100 gespeichert, wird eine Anforderung von 7 Einheiten abgelehnt. Montag = 0, ..., Sonntag = 7. Default ist Sonntag.
.DISABLEPREFETCH <i>Bool</i>	Keine Einheiten vorproduzieren.
.INFINITECAPACITYPLAN <i>Bool</i>	Kapazitätsgrenzen werden für die erste Einplanung nicht berücksichtigt, jedoch bei der Erstellung der Tagesprogramme.
.LOWERBANDWIDTHDAY ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Möglichkeit der Änderung der Kapazitätsnutzung nach unten. Es darf die in einem Tag gebuchte Menge im weiteren zeitlichen Verlauf um max. n-Prozent unterschritten werden. Die Bedarfsschwankungen beim Zulieferer im zeitlichen Nachbereich lassen sich so minimieren.
.LOWERBANDWIDTHWEEK ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Möglichkeit der Änderung der Kapazitätsnutzung nach unten. Es darf die in einer Woche gebuchte Menge im weiteren zeitlichen Verlauf um max. n-Prozent unterschritten werden. Die Bedarfsschwankungen beim Zulieferer im zeitlichen Nachbereich lassen sich so minimieren.
.UPPERBANDWIDTHDAY ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Möglichkeit der Änderung der Kapazitätsnutzung nach oben. Es darf die an einem Tag gebuchte Menge im weiteren zeitlichen Verlauf um max. n-Prozent überschritten werden. Die Bedarfsschwankungen beim Zulieferer im zeitlichen Nachbereich lassen sich so minimieren.
.UPPERBANDWIDTHWEEK ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Möglichkeit der Änderung der Kapazitätsnutzung nach oben. Es darf die in einer Woche gebuchte Menge im weiteren zeitlichen Verlauf um max. n-Prozent überschritten werden. Die Bedarfsschwankungen beim Zulieferer im zeitlichen Nachbereich lassen sich so minimieren.
Kategorie: Maximale Ausbringung	

<p><code>.INCLUDEPREFETCHSOURCE</code> <i>Bool</i></p>	<p>[Allgemeines] Der Parameter definiert ausschließlich die Interpretation des Parameters <<Maximale Ausbringung>>. [Techn. Beschreibung] Der Zulieferer überprüft bei einer Kapazitätsanfrage stets, ob hierdurch die <<Maximale Ausbringung>> überschritten wird. Hierzu wird zunächst die max. Ausbringung für den fraglichen Zeitraum ermittelt. Um eine potenzielle Überschreitung festzustellen, werden max. drei Zahlen addiert und mit dieser max. Ausbringung verglichen. Der erste Summand ist die "reguläre Kapazität", also die Teile, die für eine Periode produziert werden sollen und auch für diese Periode vom Werk angefragt wurden. Der zweite Summand ist die Anzahl Teile, die für eine andere (spätere) Periode vorproduziert werden soll. Der dritte Summand ist die Anzahl Teile, die in einer anderen (früheren) Periode für diese Periode vorproduziert wurde. Ob die Summanden zwei und drei addiert werden, kann über zwei Parameter gesteuert werden (<<Vorproduktion einschließen (Quelle)>> und <<Vorproduktion einschließen (Ziel)>>).</p> <p>[Beispiel] Der Zulieferer A hat für den 19.5. bereits Aufträge für 150 Teile verbuchen können, für den 20.5. Aufträge für 200 Teile. Die max. Ausbringung des Zulieferers sei 175 Teile für den 19.5. und 200 Teile für den 20.5. Die Kapazität betrage für jeden Tag 200 Einheiten. Das Werk fragt für den 20.5. zusätzliche 50 Einheiten an. Am 20.5. ist sowohl die Kapazität erschöpft als auch die max. Ausbringung erreicht. Jedoch können Einheiten am 19.5. vorproduziert werden. Eine vorproduzierte Einheit wird als »Quelle« von Teilen am 19.5. und als »Ziel« am 20.5. gezählt. Die Anfrage des Werkes wird als Anfrage einzelner Teile durchgeführt. Die erste Anfrage führt zu folgendem Bild: Am 19.5.: Reguläre Kapazität 150, Quelle 1, Ziel 0. Am 20.5.: Reguläre Kapazität 200, Quelle 0, Ziel 1. Diese erste Anfrage wird positiv beantwortet, wenn der Parameter <<Vorproduktion einschließen (Ziel)>> mit <<falsch>> parametrisiert wird. Die Anfrage wird abgelehnt, wenn der Parameter mit <<wahr>> parametrisiert wird (200 + 0 + 1 > max. Ausbringung). Ist der Parameter <<Vorproduktion einschließen (Quelle)>> als <<wahr>> parametrisiert, können am 19.5. 25 Einheiten für den 20.5. vorproduziert werden. Die nächste Anfrage führt zu einem Überschreiten der max. Ausbringung (175). Ist der Parameter als <<falsch>> gesetzt, können die angefragten 50 Einheiten gebucht werden.</p>
<p><code>.INCLUDEPREFETCHTARGET</code> <i>Bool</i></p>	<p>[Allgemeines] Der Parameter definiert ausschließlich die Interpretation des Parameters <<Maximale Ausbringung>>. [Techn. Beschreibung] Der Zulieferer überprüft bei einer Kapazitätsanfrage stets, ob hierdurch die <<Maximale Ausbringung>> überschritten wird. Hierzu wird zunächst die max. Ausbringung für den fraglichen Zeitraum ermittelt. Um eine potenzielle Überschreitung festzustellen, werden max. drei Zahlen addiert und mit dieser max. Ausbringung verglichen. Der erste Summand ist die "reguläre Kapazität", also die Teile, die für eine Periode produziert werden sollen und auch für diese Periode vom Werk angefragt wurden. Der zweite Summand ist die Anzahl Teile, die für eine andere (spätere) Periode vorproduziert werden soll. Der dritte Summand ist die Anzahl Teile, die in einer anderen (früheren) Periode für diese Periode vorproduziert wurde. Ob die Summanden zwei und drei addiert werden, kann über zwei Parameter gesteuert werden (<<Vorproduktion einschließen (Quelle)>> und <<Vorproduktion einschließen (Ziel)>>).</p> <p>[Beispiel] Der Zulieferer A hat für den 19.5. bereits Aufträge für 150 Teile verbuchen können, für den 20.5. Aufträge für 200 Teile. Die max. Ausbringung des Zulieferers sei 175 Teile für den 19.5. und 200 Teile für den 20.5. Die Kapazität betrage für jeden Tag 200 Einheiten. Das Werk fragt für den 20.5. zusätzliche 50 Einheiten an. Am 20.5. ist sowohl die Kapazität erschöpft als auch die max. Ausbringung erreicht. Jedoch können Einheiten am 19.5. vorproduziert werden. Eine vorproduzierte Einheit wird gezählt als "Quelle" von Teilen am 19.5. und als "Ziel" am 20.5.. Die Anfrage des Werkes wird als Anfrage einzelner Teile durchgeführt. Die erste Anfrage führt zu folgendem Bild: Am 19.5. : Reguläre Kapazität 150, Quelle 1, Ziel 0. Am 20.5.: Reguläre Kapazität 200, Quelle 0, Ziel 1. Diese erste Anfrage wird positiv beantwortet, wenn der Parameter <<Vorproduktion einschließen (Ziel)>> mit <<falsch>> parametrisiert wird. Die Anfrage wird abgelehnt, wenn der Parameter mit <<wahr>> parametrisiert wird (200 + 0 + 1 > max. Ausbringung). Ist der Parameter <<Vorproduktion einschließen (Quelle)>> als <<wahr>> parametrisiert, können am 19.5. 25 Einheiten für den 20.5. vorproduziert werden. Die nächste Anfrage führt zu einem Überschreiten der max. Ausbringung (175). Ist der Parameter als <<falsch>> gesetzt, können die angefragten 50 Einheiten gebucht werden.</p>
<p><code>.MAXOUTPUT</code> <code>ASSOCDATSUPPLIEROUTPUT</code> 0..1 →</p>	<p>Maximale Anzahl Einheiten, die der Zulieferer liefern kann, incl. Einheiten aus dem Lager. Wird ein konstanter Wertverlauf gewählt (ohne Angabe eines Intervalls), nimmt der Simulator einen Bezug auf Tage an. Das heißt, ein konstanter Wert von 100 wird als 100 pro Tag interpretiert.</p>

.MAXOUTPUTRELATIVE <i>Bool</i>	Die max. Ausbringung wird als Obergrenze für den Anteil von Fahrzeugen aufgefasst, der mit dem vom Zulieferer hergestellten Teil ausgerüstet sein darf. Beträgt die max. Ausbringung des Zulieferers 0,05 und die Kapazität im Werk 1000 Fahrzeuge pro Tag, dann darf der Anteil der Produkte mit der fraglichen Eigenschaft 5 % nicht übersteigen. Das heißt, sind zu einem Zeitpunkt 500 Fahrzeuge eingeplant, dürfen nur 50 davon mit dieser Eigenschaft ausgestattet sein.
.MAXOUTPUTRELATIVEENABLE <i>Float</i>	Legt die Mindestbelegung im Werk fest, die erforderlich ist, um die Restriktion zu aktivieren. Bsp.: Im Wertverlauf max. Ausbringung steht 0,1, die Werkskapazität betrage 1000. Dann dürfen max. 100 Aufträge mit der vom Zulieferer hergestellten Eigenschaft ausgestattet sein. Ist nun MaxOutputRelative mit 0,5 parametrisiert, dann dürfen 50 Aufträge mit dieser Eigenschaft eingeplant werden, ohne dass die Restriktion beachtet wird. Der 51. Auftrag kann erst dann eingeplant werden, wenn min. 500 Aufträge im Werk eingeplant sind.

Tabelle E.72.: SupplierBTO

E.73. Klassenstruktur SUPPLIERBTS

Basisklasse: SUPPLIER (siehe Tabelle E.71)

Attribut / Typ	Beschreibung
Kategorie: Einplanung	
.FROZENPERIOD <i>Int</i>	Ein Zulieferer führt an jedem Planzeitpunkt eine Produktionsplanung für den Planhorizont (Tage) durch. Aufgrund dieser rollierenden Planung existiert in der Regel aus vergangenen Planzeitpunkten bereits eine Planung für einen Teil des Planhorizontes. Der eingefrorene Zeitraum bestimmt den Zeitraum in Tagen ab dem Planzeitpunkt, der in diesem Fall nicht umgeplant wird.
.MINPRODUCTION ASSOCDATNUMBERS 0..1 →	Die minimale Produktion definiert die Untergrenze der Tagesproduktion des Zulieferers.
.PLANNINGHORIZON <i>Int</i>	Der Planhorizont definiert, für welchen Zeitraum die Produktionsplanung des Zulieferers zu den Planungszeitpunkten gemacht wird. Die Planung berücksichtigt dabei ggf. auch nur lediglich den Teil der Absatzprognose, der für diesen Planzeitraum gegeben ist.
.POINTSINTIMEPLANNING POINTSINTIME 0..1 →	Die Planungszeitpunkte definieren, wann die Produktionsplanung gemacht wird.
Kategorie: Fertigung	
.SUPPLIERPARTINFORMATIONOUT SUPPLIERPARTINFORMATIONOUT 0..n →	Zuliefererinformationen für prod. Produkte

Tabelle E.73.: SupplierBTS

E.74. Klassenstruktur SUPPLIERPARTINFORMATION

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Produktbeschreibung	
.PART PART 0..1 →	Teil

Tabelle E.74.: SupplierPartInformation

E.75. Klassenstruktur SUPPLIERPARTINFORMATIONIN

Basisklasse: SUPPLIERPARTINFORMATION (siehe Tabelle E.74)

Attribut / Typ	Beschreibung
----------------	--------------

Kategorie: Allgemeine Attribute	
.DEMANDLOOKAHEAD <i>Int</i>	Zeit zwischen Bestelleingang beim Zulieferer und tatsächlichem Bedarf beim Kunden des Zulieferers
Kategorie: Einplanung	
.MINSTOCKLEVEL ASSOCDATSALES 0..1 →	Sicherheitsbestand
.ORDERBATCHSIZE <i>Int</i>	Bestellosgröße
.PARTSUPPLIERORDEREDBY SUPPLIER 0..1 →	Zulieferer für Auffüllbestellung
.STOCKLEVELMONITORINGHORIZON <i>Int</i>	Horizont des Bestandsmonitoring

Tabelle E.75.: SupplierPartInformationIn

E.76. Klassenstruktur SUPPLIERPARTINFORMATIONOUT

Basisklasse: SUPPLIERPARTINFORMATION (siehe Tabelle E.74)

Attribut / Typ	Beschreibung
Kategorie: Bestandsplanung	
.MINSTOCKMONITOREDBUFFER ASSOCDATNUMBERS 0..1 →	Bei Unterschreiten des minimalen Lagerbestandes im überwachten Puffer werden zusätzliche Produktionsvolumina initiiert.
.MONITOREDBUFFERSUPPLIER BUFFER 0..1 →	Der Bestand des überwachten Puffers wird vom Zulieferer überwacht. Bei Unterschreitung des Mindestbestandes werden vom Zulieferer Produktionsaufträge generiert.
Kategorie: Distribution	
.PARTCONTAINER CONTAINER 0..1 →	Behälter für Teile
.PARTSTOCKPREFILL <i>Int</i>	Warenausgangsvorfüllung
Kategorie: Einplanung	
.PARTFORECAST ASSOCDATSALES 0..1 →	Der Zulieferer richtet seine Produktionsplanung einerseits nach der Absatzprognose, andererseits nach dem Bestand im überwachten Puffer aus. Es ist zwingend erforderlich, dass mindestens eins der Attribute "Absatzprognose" oder "überwachter Puffer" gesetzt ist.

Tabelle E.76.: SupplierPartInformationOut

E.77. Klassenstruktur SUPPLIERPRIORITISATION

Basisklasse: WEIGHTEDREFERENCE (siehe Tabelle E.87)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DEMANDLOOKAHEAD <i>Int</i>	Zeit zwischen Bestelleingang beim Zulieferer und tatsächlichem Bedarf beim Kunden des Zulieferers
Kategorie: Distribution	
.CONTAINER CONTAINER 0..1 →	Der Behälter, der für Transport dieser Eigenschaftenspezifikation vom dem Zulieferer zum Werk Verwendung findet.
Kategorie: Kapazität	
.STOCKPREFILL <i>Int</i>	Warenausgang Vorfüllung
Kategorie: Sourcing	
.REFERENCEDOBJECT SUPPLIER 0..1 →	Zulieferer
.WEIGHT ASSOCDATSALES 0..1 →	Priorität des Zulieferers als zeitlicher Verlauf

Tabelle E.77.: SupplierPrioritisation

E.78. Klassenstruktur TIMESPAN

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.ENDINT <i>DateTime</i>	Intervallende
.STARTINT <i>DateTime</i>	Intervallbeginn

Tabelle E.78.: Timespan

E.79. Klassenstruktur TRANSPORTDEFINITION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Distribution	
.TRANSPORT TRANSPORTER 0..1 →	Transportmittel
Kategorie: Kapazität	
.TRANSPORTCOUNT ASSOCDATCAPACITYINT 0..1 →	TransportCount

Tabelle E.79.: TransportDefinition

E.80. Klassenstruktur TRANSPORTER

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
Kategorie: Abmessungen	
.HEIGHT <i>Float</i>	Höhe
.LENGTH <i>Float</i>	Länge
.WIDTH <i>Float</i>	Breite
Kategorie: Gewicht	
.MAXWEIGHT <i>Float</i>	Max. Gewicht

Tabelle E.80.: Transporter

E.81. Klassenstruktur TRANSPORTRELATION

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
----------------	--------------

Kategorie: Distribution	
.DISTRIBUTIONCHANNELS PROCESS 0..n →	Distributionskanäle für die Parameter gesetzt werden.
.MAXWAIT Int	Wartezeit in Anzahl Transporte, die gewartet wird, bevor ein alternativer Kanal gewählt wird. Wenn nicht gesetzt, wird ein Wert von Null angenommen.
.PLANT PLANT 0..1 →	Werk, von dem Fahrzeuge geliefert werden.

Tabelle E.81.: TransportRelation

E.82. Klassenstruktur UNIFORMDISTRIBUTION

Basisklasse: PROBABILITYDISTRIBUTION (siehe Tabelle E.46)

Attribut / Typ	Beschreibung
Kategorie: Verteilungsparameter	
.END Float	Untere Schranke des Intervalls, aus dem Werte gezogen werden.
.START Float	Untere Schranke des Intervalls, aus dem Werte gezogen werden.

Tabelle E.82.: UniformDistribution

E.83. Klassenstruktur VISUALOBJECT

Basisklasse: SIMOBJECT (siehe Tabelle E.68)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.83.: VisualObject

E.84. Klassenstruktur WEEKLYRCT

Basisklasse: RECURRENTTASK (siehe Tabelle E.61)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.DISTANCE Int	Bezeichnet den Abstand der Ereignisse in Wochen. Ein Wert von "4" führt zu einem Ereignis alle vier Wochen.
.MINUTEOFFSET Int	Uhrzeit des Tages, an dem das Ereignis ausgelöst wird, definiert in Minuten seit Mitternacht. Beispiel: 630 löst ein Ereignis um 10:30 Uhr aus. Wird dieser Offset nicht angegeben, finden Ereignisse um 6:00 Uhr morgens statt.
.OFFSET Int	Ein Ereignis tritt jeweils am bezeichneten Wochentag ein. 1 entspricht Montag, 7 entspricht Sonntag.

Tabelle E.84.: WeeklyRCT

E.85. Klassenstruktur WEEKLYTIMESPAN

Basisklasse: TIMESPAN (siehe Tabelle E.78)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.85.: WeeklyTimespan

E.86. Klassenstruktur WEEKTIMERESSOURCE

Basisklasse: RESOURCE (siehe Tabelle E.62)

Attribut / Typ	Beschreibung
Kategorie: Kapazität	
.FLEXBAND_M ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Bandbreite, innerhalb der die Kapazität nach unten verändert werden darf.
.FLEXBAND_P ASSOCDATFLEXIBILITYBANDWIDTH 0..1 →	Bandbreite, innerhalb der die Kapazität nach oben verändert werden darf.
.FRIDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.HOLIDAYS VECTORHOLIDAYS 0..1 →	Liste aller arbeitsfreien Tage
.ISINTEGER <i>Bool</i>	Gibt an, ob für die Kapazität nur ganze Zahlen zulässig sind.
.MONDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.SATURDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.SIMPLERESOURCE <i>Bool</i>	Ressource stellt Kapazität stets vollständig zur Verfügung.
.SUNDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.THURSDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.TUESDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
.WEDNESDAY ASSOCDATCAPACITY 0..1 →	Arbeitszeit oder Ausbringung an dem Wochentag
Kategorie: Sequenzbildung	
.FAULTS FAULT 0..n →	Liste aller Störungen, die diese Ressource betreffen.

Tabelle E.86.: WeekTimeRessource

E.87. Klassenstruktur WEIGHTEDREFERENCE

Basisklasse: VISUALOBJECT (siehe Tabelle E.83)

Attribut / Typ	Beschreibung
Kategorie: Allgemeine Attribute	
.REFERENCEDOBJECT SIMOBJECT 0..1 →	Referenziertes Objekt

Tabelle E.87.: WeightedReference

E.88. Klassenstruktur YEARLYTIMESPAN

Basisklasse: TIMESPAN (siehe Tabelle E.78)

Attribut / Typ	Beschreibung
----------------	--------------

Tabelle E.88.: YearlyTimespan

F. Anhang – Tabellen

Simulationsergebnisse

F.1. Aufträge

F.1.1. Tabelle OUT_BO

Attribut	Typ	Beschreibung
BO_ID	int	Eindeutiger Identifizierer
NoAccounting	int	Aufträge, die nicht eingeplant werden konnten, aber am Ende der Simulation noch keinem Kunden zugeordnet werden konnten, werden mit einem Attributwert ungleich Null gekennzeichnet. Diese Aufträge können z.B. aus der Kennzahlenberechnung ausgenommen werden.
BOKey	[char] (64)	Schlüssel, der aus dem Attribut »Key« in den jeweiligen Eigenschaften eines Auftrags und dem Attribut »KeyPosition« der Eigenschaftenfamilie berechnet wird.
Dealer_ID	int	ID des bestellenden Händlers
CstType_ID	int	ID des Kundentypen, der dem Kunden, der das Produkt abgenommen hat, zugeordnet ist.
BOClass_ID	int	Produktklasse des Produkts
Plant_ID	int	ID des Werkes, in dem das Produkt gefertigt wurde.
Market_ID	int	ID des Marktes, dem der Händler zugehörig ist, der das Produkt bestellt hat.
PRT_ID	int	Verweis in Tabelle OUT_Properties. Jede Zeile in dieser Tabelle steht für eine Eigenschaft, die dem Auftrag zugeordnet ist.
PTS_ID	int	Verweis in Tabelle OUT_Parts. Jede Zeile in dieser Tabelle steht für ein Teil, welches benötigt wurde, um das Produkt zu fertigen.
Dlr_Ordered	datetime	Zeitpunkt, zu dem eine Händlerbestellung aufgegeben wurde.
Dlr_DesiredFrom	datetime	Beginn des Intervalls, in dem ein Händler eine Fertigstellung des Produkts wünscht.
Dlr_DesiredTo	datetime	Ende des Intervalls, in dem ein Händler eine Fertigstellung des Produkts wünscht.
Cst_Ordered	datetime	Zeitpunkt, an dem der Kunde seine Bestellung abgegeben hat.
Cst_Desired	datetime	Zeitpunkt, zu dem der Kunde eine Fertigstellung des Produkts wünscht.
DesiredStartProduction_Start	datetime	Aus dem gewünschten Termin für die Fertigstellung (Attribut »Dlr_DesiredFrom«) abgeleiteter Termin für den Fertigungsstart, der zu einer voraussichtlichen Fertigstellung zu diesem Termin führt.
DesiredStartProduction_End	datetime	Aus dem gewünschten Termin für die Fertigstellung (Attribut »Dlr_DesiredTo«) abgeleiteter Termin für den Fertigungsstart, der zu einer voraussichtlichen Fertigstellung zu diesem Termin führt.
Cst_Accuracy	float	Übereinstimmung der Ausstattung des ausgelieferten Produktes mit der vom Kunden zum Zeitpunkt der Bestellung genannten Wunschausstattung
DistributionChannel_ID	int	ID des ersten Distributionskanals, auf dem das Produkt zum Abnehmer (Importeur/Kunden) transportiert wurde.
LastDistributionChannel_ID	int	ID des letzten Distributionskanals, auf dem das Produkt zum Abnehmer (Importeur/Kunden) transportiert wurde.
StockDC_ID	int	ID des Puffers bzw. Distributionskanals, in dem ein Produkt auf einen Abnehmer wartet.
FirstPlannedSOP	datetime	Geplanter Zeitpunkt des Fertigungstermins, wie er während der ersten Einplanung berechnet wird.
FirstPlanned	datetime	Der letzte geplante Zeitpunkt des Fertigungsbeginns vor der tatsächlichen Fertigung. Eine Neuberechnung wird nach jeder Umplanung vorgenommen.

PlannedEOP_PlantAssgn	datetime	Geplanter Zeitpunkt des Fertigungsendes zum Zeitpunkt der Festlegung eines Werkes, in dem ein Auftrag gefertigt werden soll.
LastPlanned	datetime	Der letzte berechnete Termin für das Fertigungsende vor der eigentlichen Fertigung. Eine Neuberechnung wird nach jeder Umplanung vorgenommen.
Replanned	int	Gesamtanzahl der Neueinplanungen aufgrund von Störungen oder Verwirbelungsfunktionen
ReplannedPlan	int	Anzahl der Neueinplanungen aufgrund von Störungen oder Verwirbelungsfunktionen vom Zeitpunkt der ersten Einplanung bis zur Festlegung eines Werkes, welches den Auftrag fertigt.
ReplannedPlant	int	Anzahl der Neueinplanungen aufgrund von Störungen oder Verwirbelungsfunktionen vom Zeitpunkt der Festlegung eines Werkes, welches den Auftrag fertigt, bis zur tatsächlichen Fertigung.
ReplannedSegmentChange	int	Anzahl Umplanungen, die zu einer Verschiebung über eine »Segmentgrenze« hinweg geführt haben. Die Segmentgrenzen werden durch den Zeitpunkt der Zuordnung eines Werkes und dem Fixieren eines Auftrags definiert. Wird also z.B. der bereits fixierte Auftrag aufgrund einer Störung umgeplant und ist im Anschluss wieder durch einen Kunden anpassbar, wird dieser Zähler erhöht.
PlantAssignment	datetime	Zeitpunkt der Zuordnung eines Werkes, welches den Auftrag fertigt.
ChOwner	datetime	Zeitpunkt der endgültigen Fixierung eines Auftrags
ProductionStart	datetime	Zeitpunkt, an dem die Fertigung des Auftrags begonnen hat.
ProductionEnd	datetime	Zeitpunkt, an dem die Fertigung des Auftrags abgeschlossen wurde.
TransportRequested	datetime	Zeitpunkt nach Beendigung der Fertigung, an dem ein Transportmittel angefordert wurde.
LeavingBuffer	datetime	Zeitpunkt, zu dem der erste Puffer im Distributionsweg verlassen wurde. Im direkten Anschluss beginnt der erste Transport des Produkts.
EstDealerArrival	datetime	Voraussichtliche Ankunft des Produkts beim Händler. Berechnet zum Zeitpunkt der ersten Einplanung.
DealerArrival	datetime	Ankunft des Produkts beim ausliefernden Händler bzw. beim Importeur
Delivered	datetime	Zeitpunkt der Auslieferung eines Produkts
LeadtimeUnit	int	Einheit, in der Durchlaufzeiten berechnet werden.
LeadtimePlant	int	Zeit von der Fixierung des Werkes, welches den Auftrag baut, bis zur Fertigstellung des Fahrzeugs.
LeadtimeProduction	int	Zeit vom Beginn der Fertigung bis zur Fertigstellung des Fahrzeugs
LeadtimeShipping	int	Zeit von der Fertigstellung des Auftrags bis zur Auslieferung
TimeToTransport	int	Wartezeit vom Ende der Fertigung bis zum Abtransport mit dem ersten Transportmittel
LeadtimeTransport	int	Zeit vom Verlassen des ersten Puffers bis zur Ankunft am Ziel
LeadtimeOrder	int	Zeit von der Bestellung eines Produkts bis zur Auslieferung an den Händler
LeadtimeCustOrder	int	Zeit von der Bestellung durch den Kunden bis zur Auslieferung
CstOrderToProductionStart	int	Zeit von der Kundenbestellung bis zum Beginn der Fertigung des Auftrags
Locating	int	Wurde das Produkt von einem Händler »übernommen«, der nicht der bestellende Händler ist.

Source	[char] (2)	Status des Auftrags zum Zeitpunkt der Zuordnung eines Kundenauftrags. Das Kennzeichen »DS« steht z.B. für eine Abnahme eines Produkts durch einen Kunden aus dem Bestand bereits gefertigter Produkte bei einem Händler.
		AO Kunde akzeptiert (evtl. angepasste) Händlerbestellung vor dem »Einfrieren« des Auftrages.
		RA Kunde akzeptiert (evtl. angepasste) Händlerbestellung vor dem »Einfrieren« des Auftrages, aber nach der Zuordnung des Auftrages zu einem Werk.
		DS Kunden erhält Produkt nach Ankunft des gefertigten Produktes beim Händler.
		NO Kunde akzeptiert Händlerbestellung nach dem »Einfrieren« des Auftrages, also in der vom Händler bestellten Spezifikation.
		DO Produkt wurde für den Kunden bestellt (Direktverbuchung).
		IM Produkt wurde von einem »Importeur« bestellt.
		DR Produkt hat keinen Abnehmer gefunden.
		TR Händlerbestellung konnte nicht eingeplant werden.
		SE Produkt befand sich bei Simulationsende bei einem Händler; innerhalb des Simulationszeitraumes wurde kein Abnehmer gefunden.
XX Auftragsstatus konnte nicht ermittelt werden.		
FirstPlantSerial	int	Erster eindeutiger Identifizierer, der zum Zeitpunkt der Fixierung des Werkes vergeben wird.
PlantSerial	int	Letzter eindeutiger Identifizierer, der zum Zeitpunkt der Fixierung des Werkes vergeben wird. Eine Änderung des Identifizierers kann aufgrund einer Umplanung verursacht werden oder durch eine Störung geschehen.
StockArrival	datetime	Zeitpunkt der Ankunft eines Produkts in einem Lager, von dem aus das Produkt nur weitertransportiert wird, wenn ein Kunde das Produkt abnimmt.
LeavingStock	datetime	Zeitpunkt des Abtransportes eines Produkts aus einem Lager, von dem aus das Produkt nur weitertransportiert wird, wenn ein Kunde das Produkt abnimmt.
TimeOrderPool	int	Zeit zwischen Einplanung eines Auftrags und der Abnahme durch einen Kunden oder Fixierung des Auftrags
TimeInStock	int	Zeit zwischen Ankunft in einem Lager, von dem aus das Produkt nur weitertransportiert wird, wenn ein Kunde das Produkt abnimmt, und dem Weitertransport.
DelReliabilityCust	int	Liefertreue Kunde. Einheit ist Woche oder Tag, abhängig von der Länge des Intervalls, für das der Händler »Auffüllbestellungen« abgeben muss.
DelReliabilityDealer	int	Liefertreue Handel. Einheit ist Woche oder Tag, abhängig von der Länge des Intervalls, für das der Händler »Auffüllbestellungen« abgeben muss.
ProdEndReliability	int	Treuegrad, mit dem der geplante Termin (Tag) für die Fertigstellung eines Produkts erreicht wird.
PlanReliability	int	Treuegrad, mit dem der geplante Termin (Tag) für den Fertigungsstart eines Produkts erreicht wird.
DistributionSteps	int	Anzahl Distributionssegmente, über die das Produkt transportiert wurde.
CapacityRestriction1	int	Erste Restriktion (Kapazitätsengpass Werk oder Lieferant), aufgrund der der Auftrag nicht eingeplant werden konnte.
CRCount1	int	Anzahl versuchter Buchungen, die wegen des Engpasses abgelehnt wurden.
LimitingResource1	[char] (25)	Bezeichnung der Ressource, die Engpass verursacht hat .
CapacityRestriction2	int	Wie »CapacityRestriction1«. Weitere Engpassressource.
CRCount2	int	Siehe oben.
LimitingResource2	[char] (25)	Siehe oben.
CapacityRestriction3	int	Siehe oben.
CRCount3	int	Siehe oben.
LimitingResource3	[char] (25)	Siehe oben.
CapacityRestriction4	int	Siehe oben.

CRCCount4	int	Siehe oben.																								
LimitingResource4	[char] (25)	Siehe oben.																								
OBJ_CostMaterial	float	Materialkosten für diesen Auftrag																								
OBJ_CostProduction	float	Fertigungskosten für diesen Auftrag																								
OBJ_CostDistribution	float	Distributionskosten für diesen Auftrag																								
OBJ_CostSales	float	Vertriebskosten für diesen Auftrag. Zu den Vertriebskosten zählen auch Rabatte, die einem Kunden eingeräumt werden, wenn ein Produkt lange Zeit gelagert wird (siehe Klasse CUSTOMERTYPE).																								
OBJ_CostAdmin	float	Verwaltungskosten für diesen Auftrag																								
OBJ_Revenue	float	Ertrag für diesen Auftrag																								
CC_AccountingDays ¹	int	Anzahl Tage, für die Kapitalbindungskosten berechnet werden.																								
CapitalCommitmentCost	float	Kapitalbindungskosten																								
CostInfo	int	<p>Kostenarten, die für Kosteninformationen zugrunde gelegt wurden. Summe wird gebildet aus den Kostenarten:</p> <table border="1"> <tr><td>ECT_CM_Base</td><td>2</td></tr> <tr><td>ECT_CM_Material</td><td>4</td></tr> <tr><td>ECT_CM_Production</td><td>8</td></tr> <tr><td>ECT_CM_Distribution</td><td>16</td></tr> <tr><td>ECT_CM_Sales</td><td>32</td></tr> <tr><td>ECT_CM_Admin</td><td>64</td></tr> <tr><td>ECT_CM_Invest</td><td>128</td></tr> <tr><td>ECT_OBJ_Material</td><td>256</td></tr> <tr><td>ECT_OBJ_Production</td><td>512</td></tr> <tr><td>ECT_OBJ_Distribution</td><td>1024</td></tr> <tr><td>ECT_OBJ_Sales</td><td>2048</td></tr> <tr><td>ECT_OBJ_Revenue</td><td>4096</td></tr> </table> <p>Der Präfix »CM« steht für Gemeinkosten, der Präfix »OBJ« für Einzelkosten. Beispiel: Ein Wert für das Attribut »CostInfo« von 28 bedeutet, dass die Kostenarten ECT_CM_Material, ECT_CM_Production und ECT_CM_Distribution berücksichtigt wurden (4 + 8 + 16 = 28).</p>	ECT_CM_Base	2	ECT_CM_Material	4	ECT_CM_Production	8	ECT_CM_Distribution	16	ECT_CM_Sales	32	ECT_CM_Admin	64	ECT_CM_Invest	128	ECT_OBJ_Material	256	ECT_OBJ_Production	512	ECT_OBJ_Distribution	1024	ECT_OBJ_Sales	2048	ECT_OBJ_Revenue	4096
ECT_CM_Base	2																									
ECT_CM_Material	4																									
ECT_CM_Production	8																									
ECT_CM_Distribution	16																									
ECT_CM_Sales	32																									
ECT_CM_Admin	64																									
ECT_CM_Invest	128																									
ECT_OBJ_Material	256																									
ECT_OBJ_Production	512																									
ECT_OBJ_Distribution	1024																									
ECT_OBJ_Sales	2048																									
ECT_OBJ_Revenue	4096																									

Tabelle F.1.: Ergebnisdaten Auftrag und Produkt (Tabelle OUT_BO)

F.1.2. Tabelle OUT_CustomerOrder

Attribut	Typ	Beschreibung
CST_ID	int	Eindeutiger Identifizierer
NoAccounting	int	Aufträge, die aufgrund eines Ressourcenengpasses nicht eingeplant werden konnten, werden mit einem Attributwert ungleich Null gekennzeichnet. Diese Aufträge können z.B. aus der Kennzahlenberechnung ausgenommen werden.
BOKey	char (64)	Schlüssel, der aus dem Attribut »Key« in den jeweiligen Eigenschaften eines Auftrags und dem Attribut »KeyPosition« der Eigenschaftenfamilie berechnet wird.
Dealer_ID	int	ID des Händlers, bei dem der Kunde ein Produkt bestellt.
Plant_ID	int	ID des Werkes, in dem Auftrag gefertigt wird.
Market_ID	int	Markt, dem der Händler zugeordnet ist, bei dem der Kunde das Produkt bestellt.
BO_ID	int	ID des Produkts (Referenz in Tabelle OUT_BO), das der Kunde akzeptiert und bestellt hat.
BO_CLASS_ID	int	Produktklasse des Produkts, das der Kunde bestellen wollte. Muss nicht gleich der Produktklasse des Produkts sein, das tatsächlich ausgeliefert wurde (siehe zugeordneter Eintrag in Tabelle OUT_BO). Ein Kunde kann z.B. aufgrund eines hohen gewährten Rabattes eine andere, als die ursprünglich geplante Produktklasse wählen.

¹Voller Attributname: »CapitalCommitmentAccountingDays«

PRT_ID	int	Verweis in Tabelle OUT_Properties. Jede Zeile in dieser Tabelle steht für eine Eigenschaft, die dem Auftrag zugeordnet ist. Die Eigenschaften repräsentieren den ursprünglichen Kundenwunsch. Die Eigenschaften des tatsächlich gelieferten Produkts werden über die Referenz in Tabelle OUT_BO repräsentiert.
CustomerType_ID	int	ID der Kundentypen des Kunden, der Produkt bestellt.
Ordered	datetime	Zeitpunkt der Bestellung des Produkts beim Händler
Desired	datetime	Vom Kunden gewünschter Termin der Fertigstellung des Produkts
LostSales	int	Wenn Wert definiert und ungleich Null ist, konnte für den Kunden kein Produkt in einer für diesen akzeptablen Zeit bestellt werden; der Kunde hat kein Produkt bestellt und ist als Kunde »verloren«.
OfferedDlrArrival	datetime	Letzter vom Kunden als »zu lang« abgelehnter Liefertermin
CapacityRestriction1	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
CRCount1	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
LimitingResource1	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1.
CapacityRestriction2	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
CRCount2	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
LimitingResource2	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1.
CapacityRestriction3	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
CRCount3	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
LimitingResource3	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1.
CapacityRestriction4	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
CRCount4	int	Analog dem gleichnamigen Attribut in Tabelle F.1.
LimitingResource4	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1.
Deviation	float	Anzahl Eigenschaften, die im gelieferten Produkt vom ursprünglichen Kundenwunsch abweichen.
Accuracy	float	Abweichung des gelieferten Produkts bezogen auf den ursprünglichen Kundenwunsch. Berechnet auf Basis der parametrisierten Abweichungen im zugeordneten Objekt der Klasse CUSTOMERTYPE.
SpecialSales	int	Produkt wurde im Rahmen einer »Sonderaktion« unter Gewährung eines Rabattes an einen Kunden verkauft. Parameter hierfür stehen im zugeordneten Objekt der Klasse CUSTOMERTYPE.

Tabelle F.2.: Ergebnisdaten Kundenaufträge (Tabelle OUT_CustomerOrder)

F.1.3. Tabelle OUT_DealerOrder

Attribut	Typ	Beschreibung
DOR_ID	int	Eindeutiger Identifizierer
NoAccounting	int	Aufträge, die aufgrund eines Ressourcenengpasses nicht eingeplant werden konnten, werden mit einem Attributwert ungleich Null gekennzeichnet. Diese Aufträge können z.B. aus der Kennzahlenberechnung ausgenommen werden.
BO_ID	int	ID des Produkts (Referenz in Tabelle OUT_BO), das der Kunde akzeptiert und bestellt hat.
Dealer_ID	int	ID des bestellenden Händlers
BO_CLASS_ID	int	Produktklasse des bestellten Produkts
PRT_ID	int	Verweis in Tabelle OUT_Properties. Jede Zeile in dieser Tabelle steht für eine Eigenschaft, die dem Auftrag zugeordnet ist.
Ordered	datetime	Zeitpunkt der Bestellung des Produkts
DesiredFrom	datetime	Beginn des Intervalls, in dem der Händler eine Fertigstellung des Produkts wünscht.
DesiredTo	datetime	Beginn des Intervalls, in dem der Händler eine Fertigstellung des Produkts wünscht.
CapacityRestriction1	int	Analog dem gleichnamigen Attribut in Tabelle F.1
CRCount1	int	Analog dem gleichnamigen Attribut in Tabelle F.1
LimitingResource1	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1
CapacityRestriction2	int	Analog dem gleichnamigen Attribut in Tabelle F.1
CRCount2	int	Analog dem gleichnamigen Attribut in Tabelle F.1
LimitingResource2	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1
CapacityRestriction3	int	Analog dem gleichnamigen Attribut in Tabelle F.1
CRCount3	int	Analog dem gleichnamigen Attribut in Tabelle F.1

LimitingResource3	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1
CapacityRestriction4	int	Analog dem gleichnamigen Attribut in Tabelle F.1
CRCCount4	int	Analog dem gleichnamigen Attribut in Tabelle F.1
LimitingResource4	char (25)	Analog dem gleichnamigen Attribut in Tabelle F.1

Tabelle F.3.: Ergebnisdaten Händlernaufträge (Tabelle OUT_DealerOrder)

F.1.4. Tabelle OUT_Properties

Attribut	Typ	Beschreibung
PRTIDN_ID	int	Eindeutiger Identifizierer
PRT_ID	int	ID, die von Einträgen aus den Tabellen OUT_BO, OUT_CustomerOrder und OUT_DealerOrder referenziert wird.
PropertyGroup_ID	int	Eigenschaftenfamilie, der die Eigenschaft angehört.
PropertySpec_ID	int	ID der kodierten Eigenschaftenspezifikation
Property_ID	int	ID der kodierten Eigenschaft

Tabelle F.4.: Ergebnisdaten Eigenschaften (Tabelle OUT_Properties)

F.2. Märkte und Händler

F.2.1. Tabelle OUT_Stat_Dealer

Attribut	Typ	Beschreibung
STD_ID	int	Eindeutiger Identifizierer
Dealer_ID	int	ID des Händlers, über den Informationen abgelegt werden.
CurrentDate	datetime	Datum
CustomerQueue	int	Zukünftig: Anzahl wartender Kunden. Attribut wird gegenwärtig nicht genutzt.
Stock	int	Anzahl Produkte zwischen Fertigstellung und Auslieferung an den Kunden
SalesStock	int	Anzahl gefertigter Produkte des Händlers, die noch nicht an einen Kunden vermittelt werden konnten.
SalesStockAdd	int	Anzahl Produkte, die zum »SalesStock« hinzukommen.
SalesStockSub	int	Anzahl Produkte, die aus dem »SalesStock« an einen Kunden »vermittelt« wurden.
AlterableOrders	int	Anzahl Aufträge, die noch an Kundenwünsche angepasst werden können.
NonAlterableOrders	int	Anzahl Aufträge, die nicht mehr an einen Kundenwunsch angepasst werden können.
Production	int	Anzahl Aufträge des Händlers im Bestand eines oder mehrerer Werke zwischen Fixierung der Aufträge und Fertigstellung der Produkte
Distribution	int	Anzahl Produkte zwischen Fertigstellung und Auslieferung an den Abnehmer
SuccessfulManipulations	int	Anzahl erfolgreicher Anpassungen eines Händlernauftrags an einen Kundenwunsch an diesem Tag
SuccessfulManipulationsOv	int	Anzahl erfolgreicher Anpassungen eines Händlernauftrags an einen Kundenwunsch insgesamt
ProductManipulations	int	Anzahl versuchter Anpassungen eines Händlernauftrags an einen Kundenwunsch an diesem Tag
ProductManipulationsOv	int	Anzahl versuchter Anpassungen eines Händlernauftrags an einen Kundenwunsch insgesamt
ProductsTested	int	Anzahl Aufträge, die den Kunden an diesem Tag zur Auswahl angeboten wurden.
ProductsTestedOv	int	Anzahl Aufträge, die den Kunden insgesamt zur Auswahl angeboten wurden.
CustomersProcessed	int	Anzahl Kunden, die an diesem Tag beim Händler ein Produkt bestellen.
CustomersProcessedOv	int	Anzahl Kunden, die insgesamt beim Händler ein Produkt bestellt haben.
CustomersAcceptingProducts	int	Anzahl Kunden, die an diesem Tag ein Produkt (Auftrag) aus dem Bestand des Händlers akzeptiert haben.

CustomersAcceptingProductsOv	int	Anzahl Kunden, die insgesamt ein Produkt (Auftrag) aus dem Bestand des Händlers akzeptiert haben.
ProductsOrdered	int	Abgegebene Bestellungen des Händlers an diesem Tag
ProductsOrderedOv	int	Abgegebene Bestellungen des Händlers insgesamt
ProductsBooked	int	Anzahl Aufträge des Händlers, die an diesem Tag erfolgreich eingebucht wurden.
ProductsBookedOv	int	Anzahl Aufträge des Händlers, die insgesamt erfolgreich eingebucht wurden.
ProductsRejected	int	Anzahl Aufträge, die an diesem Tag nicht erfolgreich eingebucht werden konnten.
ProductsRejectedOv	int	Anzahl Aufträge, die insgesamt nicht erfolgreich eingebucht werden konnten.

Tabelle F.5.: Ergebnisdaten Händler (Tabelle OUT_Stat_Dealer)

F.2.2. Tabelle OUT_Stat_All

Attribut	Typ	Beschreibung
STA_ID	int	Eindeutiger Identifizierer
CurrentDate	datetime	Aktuelles Datum
SimDay	int	Aktuelles Datum als n-ter Simulationstag
CM_CostBase	float	Grundkosten
CM_CostMaterial	float	Materialkosten
CM_CostProduction	float	Produktionskosten
CM_CostDistribution	float	Distributionskosten
CM_CostSales	float	Vertriebskosten
CM_CostAdmin	float	Verwaltungskosten
CM_CostInvest	float	Investitionen
OBJ_CostMaterial	float	Materialkosten
OBJ_CostMaterialCount	int	Anzahl berücksichtigter Aufträge für die Berechnung der Materialkosten
OBJ_CostProduction	float	Produktionskosten
OBJ_CostProductionCount	int	Anzahl berücksichtigter Aufträge für die Berechnung der Produktionskosten
OBJ_CostDistribution	float	Distributionskosten
OBJ_CostDistributionCount	int	Anzahl berücksichtigter Aufträge für die Berechnung der Distributionskosten
OBJ_CostSales	float	Vertriebskosten
OBJ_CostSalesCount	int	Anzahl berücksichtigter Aufträge für die Berechnung der Vertriebskosten
OBJ_CostAdmin	float	Verwaltungskosten
OBJ_CostAdminCount	int	Anzahl berücksichtigter Aufträge für die Berechnung der Verwaltungskosten
OBJ_Revenue	float	Ertrag
OBJ_RevenueCount	int	Anzahl berücksichtigter Aufträge für die Berechnung des Ertrags
Stock	int	Summe der Anzahlen Produkte zwischen Fertigstellung und Auslieferung an den Kunden über alle Händler
SalesStock	int	Anzahl gefertigter Produkte der Händler, die noch nicht an einen Kunden vermittelt werden konnten. Summe über alle Händler.
SalesStockAdd	int	Anzahl Produkte, die zum »SalesStock« hinzukommen. Summe über alle Händler.
SalesStockSub	int	Anzahl Produkte, die aus dem »SalesStock« an einen Kunden »vermittelt« wurden. Summe über alle Händler.
AlterableOrders	int	Anzahl Aufträge, die noch an Kundenwünsche angepasst werden können. Summe über alle Händler.
NonAlterableOrders	int	Anzahl Aufträge, die nicht mehr an einen Kundenwunsch angepasst werden können. Summe über alle Händler.
Production	int	Anzahl Produkte der Händler im Bestand eines oder mehrerer Werke zwischen Fixierung der Aufträge und Fertigstellung der Produkte. Summe über alle Händler.
Distribution	int	Anzahl Produkte zwischen Fertigstellung und Auslieferung an den Abnehmer. Summe über alle Händler.

eop_m8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) früher als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) fertig gestellt wurden.
eop_m7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) früher als geplant fertig gestellt wurden.
eop_0	int	Anzahl Produkte, die tages- bzw. wochengenau fertig gestellt wurden.
eop_7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) fertig gestellt wurden.
eop_8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) fertig gestellt wurden.
delivery_m8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) früher als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) an einen Kunden ausgeliefert wurden.
delivery_m7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) früher als geplant an einen Kunden ausgeliefert wurden.
delivery_0	int	
delivery_7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant an einen Kunden ausgeliefert wurden.
delivery_8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) an einen Kunden ausgeliefert wurden.
arrival_m8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) früher als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) bei einem Händler angeliefert wurden.
arrival_m7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) früher als geplant bei einem Händler angeliefert wurden.
arrival_0	int	Anzahl Produkte, die tages- bzw. wochengenau bei einem Händler angeliefert wurden.
arrival_7	int	Anzahl Produkte, die zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant bei einem Händler angeliefert wurden.
arrival_8p	int	Anzahl Produkte, die mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) bei einem Händler angeliefert wurden.
PlanRel_m8p	int	Anzahl Produkte, bei denen die Fertigung mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) früher als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) begonnen wurde.
PlanRel_m7	int	Anzahl Produkte, bei denen die Fertigung zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) früher als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) begonnen wurde.
PlanRel_0	int	Anzahl Produkte, bei denen die Fertigung tages- bzw. wochengenau entsprechend der Planung (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) begonnen wurde.
PlanRel_7	int	Anzahl Produkte, bei denen die Fertigung zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) begonnen wurde.
PlanRel_8p	int	Anzahl Aufträge, bei denen die Fertigung zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant (geplant zum Zeitpunkt der ersten Einplanung des Auftrags) begonnen wurde.
PlanRelInter_m8p	int	Anzahl Aufträge, bei denen die Fertigung mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) früher als geplant (geplant zum Zeitpunkt der Festlegung des Werkes, welches den Auftrag fertigen soll) begonnen wurde.

PlanRelInter_m7	int	Anzahl Aufträge, bei denen die Fertigung zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) früher als geplant (geplant zum Zeitpunkt der Festlegung des Werkes, welches den Auftrag fertigen soll) begonnen wurde.
PlanRelInter_0	int	Anzahl Aufträge, bei denen die Fertigung tages- bzw. wochengenau entsprechend der Planung (geplant zum Zeitpunkt der Festlegung des Werkes, welches den Auftrag fertigen soll) begonnen wurde
PlanRelInter_7	int	Anzahl Aufträge, bei denen die Fertigung zwischen einem und sieben Tagen (bei Messung in Wochen bis zu einer Woche) später als geplant (geplant zum Zeitpunkt der Festlegung des Werkes, welches den Auftrag fertigen soll) begonnen wurde.
PlanRelInter_8p	int	Anzahl Aufträge, bei denen die Fertigung mindestens 8 Tage (bei Messung in Wochen mehr als zwei Wochen) später als geplant (geplant zum Zeitpunkt der Festlegung des Werkes, welches den Auftrag fertigen soll) begonnen wurde.

Tabelle F.6.: Ergebnisdaten Zusammenfassung Händler (Tabelle OUT_Stat_All)

F.2.3. Tabelle OUT_PropertyQuota

Attribut	Typ	Beschreibung
QUT_ID	int	Eindeutiger Identifizierer
VOL_ID	int	ID des PropertyQuota Objektes. Klammert n-Zeilen dieser Tabelle zu einem Datensatz.
PropertySpec_ID	int	ID Eigenschaftenspezifikation
PropertyConstraint_ID	int	ID Eigenschaftengruppe (Constraint)
Volume	int	Anzahl Eigenschaften

Tabelle F.7.: Ergebnisdaten Quoten (Tabelle OUT_PropertyQuota)

F.3. Ressourcen

F.3.1. Tabelle OUT_Resource

Attribut	Typ	Beschreibung
RSC_ID	int	Eindeutiger Identifizierer
Resource_ID	int	ID der Ressource
CurrentDate	datetime	Simulationsdatum, für welches die Information gültig ist.
MinCap	float	Untere Grenze der nutzbaren Kapazität. Eine geringere Nutzung unterschreitet die spezifizierten Flexibilitätsgrenzwerte. Eine solche Unterschreitung der verfügbaren Ressourcenkapazität hat keine weiteren Folgen, außer das eine entsprechende Information an dieser Stelle vermerkt wird.
MaxCap	float	Max. innerhalb der spezifizierten Flexibilitätsgrenzen verfügbare Kapazität. Eine Überschreitung dieser Kapazität ist nicht möglich und wird von der Ressource verhindert.
AvailCap	float	Noch verfügbare Kapazität
UsedCap	float	Genutzte Kapazität

Tabelle F.8.: Ergebnisdaten Ressourcen (Tabelle OUT_Resource)

F.4. Prozesse

F.4.1. Tabelle OUT_BO_Process

Attribut	Typ	Beschreibung
PRI_ID	int	Eindeutiger Identifizierer

PRC_ID	int	ID des Prozesses
BO_ID	int	ID des Auftrags bzw. Produkts, welches den Prozess durchlaufen hat.
ProcessEntry	datetime	Zeitpunkt, an dem Auftrag bzw. Produkt in Prozess eingetreten ist.
ProcessExit	datetime	Zeitpunkt, an dem Auftrag bzw. Produkt aus Prozess ausgetreten ist.
Stock	int	Anzahl Produkte bzw. Aufträge, die sich zwischen Prozesseingang und -ausgang befinden.
CapitalCommitment	float	Gebundenes Kapital für den Auftrag
CapitalCommitmentCost	float	Kapitalbindungskosten, die während der Verweildauer des Produkts bzw. Auftrags entstanden sind.
AccountedFrom	datetime	Beginn des Intervalls, für das die Kapitalbindung berechnet wird.
AccountedTo	datetime	Ende des Intervalls, für das die Kapitalbindung berechnet wird.

Tabelle F.9.: Ergebnisdaten Auftragsdurchlauf Prozesse (Tabelle OUT_BO_Process)

F.5. Werke

F.5.1. Tabelle OUT_Plant

Attribut	Typ	Beschreibung
PLR_ID	int	Eindeutiger Identifizierer
Plant_ID	int	ID des Werkes
CurrentDate	datetime	Gegenwärtiges Simulationsdatum
SimDay	int	Gegenwärtiges Simulationsdatum (als n-ter Simulationstag)
PlantAssignmentStartDay	int	Start des Intervalls, innerhalb dessen zuletzt Aufträge an das Werk übergeben wurden.
PlantAssignmentEndDay	int	Ende des Intervalls, innerhalb dessen zuletzt Aufträge an das Werk übergeben wurden.
FreezeStartDay	int	Beginn des Intervalls (als Simulationstag), dessen Aufträge zuletzt eingefroren wurden (bezogen auf geplanten Fertigungsbeginn).
FreezeEndDay	int	Ende des Intervalls (als Simulationstag), dessen Aufträge zuletzt eingefroren wurden (bezogen auf geplanten Fertigungsbeginn).
FirstCapAvail	int	Erster Simulationstag, an dem Fertigungskapazität für die Einplanung von Aufträgen zur Verfügung steht.
OrderQueuePlan	int	Anzahl Aufträge, die in einem werksübergreifenden Auftragspool stehen. Dieser Pool wird nur genutzt, wenn eine Verwirbelungsfunktion für Aufträge vor der Werkszuordnung spezifiziert ist. Befinden sich mehrere Werke in einem Simulationsmodell, wird für alle Werke derselbe Wert ausgewiesen.
OrderQueue	int	Anzahl Aufträge, die im Auftragspool des Werkes stehen.
OrderQueueCap	int	Kapazität, die für die Abarbeitung der Aufträge in der Queue des Werkes zur Verfügung steht.
OrderQueueDays	int	Anzahl Tagesvolumina, die in der Auftragsqueue stehen.
OrderQueueAdd	int	Anzahl Aufträge, die an diesem Tag der Auftragsqueue hinzugefügt wurden.
Capacity	int	Fertigungskapazität an diesem Tag, die von der Ressource zur Verfügung gestellt wurde, vermindert um ein etwaiges Kapazitätslimit, welches an der Programmplanung ausgerichtet wird.
TechCapacity	int	Fertigungskapazität an diesem Tag, die von der Ressource zur Verfügung gestellt wurde (siehe auch Attribut »Capacity«).
CapacityPlan	int	Plankapazität an diesem Tag, die von der Ressource zur Verfügung gestellt wurde.
PlantAssignmentTarget	int	Vorgabe Aufträge, die gemäß Programmplanung an die Werke zu verteilen sind.
PlantAssignmentDeviation	int	Abweichung gegenüber der Vorgabe Aufträge, die zu verteilen ist (siehe Attribut »PlantAssignmentTarget«).
BO_Planned	int	Anzahl Aufträge, die gemäß Planung (Anzahl eingeplanter Aufträge) zu fertigen sind.
BO_ProgrTarget	int	Anzahl Aufträge, die unter Einbeziehung der Anzahl eingeplanter Aufträge und etwaiger Rückstände zu fertigen sind.
BO_ProgrDeviation	int	Abweichung der tatsächlich gefertigten Aufträge gegenüber der Vorgabe aus der Planung

BO_SOP	int	Anzahl Aufträge, deren Fertigung begonnen wurde.
BO_EOP	int	Anzahl Produkte, die am heutigen Tag fertig gestellt wurden.
SumBuildProducts	int	Anzahl Produkte, die insgesamt bislang gefertigt wurden.
BO_Reject	int	Anzahl Aufträge, deren Einplanung versucht wurde, aber aufgrund von Kapazitätsengpässen nicht eingeplant werden konnten.
BO_AffectedByFailure	int	Anzahl Aufträge in der Auftragsqueue des Werkes, die von gegenwärtigen Störungen betroffen sind.
BO_Late	int	Anzahl Aufträge in der Auftragsqueue des Werkes, die gegenüber der ursprünglichen Planung verspätet sind.
BO_AvgDelay	float	Durchschnittliche Verspätung der Aufträge, deren Fertigung an diesem Tag begonnen wurde.
BO_Prefetched	int	Anzahl Aufträge, die an diesem Tag vorgezogen wurden.
BO_PrefetchedOV	int	Anzahl Aufträge, die insgesamt vorgezogen wurden (Summe über BO_Prefetched bislang).
BO_Replanned	int	Anzahl Aufträge, die an diesem Tag umgeplant wurden.
PrefetchSimDay	int	Am weitesten in der Zukunft liegender Tag, von dem Aufträge vorgezogen wurden.
CapacityToReserve	int	Kapazität, die an diesem Tag reserviert werden sollte.
ReservedNew	int	Kapazität, die an diesem Tag erfolgreich reserviert werden konnte. Es wird an in der Zukunft liegenden Produktionstagen Kapazität reserviert, um bestehende Rückstände zu reduzieren.
DayReservedCap	int	Kapazität, die an diesem Tag reserviert wurde, um Rückstände zu reduzieren.
SumReservedCap	int	Summe der in der »Zukunft« reservierten Kapazität. Die reservierte Kapazität, die für die Reduzierung der gegenwärtigen Rückstände vorgesehen ist.
SumReservedCapOV	int	Summe der bislang reservierten Kapazität. Dieser Wert schließt auch Kapazität ein, die an Tagen in der Vergangenheit reserviert wurde.
ActiveFailures	int	Anzahl gegenwärtiger Störungen
PlantStock	int	Anzahl Aufträge zwischen Fertigungsbeginn und Fertigungsende
RestrSupp_ID	int	ID des Lieferanten, aufgrund dessen beschränkter Kapazität die meisten Aufträge nicht eingeplant werden konnten.
BO_SupplBy	int	Anzahl Aufträge in der Auftragsqueue des Werkes, die Eigenschaften enthalten, die vom Lieferanten mit der ID »RestrSupp_ID« hergestellt werden.
RestrSupp_Name	char (255)	Bezeichnung (Name) des Lieferanten mit der ID »RestrSupp_ID«
RestrPropSpec_ID	int	ID der Eigenschaftenspezifikation, die aufgrund fehlender Kapazität zur Nichteinplanung der meisten Aufträge führte. Das heißt, diese Eigenschaftenspez. ist der wichtigste Engpass, weitere Engpässe können vorhanden sein.
BO_PropSpecRestr	int	Anzahl Aufträge in der Auftragsqueue des Werkes, die die Eigenschaftenspez. mit der ID »RestrPropSpec_ID« enthalten.
RestrPropSpec_Name	char (255)	Bezeichnung (Name) der Eigenschaftenspez. mit der ID »RestrPropSpec_ID«
CapitalCommitment	float	Kapitalbindung über alle Aufträge/Produkte, die sich zwischen Fertigungsbeginn und Fertigungsende befinden.
CapitalCommitmentCost	float	Kapitalbindungskosten, die an diesem Tag für alle Aufträge/Produkte entstanden sind, die sich zwischen Fertigungsbeginn und Fertigungsende befinden.
ContainerReceived	int	Anzahl Behälter, die an diesem Tag im Werk eingetroffen sind.
SumContainerReceived	int	Anzahl Behälter, die insgesamt im Werk eingetroffen sind.
ContainerInTransit	int	Anzahl Behälter, die sich »im Zulauf« auf das Werk befinden.

Tabelle F.10.: Ergebnisdaten Werke (Tabelle OUT_Plant)

F.6. Zulieferer

F.6.1. Tabelle OUT_Supplier

Attribut	Typ	Beschreibung
OSP_ID	int	Eindeutiger Identifizierer

SPL_ID	int	ID des Zulieferers
SimDay	int	Aktuelles Datum als n-ter Simulationstag
CurrentDate	datetime	Aktuelles Datum
Stock	int	Bestand Teile, die vom Lieferanten produziert, aber noch nicht im zugeordneten Werk verbaut wurden. Werden Teile vom Lieferanten via Distributionskanal zum Werk transportiert, ist der Wert Null. Analoge Informationen können dann am Attribut »StockDS« gefunden werden.
StockDS	int	Bestand Teile, die vom Lieferanten produziert, aber noch nicht im zugeordneten Werk verbaut wurden. Werden Teile vom Lieferanten nicht via Distributionskanal zum Werk transportiert, ist der Wert Null. Analoge Informationen können dann am Attribut »Stock« gefunden werden.
StockPlan	int	Attribut wird nicht genutzt.
ActualCapacity	int	Noch verfügbare Kapazität an diesem Tag. Die an diesem Tag genutzte Kapazität (ohne Berücksichtigung vorproduzierter Einheiten; siehe auch Erläuterung zu PrefetchTarget) ergibt durch InitialCapacity - ActualCapacity. Teilen sich mehrere Zulieferer eine Ressource, steht ein identischer Wert bei all diesen Zulieferern für das Attribut.
InitialCapacity	int	Verfügbare Kapazität vor der ersten Buchung. Teilen sich mehrere Zulieferer eine Ressource, steht ein identischer Wert bei all diesen Zulieferern für das Attribut.
PrefetchSource	int	Anzahl Teile, die an diesem Tag für andere Tage vorproduziert werden. Die hier verbuchte Kapazität wird von ActualCapacity subtrahiert und steht an diesem Tag nicht mehr zur Verfügung. Teilen sich mehrere Zulieferer eine Ressource, steht ein identischer Wert bei all diesen Zulieferern für das Attribut.
PrefetchTarget	int	Anzahl Teile, die für diesen Tag an anderen Tagen vorproduziert wurden. Die hier verbuchte Kapazität erhöht die Anzahl der an diesem Tag insgesamt verfügbaren Einheiten (InitialCapacity + PrefetchTarget). Teilen sich mehrere Zulieferer eine Ressource, steht ein identischer Wert bei all diesen Zulieferern für das Attribut.
SumInitialCapacity	int	Summe der bis zu diesem Tag zur Verfügung stehenden Kapazität
CallOff	int	An diesem Tag vom Werk benötigte Teile
SumCallOff	int	Summe der bis zu diesem Tag benötigten Teile
PartsSent	int	Anzahl Teile, die an diesem Tag über einen Distributionskanal versandt wurden.
SumPartsSent	int	Summe der Anzahl Teile, die insgesamt über einen Distributionskanal versandt wurden.
PartsDelivered	int	Anzahl Teile, die über einen Distributionskanal versandt wurden und an diesem Tag am Zielpunkt angekommen sind.
SumPartsDelivered	int	Gesamtanzahl Teile, die über einen Distributionskanal versandt wurden und am Zielpunkt angekommen sind.
OA_MaxOutputPerDay	int	Obergrenze Ausbringung an diesem Tag
CapRestriction	char (256)	Kapazitätsrestriktion (z.B. gegenwärtige Störung)
ContainerWaitingForShipment	int	Anzahl Behälter, die im Ausgangspuffer auf einen Abtransport warten.
ContainerFillTime	int	Zeit, die benötigt wird, so viele Teile zu produzieren, dass ein Behälter befüllt werden kann.
CapacityChange	float	Kapazitätsveränderung durch eine Störung. Es wird die verbleibende Kapazität angezeigt.
FailureCount	int	Anzahl der Störungen, die bislang aufgetreten sind.

Tabelle F.11.: Ergebnisdaten Zulieferer (Tabelle OUT_Supplier)

F.6.2. Tabelle OUT_Parts

Attribut	Typ	Beschreibung
PTS_ID	int	Eindeutiger Identifizierer
PART_ID	int	ID des Teiles
BO_ID	int	ID des Auftrags, für den das Teil benötigt wird.
Count	int	Anzahl der Teile, die benötigt werden.

Tabelle F.12.: Ergebnisdaten Teile (Tabelle OUT_Parts)

F.7. Distribution

F.7.1. Tabelle OUT_Transports

Attribut	Typ	Beschreibung
TRT_ID	int	Eindeutiger Identifizierer
DCL_ID	int	ID des Distributionskanals
TTY_ID	int	ID der Transporttypen
LSP_ID	int	ID des Logistikdienstleisters
TRP_ID	int	ID des zugehörigen Transportplans
RTI_ID	int	ID der Routeninformation
LSC_ID	int	ID der Routenplanung des Logistikdienstleisters
Distance	float	Distanz, über die transportiert wurde.
TransportMode	int	Transportmodus. Beladen (=2), Leerfahrt zum Kunden (=1), Leerfahrt zurück zum Logistikdienstleister(=3)
LocationStart_ID	int	Startpunkt des Transportes
LocationEnd_ID	int	Endpunkt des Transportes
Depart	datetime	Zeitpunkt, an dem der Transport startet
Arrival	datetime	Zeitpunkt, an dem der Transport das Ziel erreicht
LeadtimeTransport	int	Benötigte Transportzeit
LeadtimeUnit	int	Einheit, in der Durchlaufzeit angegeben wird.
BO	int	Anzahl transportierter Einheiten (Produkte oder Behälter)
Capacity	int	Transportkapazität
CapacityM3	float	Kapazität in Kubikmetern
REMCapacityM3	float	Verbleibende Kapazität
Cost	float	Summe aller Kosten (s.u.)
OBJ_CostMaterial	float	Materialkosten (Einzelkosten)
OBJ_CostProduction	float	Produktionskosten (Einzelkosten)
OBJ_CostDistribution	float	Distributionskosten (Einzelkosten)
OBJ_CostSales	float	Vertriebskosten (Einzelkosten)
OBJ_CostAdmin	float	Administrative Kosten (Einzelkosten)
OBJ_Revenue	float	Ertrag für diesen Transport
BOInBuffer	int	Anzahl Produkte, die im Puffer stehen, nachdem der Transport vollständig beladen wurde.
WaitingForCust	int	Anzahl Produkte, die erst weitertransportiert werden, wenn ein Abnehmer (Kunde) gefunden wurde.

Tabelle F.13.: Ergebnisdaten Transporte (Tabelle OUT_Transports)

F.7.2. Tabelle OUT_BO_Transport

Attribut	Typ	Beschreibung
BTR_ID	int	Eindeutiger Identifizierer
TRT_ID	int	ID des Transports, auf dem Produkt bzw. Behälter transportiert wurde.
DCL_ID	int	ID des Distributionskanals, auf dem Transport durchgeführt wurde.
BO_ID	int	ID des transportierten Produkts. NULL, wenn Behälter transportiert wird.
CNT_ID	int	ID des transportierten Behälters. NULL, wenn ein Produkt transportiert wird.
CNT_TYPE_ID	int	Behältertyp. NULL, wenn ein Produkt transportiert wird.
CNT_Parts	int	Anzahl Teile im Behälter
RTI_ID	int	ID der Routeninformation
EnteringBuffer	datetime	Wann ist Produkt in den Eingangspuffer desjenigen Distributionskanals eingetreten, innerhalb dessen der gegenwärtige Transport stattgefunden hat (siehe DCL_ID).
LeavingBuffer	datetime	Wann hat Produkt den Eingangspuffer desjenigen Distributionskanals verlassen, innerhalb dessen der gegenwärtige Transport stattgefunden hat (siehe DCL_ID).
StartTransport	datetime	Zeitpunkt, an dem Transport gestartet ist.
EndTransport	datetime	Zeitpunkt, an dem der Transport das Ziel erreicht.
StartDelay	datetime	Zu welchem Zeitpunkt hat die Wartezeit im Puffer begonnen.

EndDelay	datetime	Zu welchem Zeitpunkt endete die Wartezeit im Puffer.
LeadtimeUnit	int	Einheit, in der die Durchlaufzeit angegeben wird.
BufferTime	int	Zeit, die das Produkt im Puffer stand (Anzahl Minuten).
WaitingTime	int	Wartezeit auf den Transport
TransportTime	int	Zeit, die für den Transport benötigt wurde.
TransportCount	int	Der wievielte Transport findet gegenwärtig statt (inklusive dieses Transportes)?
DelayTime	int	Wartezeit im Puffer, die aufgrund der spezifizierten Wartezeit entstanden ist (kleiner gleich Attribut »BufferTime«).
CapitalCommitment	float	Gebundenes Kapital (nur Produkte)
CapitalCommitmentCost	float	Kapitalbindungskosten, die während des Transportes angefallen sind.
AccountedFrom	datetime	Zeitpunkt, von dem die Kapitalbindungskosten berechnet werden.
AccountedTo	datetime	Zeitpunkt, bis zu dem die Kapitalbindungskosten berechnet werden.

Tabelle F.14.: Ergebnisdaten Produkte auf Transporten (Tabelle OUT_BO_Transport)

F.7.3. Tabelle OUT_Stat_Distribution

Attribut	Typ	Beschreibung
OSD_ID	int	Eindeutiger Identifizierer
DC_ID	int	ID des Distributionskanals
CurrentDate	datetime	Gegenwärtiges Datum
SimDay	int	Gegenwärtiger Simulationstag
EntryBuffer	int	Anzahl Produkte bzw. Behälter in Eingangspuffer des Distributionskanals
WaitingForTransport	int	Anzahl Produkte bzw. Behälter, die auf einen Transport warten.
OnTransport	int	Anzahl Produkte bzw. Behälter, die sich derzeit auf einem Transport befinden.
ExitBuffer	int	Anzahl Produkte bzw. Behälter, die sich im Ausgangspuffer befinden.
CapitalCommitment	float	Summe des gebundenen Kapitals für alle Produkte, die sich im Distributionskanal befinden.
CapitalCommitmentCost	float	Kapitalbindungskosten, die an diesem Tag im Distributionskanal angefallen sind.

Tabelle F.15.: Ergebnisdaten Zusammenfassung Distribution (Tabelle OUT_Stat_Distribution)

G. Anhang – Auftragsortierung

Unten stehend ein Beispiel für eine Vergleichsfunktion, die eine Sortierung der Aufträge im Werk herbeiführt. Die Grammatik unter der Vergleichsfunktion beinhaltet die Regeln, die für die Formulierung der Vergleichsfunktionen zu beachten sind.

G.1. Vergleichsfunktion

```
prod1Day = @[0].Production_Start_Day + @[0].Delay / 1440;
prod2Day = @[1].Production_Start_Day + @[1].Delay / 1440;
hasCustomer1 = @[0].HasCustomer;
hasCustomer2 = @[1].HasCustomer;
repl1 = @[0].Replanned;
repl2 = @[1].Replanned;

if(prod1Day < prod2Day ; 1 ;
if(prod1Day > prod2Day; 0 ;
if(repl1 > repl2; 1 ;
if(hasCustomer1 and not hasCustomer2 ; 1 ;
if(hasCustomer2 and not hasCustomer1 ; 0 ;
if(@[0].SerialNo < @[1].SerialNo; 1 ; 0)))))) ;
```

G.2. Grammatik

```
\%token <floatval>FLOATNUM /* Simple double precision number */
\%token <intval>INTNUM /* Simple int number */
\%token <tptr>VAR /* Variable*/
\%token <tptr>IVAR /* Typed Variable holding integer values */
\%token <tptr>FVAR /* Typed Variable holding float values */
\%token <tptr>STRING /* String */

\%token SIN, COS, TAN, ATAN, LOG, LOG10, SQRT, ABS
\%token SELECTOR\_PRODUCTION\_START\_DAY, SELECTOR\_REPLANNED, SELECTOR\_SERIALNO,
\%token SELECTOR\_DELAY, SELECTOR\_PRODUCTION\_START\_MIN, SELECTOR\_PRODUCTION\_START\_TIME
\%token SELECTOR\_DEALER\_ID, SELECTOR\_MARKET\_ID, SELECTOR\_CLASS\_ID, SELECTOR\_HAS\_PROPERTY\_KEY
\%token SELECTOR\_HAS\_CUSTOMER
\%token IF\_EXPR, SM\_EQ, BG\_EQ, EQUAL
\%token SIMDAY INTCAST FLOATCAST
\%token OR AND XOR NOT
\%token PATTERN\_PREFIX

\%type <floatval> exp
\%type <intval> iexp
\%type <tptr> STRING

\%right '='
\%left '-' '+' OR XOR
\%left '*' '/' AND
\%left NEG NOT /* Negation--unary minus */
\%right '^' /* Exponentiation */
```

```

/* Grammar follows */

\%\%
input: /* empty */
| input line
;

line:
';'
| exp ';'
| iexp ';'
| error ';'
;

operand:
| '@' '[' iexp ']'
| '@'
;

exp: FLOATNUM
| FVAR
| VAR '=' exp
| FVAR '=' exp
| FLOATCAST iexp

// Functions
| SIN '(' exp ')''
| COS '(' exp ')''
| TAN '(' exp ')''
| ATAN '(' exp ')''
| LOG10 '(' exp ')''
| LOG '(' exp ')''
| SQRT '(' exp ')''
| ABS '(' exp ')''

| exp '+' exp
| exp '-' exp
| exp '*' exp
| exp '/' exp
| '-' exp \%prec NEG
| exp '^' exp
| '(' exp ')''
| exp ',' exp
| exp '<' exp
| exp SM\_EQ exp
| exp '>' exp
| exp BG\_EQ exp
| exp EQUAL exp
;

iexp: INTNUM
| IVAR
| VAR '=' iexp
| IVAR '=' iexp
| IF\_EXPR '(' iexp ';' iexp ';' iexp ')''
| SIMDAY
| iexp OR iexp
| iexp AND iexp
| iexp XOR iexp
| NOT iexp \%prec NEG
| operand SELECTOR\_HAS\_CUSTOMER
| operand SELECTOR\_CUSTOMER\_ID
| operand SELECTOR\_PLANNED\_EOP
| operand SELECTOR\_PRODUCTION\_START\_DAY
| operand SELECTOR\_PRODUCTION\_START\_MIN
| operand SELECTOR\_PRODUCTION\_START\_TIME

```

```

| operand SELECTOR\_SERIALNO
  | operand SELECTOR\_REPLANNED
| operand SELECTOR\_DELAY
| operand SELECTOR\_DEALER\_ID
| operand SELECTOR\_MARKET\_ID
| operand SELECTOR\_CLASS\_ID
| operand SELECTOR\_HAS\_PROPERTY\_KEY '(' STRING ')
| ABS '(' iexp ')
| INTCAST exp
| iexp '+' iexp
| iexp '-' iexp
| iexp '*' iexp
| iexp '/' iexp
| '-' iexp \%prec NEG
| iexp '^' iexp
| '(' iexp ')
| iexp ',' iexp
| iexp '<' iexp
| iexp SM\_EQ iexp
| iexp '>' iexp
| iexp BG\_EQ iexp
| iexp EQUAL iexp
;

/* End of grammar */
\%\%

```

In unten stehender Tabelle werden die syntaktischen Elemente aufgezählt, die in der oben stehenden Grammatik als »Token« verwendet werden. Das Format ist angelehnt an das des Werkzeugs `Flex`, mit welchem die lexikalische Analyse für den hier geschilderten Anwendungsfall realisiert wurde.

```

ws [ \t]+
cr [\n\r]+
alpha [A-Za-z]
dig [0-9]
name ({alpha}|{dig}|\$)({alpha}|{dig}|[_.\-/$])*
string '{name}'
num1 {dig}+\.{([eE][+]?{dig})?
num2 {dig}*\.{([eE][+]?{dig})?
floatnumber {num1}|{num2}
intnumber {dig}+

%%

{ws} {;}

{cr} {
++yyloc.last_line;
yyloc.last_column = 0;
GL_ScriptArgs.m_LineNo++;
}

{floatnumber} {return FLOATNUM;}

{intnumber} {return INTNUM;}

"if" {return IF_EXPR; }
"<=" {return SM_EQ; }
"=<" {return SM_EQ; }
">=" {return BG_EQ; }
"=>" {return BG_EQ; }
"==" {return EQUAL; }
"sin" {return SIN; }
"cos" {return COS; }

```

```
"tan" {return TAN; }
"atan" {return ATAN; }
"sqrt" {return SQRT; }
"log" {return LOG; }
"log10" {return LOG10; }
"abs" {return ABS; }
"or" {return OR; }
"and" {return AND; }
"xor" {return XOR; }
"not" {return NOT; }

".HasCustomer" {return SELECTOR_HAS_CUSTOMER;}
".Production_Start_Day" {return SELECTOR_PRODUCTION_START_DAY;}
".Production_Start_Min" {return SELECTOR_PRODUCTION_START_MIN;}
".Production_Start_Time" {return SELECTOR_PRODUCTION_START_TIME;}
".Replanned" {return SELECTOR_REPLANNED;}
".SerialNo" {return SELECTOR_SERIALNO;}
".Delay" {return SELECTOR_DELAY;}
".Dealer_ID" {return SELECTOR_DEALER_ID;}
".Customer_ID" {return SELECTOR_CUSTOMER_ID;}
".Market_ID" {return SELECTOR_MARKET_ID;}
".Class_ID" {return SELECTOR_CLASS_ID;}
".Has_Property_Key" {return SELECTOR_HAS_PROPERTY_KEY;}
".PlannedEOP" {return SELECTOR_PLANNED_EOP;}

"SimDay" {return SIMDAY;}
{string} {return STRING;}
(float) {return FLOATCAST;}
(int) {return INTCAST;}

{name} {return IVAR; oder return FVAR; oder return VAR;}
. {return yytext[0];}

%%
```

G.3. Softwareauswahl nach VDI 3633 Blatt vier

Leistungsbereich	Bewertungskriterien	Bewertung <i>B</i>	Gewichtung <i>G</i>	Gesamt $B \times G$
Systemumgebung	Erfahrung des Vertriebspartners			
	Marktpräsenz und Referenzen			
	Anwendungsunterstützung / Hotline			
	Hardwareanforderungen			
	Gängiges Betriebssystem			
	Qualifikationsanforderungen an den Anwender			
	Preis des kompletten Arbeitsplatzes			
	Wartungskosten			
Softwareleistung	Empfohlener Einsatzbereich ähnlich der Problemstellung			
	Anzeige zeitorientierter Zustandsänderungen			
	Modellgröße			
	Dimensionierung von Variablen			
	Alle existierenden Materialflusssysteme als vorkonfigurierte Systembausteine vorhanden			
	Programmieraufwand			
	Systemhandhabung/Anwenderfreundlichkeit			
	Modell-Editor			
	Fehlererkennung			
	Programmierhilfen			
	Debugger			
	Darstellung und Animation des Modells			
	Maßstäblichkeit der Darstellung			
	Syntaxkontrolle und Konsistenzprüfung			
	Darstellungsmöglichkeiten der Ergebnisse			
Ausgabemöglichkeiten der Ergebnisse an weitere Ergebnisse				
	CAD-Schnittstellen			
Sonstiges	Weiterentwicklung des Systems gewährleistet			
	Informationsaustausch mit weiteren Anwendern			
	Produktinformation des Entwicklers			
	Integration in die vorhandene EDV-Landschaft sinnvoll			
	Simulationsgeschwindigkeit akzeptabel			
	Gesamteindruck des Simulationssystems			

Tabelle G.2.: Auswahl von Simulationswerkzeugen nach VDI 3633 am Beispiel Automobilindustrie (Quelle: [VDI97b, S. 11])