



Thema:

**Konzeption und Entwicklung eines Workflowmanagementsystems für
Zwischen- und Endbevorratungen von Ersatzteilen in der Automobilindustrie**

Diplomarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Andreas Maunz, DaimlerChrysler AG GLC Germersheim

Betreuer: Prof. Dr. Hans-Knud Arndt

Vorgelegt von: André Bögelsack

Abgabetermin: 09.04.06

Inhaltsverzeichnis

Inhaltsverzeichnis	III
Verzeichnis der Abkürzungen und Akronyme.....	VII
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	IX
1 Globale Ersatzteilversorgung bei DaimlerChrysler	1
1.1 Die Rolle des Global Logistic Center	1
1.2 End- und Zwischenbevorratungen	2
1.3 Motivation zur Einführung eines Workflowmanagementsystems.....	2
1.4 Aufbau und Ziele dieser Arbeit.....	3
2 Theoretische Grundlagen.....	4
2.1 Betrachtung von Prozessen und Geschäftsprozessen	4
2.1.1 Der Prozess	4
2.1.2 Geschäftsprozesse	6
2.1.3 Kernkompetenz als Untermenge von Geschäftsprozessen.....	7
2.2 Theoretische Betrachtung eines Workflows.....	8
2.2.1 Workflowdefinition der Workflow Management Coalition.....	8
2.2.2 Vorgehen zur Bestimmung eines Workflows laut Object Management Group ..	10
2.2.3 Von der Prozessdefinition zum Workflow	11
2.2.4 Workflow-Typisierung	12
2.3 Theoretische Betrachtung eines Workflowmanagementsystems	14
2.3.1 Definition eines Workflowmanagementsystems	14
2.3.2 Anforderungen an ein Workflowmanagementsystems	15
2.3.3 Adaption eines Workflowmanagementsystems	16
2.4 Nutzung der Prozessmodellierung in der Softwareentwicklung.....	17
2.4.1 Die Sieben-Phasen-Softwareentwicklung.....	17
2.4.2 Ziele der Prozessmodellierung	19
2.4.3 Allgemeine Prozessmodellierung in den Softwareentwicklungsphasen	19
2.5 Grundlegendes Verständnis zur Teileversorgung.....	20
2.5.1 Lebenszyklus eines Teils im Teilewesen.....	20
2.5.2 Regeldisposition und Bevorratungsaktionen allgemein	22

2.5.3	Zwischen- und Endbevorratungen.....	22
2.6	Gründe für Zwischenbevorratungen und Endbevorratungen.....	23
2.6.1	Zusammenarbeit zwischen Lieferanten und DaimlerChrysler	23
2.6.2	Grund 1: Lieferanteninsolvenz.....	24
2.6.3	Grund 2: Bauteilabkündigung vom Vorlieferanten	25
2.6.4	Grund 3: Auslauf der vertraglichen Versorgungsdauer	26
2.6.5	Grund 4: Wirtschaftliche Aspekte – Auslaufmanagement	26
2.6.6	Grund 5: Verlagerungsmanagement	27
2.6.7	Grund 6: Beschaffungskostenoptimierung.....	28
2.6.8	Weitere Gründe	29
2.7	Kapitelabschluss.....	30
3	Beschreibung des Projektes “Integrated Process for Optimized Stocking”	31
3.1	Allgemeine Einführung.....	31
3.2	Ziele des Projektes	31
3.3	Strukturierung des Projektes.....	32
3.4	Vorgängerprojekte	34
3.5	Zusammenfassung	35
4	Problemanalyse und Anforderungsanalyse	36
4.1	Problem der Mengenbestimmung.....	36
4.2	Problem der Kapitalbindung und Flächenproblematik	37
4.3	Problem des Schrottpotenzials	38
4.4	Allgemeine Probleme.....	39
4.5	Anforderungen an das zukünftige Softwaresystem	40
4.5.1	Theoretische Anforderungen	40
4.5.2	Funktionale Anforderungen	41
4.5.3	Informelle Anforderungen	42
4.5.4	Anforderungen an die Benutzerfreundlichkeit.....	43
4.6	Abstraktion der Anforderungen	44
4.7	Zusammenfassung	45
5	Betrachtung von Lösungsmöglichkeiten	46
5.1	Lösung ohne Softwaresystem.....	46
5.1.1.1	Grundcharakter.....	46

5.1.1.2	Vorteile	46
5.1.1.3	Nachteile.....	47
5.2	Einsatz eines kommerziellen Softwaresystems	47
5.2.1.1	Grundcharakter.....	47
5.2.1.2	Vorteile	48
5.2.1.3	Nachteile.....	48
5.3	Einsatz einer Individualsoftware	49
5.3.1.1	Grundcharakter.....	49
5.3.1.2	Vorteile	49
5.3.1.3	Nachteile.....	50
5.3.1.4	Erkennung der Softwarephasen im Projekt	50
5.4	Kapitelabschluss	51
6	Entwicklung einer konzeptionellen Applikationsschicht	52
6.1	Referenzen der Workflow Management Coalition	52
6.1.1	Referenzarchitektur	52
6.1.1.1	Vorteile	54
6.1.1.2	Nachteil.....	54
6.1.2	Workflow Management Application Programming Interface	55
6.2	Einsatz von Patterns	56
6.2.1	Der Ansatz von Pattern.....	56
6.2.1.1	Vorteile	57
6.2.1.2	Nachteile.....	57
6.2.2	Zwei Patterns zur Erstellung eines Workflowmanagementsystems	58
6.2.2.1	„A Pattern for Managing Distributed Workflows“	58
6.2.2.2	„A Pattern Language for Workflow Systems“	60
6.2.3	Einordnung der beiden Patterns in die Referenzarchitektur der WFMC.....	62
6.3	Erstellung und Überführung des Soll-Prozesses	63
6.3.1	Gründe, Ziele und Vorgehen zur Soll-Prozesserstellung.....	63
6.3.2	Segment 1: Datenkollektion.....	64
6.3.3	Segment 2: Prognoseerstellung – Schleifenbeginn.....	67
6.3.4	Segment 3: EBV-Workflow und Einkauf	69
6.3.5	Segment 4: Classic Center, Wirtschaftlichkeitsrechnung und Eskalation.....	71
6.3.6	Segment 5: Werksplanung, Ende der Schleife und des Prozesses	73
6.3.7	Extraktion der Stationen – Schnittstellendefinition.....	78
6.3.7.1	Station 1 – „Daten in das System einspielen“	78

6.3.7.2	Station 2 – „Prognose erstellen“	79
6.3.7.3	Station 3 – „JDAX“	79
6.3.7.4	Station 4 – „Einkauf“	79
6.3.7.5	Station 5 – „Classic Center und Fragenkatalog“	79
6.3.7.6	Station 6 – „WiRe“	80
6.3.7.7	Station 7 – „Eskalation“	80
6.3.7.8	Station 8 – „Werksplanung“	80
6.4	Konzeption des Workflow Enactment Services	81
6.4.1	Das Konstrukt „Teilenummer-Workflow-Station-Prozess-Aktivität“	81
6.4.2	Entwicklung der Workflow Engine	83
6.4.3	Entwicklung des Workflow Enactment Services	84
6.4.4	Workflow Application Programming Interface	86
6.5	Zusätzliche Komponenten im WFMS	87
6.5.1	Der Benutzer	87
6.5.2	Das Portal	88
6.5.3	Der MappingManager	88
6.5.4	Stationskomponenten	89
6.6	Überführung der Komponenten in Klassen	90
6.6.1	Erfassen der Kardinalitäten und Komponentenbeziehungen	91
6.6.2	Überführung der Beziehungen	94
6.6.3	Funktionen der Klassen in der Übersicht	96
6.7	Betrachtung der typischen Anwendungsfälle – die Interfaces der WFMC	99
6.7.1	Neue Prozesses	99
6.7.2	Interaktion mit anderen Workflow Enactment Services	100
6.7.3	Interaktion mit anderen Applikationen und dem Benutzer	100
6.7.4	Administration und Monitoring	100
6.8	Zusammenfassung	101
7	Zusammenfassung und Ausblick	102
7.1	Zusammenfassung	102
7.2	Ausblick	102
	Literaturverzeichnis	104

Verzeichnis der Abkürzungen und Akronyme

BKO	Beschaffungskostenoptimierung
BT-Abkündigung	Bauteilabkündigung
BV-Aktion	Bevorratungsaktion
BV-Menge	Bevorratungsmenge
CLR	Common Language Runtime
CSCW	Computer Supported Cooperative Work
DCAG	DaimlerChrysler AG
DLL	Dynamic Link Library
DuB	Disposition und Beschaffung
EBV	Endbevorratung
EE-Komponenten	Elektronik-Komponenten
eEPK	Erweiterte Ereignisgesteuerte Prozesskette
GLC	Global Logistic Center
GuK	Garantie und Kulanz
IMDS	Inventory Management Documentation System
IPOS	Integrated Process of Optimized Stocking
JDAX	Javadataänderung auf UNIX
KW	Kapitalwert
LDAP	Lightweight Directory Access Protocoll
OMG	Object Management Group
SOAP	Simple Object Access Protocol
WAPI	Workflow Application Programming Interface
WfMC	Workflow Management Coalition
WFMS	Workflowmanagementsystem
WiRe	Wirtschaftlichkeitsrechnung
XML	Extensible Markup Language
XPDL	XML Process Definition Language
ZBV	Zwischenbevorratung

Abbildungsverzeichnis

Abbildung 2-1: Prozessvorgangsmodell.....	5
Abbildung 2-2: Einordnung der Kernkompetenz im Prozessumfeld.....	7
Abbildung 2-3: Workflowdimensionen.....	10
Abbildung 2-4: Workflow-Typisierung [HE03].....	13
Abbildung 2-5: Typische Lebenszykluskurve eines Teils [Gaschler].....	21
Abbildung 2-6: Lieferantenkette.....	25
Abbildung 2-7: Matrix Muss-Kann-EBV/ZBV.....	29
Abbildung 6-1: Schematische Darstellung der Referenzarchitektur [WfMC97].....	53
Abbildung 6-2: Klassendiagramm nach [Seu00].....	58
Abbildung 6-3: Komponentenhierarchie.....	59
Abbildung 6-4: Single central workflowproduct router, nach [Mes97].....	61
Abbildung 6-5: Segment 1: Datenkollektion.....	65
Abbildung 6-6: Einstellen des Sortiments.....	66
Abbildung 6-7: Abgleich Sortimentsdaten.....	66
Abbildung 6-8: Segment 2: Prognoseerstellung.....	68
Abbildung 6-9: Tätigkeiten zur Veränderung der Prognose.....	69
Abbildung 6-10: Segment 3-1: EBV-Workflow.....	70
Abbildung 6-11: Segment 3-2: Suche nach alternativem Lieferanten.....	70
Abbildung 6-12: Segment 4: Classic Center, WiRe und Eskalation.....	72
Abbildung 6-13: Segment 5: Werksplanung und Ende des Prozesses.....	75
Abbildung 6-14: Überblick zum gesamten Soll-Prozess.....	77
Abbildung 6-15: Konstrukt Teilenummer-Workflow-Station-Prozess-Aktivität.....	82
Abbildung 6-16: Workflow Engine des Systems IPOS.....	84
Abbildung 6-17: Workflow Enactment Service und Workflow Engine auf einen Blick.....	85
Abbildung 6-18: Umgebene WAPI.....	86
Abbildung 6-19: Workflow Enactment Service mit umgebener WAPI und zusätzlichen Komponenten.....	90
Abbildung 6-20: n:m-Beziehung in Krähenfußnotation.....	91
Abbildung 6-21: Beziehungen und Kardinalitäten zwischen Komponenten.....	92
Abbildung 6-22: Klassendiagramm.....	95
Abbildung 6-23: Legende zum Diagramm.....	97
Abbildung 6-24: Gemeinsames Klassen -, Sequenz- und Aktivitätsdiagramm.....	98

Tabellenverzeichnis

Tabelle 2-1: Unterschiede EBV - ZBV	23
Tabelle 2-2: Szenarien bei Verlagerungen	28
Tabelle 6-1: Übersicht zu den Funktionen der Klassen.....	96

1 Globale Ersatzteilversorgung bei DaimlerChrysler

Im After-sales-Bereich der DaimlerChryslerAG, kurz DCAG, existiert ein umfangreiches Logistik-Netzwerk, welches die Verfügbarkeit der Ersatzteile sicherstellt. Um einen kurzen Einblick in den After-sales Logistik-Bereich zu geben, wird in dem kommenden Unterabschnitt diese Logistikkette beschrieben und auf die Rolle des Global Logistic Center in Germersheim eingegangen. Der Vorgang einer End- bzw. Zwischenbevorratung soll danach eine erste Erwähnung finden, um die Anforderungen eines gesteuerten Prozesses durch ein Softwareprodukt in diesem Falle kurz zu verdeutlichen. Den Abschluss dieses Abschnittes wird die Erläuterung zu dem Vorgehen bei der Entwicklung dieses Softwareproduktes machen und somit den Aufbau der Arbeit genauer aufzeigen.

1.1 Die Rolle des Global Logistic Center

Das Global Logistic Center, kurz GLC, bildet den Ausgangspunkt der Logistikkette und wird von Werken der Mercedes Benz Car Group und internen, als auch externen, Lieferanten beliefert. Das GLC gibt diese Lieferteile bei Anforderung weiter an den Großhandel, welcher durch Logistikcenter in Deutschland und Europa verkörpert ist. Sie dienen dabei als Logistik-Knotenpunkte. Die Logistikcenter wiederum dienen als Bezugsquellen für den Einzelhandel, welcher als Retailer bezeichnet wird, der als Anlaufstelle und Bezugsquelle für den Endkunden gilt. Auf diese Weise ergibt sich eine 3-stufige Logistikkette aus GLC-LC-Retail, die sich weltumfassend aufzieht.

Das GLC in Germersheim regelt die Ersatzteilversorgung für den gesamten After-sale-Bereich des DaimlerChrysler-Konzerns, ausgenommen aller Chrysler-Komponenten, welche lokal in Nordamerika durch ein Zentrallager vertrieben werden. Um die Größe des GLCs an dieser Stelle noch einmal hervorzuheben, seien ein paar Zahlen zur Orientierung genannt [Ku01]:

- 2,9 Mrd. Euro Bestandwert
- 7,8 Mrd. Euro Brutto-Umsatz im Jahr
- 383.000 gelagerte Sachnummern
- 10.800 m³ Transportvolumen pro Tag

Das GLC ist somit die zentrale Anlaufstelle für alle Ersatzteile im PKW-Bereich, als auch im Nutzfahrzeugbereich. Um eine möglichst hohe Verfügbarkeit aller Teile zu gewährleisten, werden immer wieder Verbesserungsmaßnahmen vollzogen, die sich mit der Menge aller Prozesse, welche zur Lagerhaltung gehören, befassen. Mit einer dieser Verbesserungsmaßnahmen wird sich im Rahmen dieser Diplomarbeit beschäftigt.

1.2 End- und Zwischenbevorratungen

Eine Endbevorratung, kurz EBV, ist ein Vorgang, bei dem ein Ersatzteil auf eine Dauer von 15 bis 30 Jahren im GLC bevorratet wird und so dauerhaft gelagert wird. Dieser Vorgang kann durch unterschiedliche Ursachen ausgelöst werden.

Zwischenbevorratungen, kurz ZBV, sind im Wesentlichen dem Vorgang einer EBV gleich, unterscheiden sich aber hauptsächlich durch den kürzeren zeitlichen Rahmen. Daher werden beide Begriffe aggregiert als EBV/ZBV verwendet. Treten in einem Kontext Unterschiede zwischen beiden Bevorratungsarten auf, so werden diese in der Arbeit explizit erwähnt und ausgeführt.

Im Zentrum der Betrachtung einer EBV/ZBV stehen der Disponent, die Teile und die Bevorratungsmenge. Der Disponent ist für die Bevorratung verantwortlich und muss, um zu einer optimalen Bevorratungsmenge zu kommen, viele unterschiedliche interne und externe Abteilungen des GLC kontaktieren. Dieser Vorgang soll nicht automatisiert werden, sondern mit einem entsprechenden Tool zur Unterstützung des Disponenten vereinfacht werden.

1.3 Motivation zur Einführung eines Workflowmanagementsystems

Der Vorgang einer EBV/ZBV beruht auf einer engen Zusammenarbeit zwischen Disponenten und anderen Abteilungen. Diese Zusammenarbeit wird allerdings von jedem Disponenten anders durchgeführt, sodass sich hier Unstimmigkeiten zwischen den einzelnen Arbeitsschritten bei verschiedenen Disponenten ergeben.

Eine Einführung eines Workflowmanagementsystems kann hier die Prozesse standardisieren und die Kommunikation zwischen Disponenten und den beteiligten Abteilungen fördern. Es werden sich darüber hinaus Synergieeffekte zwischen der Arbeit der einzelnen Disponenten erwartet, als auch eine höhere Effektivität bei der Abwicklung eines EBV/ZBV-Vorganges.

Grundlegend werden sich alle EBV/ZBV-Vorgänge weg von dezentralen Prozessen mit unkoordinierten Kommunikationen entfernen, hin zu einem Gesamtprozess, der sich in einem übergreifenden, mit externen Komponenten kommunizierenden, System bewegt. Somit lassen sich auf eine einfache Art und Weise Schnittstellen zu anderen Systemen finden, mit dem Ziel einen Informationsverlust zu vermeiden. Das zukünftige System trägt den Namen „IPOS“, was für „Integrated Process for Optimized Stocking“ steht. Die Namensgebung erklärt schon das Hauptziel des Systems – die Schaffung eines integrierten Prozesses zur optimalen Lagerhaltung.

1.4 Aufbau und Ziele dieser Arbeit

In der Arbeit wird zuerst eine Einführung zu den theoretischen Grundlagen (siehe Abschnitt 2) gegeben. Danach erfolgt eine detaillierte Beschreibung zu dem Projekt, in dessen Umfeld diese Arbeit entstanden ist (siehe Abschnitt 3). Im darauf folgenden Abschnitt wird der Autor eine Problemanalyse und eine Anforderungsanalyse vor dem Hintergrund eines neu zu konzipierenden Softwaresystems durchführen (siehe Abschnitt 4). Auf diesem Abschnitt aufbauend, werden kurz verschiedene Lösungswege skizziert und diskutiert (Abschnitt 5), von denen einer im weiteren Verlauf der Arbeit weiterverfolgt wird. Dieser Lösungsweg ist die Entwicklung einer Individualsoftware, für die der Autor im Abschnitt 6 der Arbeit ein Konzept für eine Applikationsschicht entwickelt. Hierzu zieht der Autor die Referenzarchitektur der Workflow Management Coalition und zwei verschiedene Patterns heran. Im letzten Abschnitt resümiert der Autor seine Arbeit und gibt Ratschläge für ein weiteres Vorgehen zur weiteren Entwicklung des Konzeptes.

Das Ziel der Arbeit ist die Konzeption einer Applikationsschicht eines Workflow Management Systems. Dabei setzt sich der Autor das Ziel die Schicht durch ein Klassendiagramm zu visualisieren und durch ein Diagramm zur Funktionsweise zu unterstützen. Wichtig bei der Entwicklung ist dem Autor die Orientierung an der Referenzarchitektur der Workflow Management Coalition.

2 Theoretische Grundlagen

Innerhalb dieses Kapitels werden theoretischen Grundlagen zur Betrachtung von Prozessen und Geschäftsprozessen behandelt. Danach sollen die Workflows und die Workflowmanagementsysteme im Fokus der Betrachtung stehen. Weiterhin wird auf die Prozessmodellierung und deren Nutzung in der Softwareentwicklung eingegangen. Weiterhin werden dem Leser die Grundlagen zur Teileversorgung bei der DCAG vermittelt und die Gründe für End- und Zwischenbevorratungen erläutert.

2.1 Betrachtung von Prozessen und Geschäftsprozessen

Bevor eine Betrachtung zu der Thematik Workflow starten kann, empfiehlt es sich, auf eine genaue Definition und Abgrenzung des Begriffes „Geschäftsprozess“ einzugehen, da Geschäftsprozesse das grundlegende Subjekt in der Betrachtung von den Workflowmanagementsystemen sind.

2.1.1 Der Prozess

Eine nähere Betrachtung des Begriffes Prozess ergibt eine Ableitung aus dem lateinischen Verb „procedere“. Übertragen in unsere heutige Sprache lassen sich mehrere Bedeutungen ableiten: vorwärts schreiten, ablaufen, Fortschritte machen. Substantiviert man nun diese Verben, so entstehen sinngemäße Worte, wie z.B. Fortschritt, Vorgang, Ablauf, Verfahren.

Laut der DIN EN ISO 8402 [ISO03] ist unter einem Prozess ein

*„Satz von in Wechselbeziehungen stehenden Mitteln und Tätigkeiten“ zu verstehen,
„die Eingaben in Ergebnisse umgestalten.“*

Folgende Grundcharakteristika lassen sich demnach für einen Prozess feststellen:

- Ein Prozess besteht aus einer vorgeschriebenen Abfolge von Funktionen. Diese Abfolge ist vorher festgelegt und nicht beliebig.
- An einem Prozess sind verschiedene Ressourcen beteiligt. Diese Ressourcen können verschiedener Natur sein und in verschiedener Weise an dem Prozess beteiligt sein.
- Jedem Prozess sind eine definierte Eingabe und Ergebnisse zugeordnet. Solange nicht alle Eingabe-Faktoren (Ressourcen) vorhanden sind, kann der Prozess nicht starten, solange nicht alle Ergebnisse vorhanden sind, ist der Prozess noch nicht abgeschlossen.
- Es besteht im Allgemeinen ein Wertzuwachs zwischen Eingaben und Ergebnissen.

Das nachstehende Vorgangsmodell in der Abbildung 2-1 illustriert die Wirkungsweise eines Prozesses. Als Menge von Aktivitäten ‚verarbeitet‘ der Prozess Eingaben zu Ergebnissen, verbraucht dabei Ressourcen und steigert während der Umwandlung den Wert.

Das Objekt innerhalb des Prozesses kann von materieller Natur, als auch von immaterieller Natur sein. Materielle Objekte sind hierbei z.B. Baugruppen, immaterielle Objekte sind z.B. Informationen. Eine Kombination aus beiden Arten von Objekten ist ebenfalls möglich.

Ressourcen sind alle die Hilfsmittel, die zu einer Transformation benötigt werden. Dabei kann es sich um einfache Maschinen (Computer, Telefon), als auch um menschliche Ressourcen (Arbeitskraft) handeln. Auch hier kann eine Unterscheidung zwischen materiellen (beispielsweise Arbeitskraft und Maschinen) und immateriellen Ressourcen (Informationen) vorgenommen werden.

Während der Transformation wirken interne und externe Faktoren. Als interne Faktoren können dabei Vorgaben, wie z.B. ein Zielerreichungsgrad, gelten. Diese Faktoren sind unternehmensabhängig und können demnach einfach geändert werden.

Externe Faktoren sind Größen, die vom Unternehmen nicht einfach geändert werden können. Diese Faktoren können beispielsweise Umweltbedingungen sein, als auch staatliche Vorgaben zur Beschäftigung eines Mitarbeiters (Höchstarbeitsdauer pro Tag usw.).

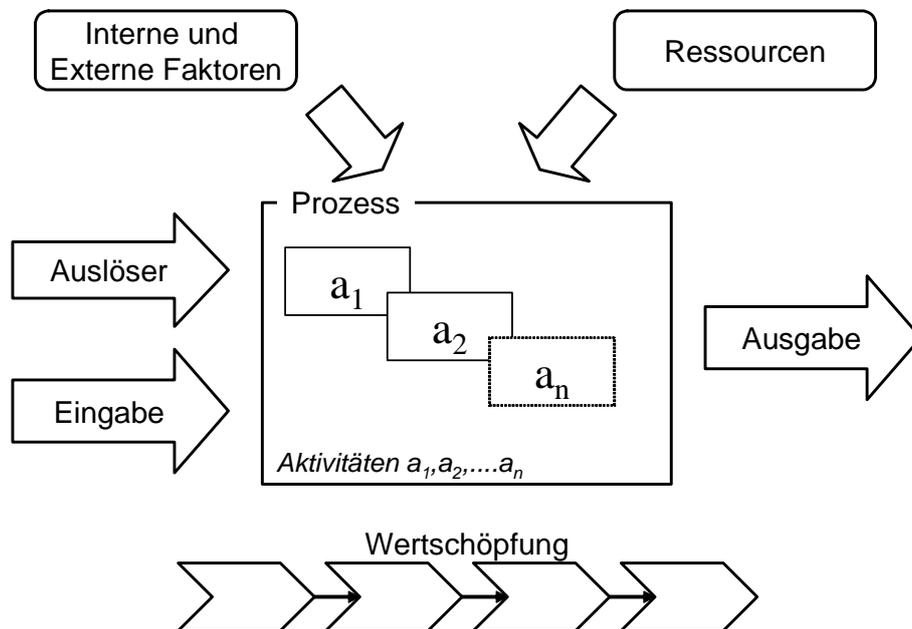


Abbildung 2-1: Prozessvorgangsmodell

Auslösende Ereignisse (in der Abbildung als Auslöser bezeichnet) lösen den Prozess aus bzw. führen zur Aktivierung der ersten Aktivität. Hierzu zählen beispielsweise andere vorangestellte Prozesse. Solche Ereignisse können einfache Impulse sein, wie z.B. das Eintreffen eines Auftrages, als auch schwerwiegende Ereignisse, beispielsweise das

Überschreiten eines Grenzwertes. In beiden Fällen wird bei Vorhandensein aller notwendigen Faktoren die erste Aktivität im Prozess angestoßen. Fehlt eine entsprechende Ressource, so kann die erste Aktivität nicht ausgeführt werden und wartet solange, bis alle notwendigen Ressourcen zur Verfügung stehen.

2.1.2 Geschäftsprozesse

Geschäftsprozesse stellen sich als ein Typ von Prozessen dar. Wie diese definiert und charakterisiert sind, wird in diesem Unterabschnitt erklärt..

In [Roh95] wird ein Geschäftsprozess, wie folgt definiert:

„Als Geschäftsprozesse werden die erfolgsrelevanten grundlegenden Unternehmenstätigkeiten, die zur Umsetzung der Unternehmensziele und Sicherung des Unternehmenserfolgs dienen, definiert. Sie beschreiben die wesentlichen Aufgaben, die das Geschäft eines Unternehmens charakterisieren.“

Ein Geschäftsprozess stellt sich demnach als eine Tätigkeit dar, die erfolgsrelevant ist und sich strikt an den unternehmerischen Aktivitäten orientiert.

Es handelt sich bei Geschäftsprozessen nicht nur um einzelne Prozesse. Oft werden bestimmte Unternehmenstätigkeiten als eine Menge von inhaltlich zusammenhängenden Prozessen aggregiert zu einem Geschäftsprozess zusammengefasst. Mit dieser Aggregation wird oft eine Abstraktion der Prozesse durchgeführt.

Um dem Leser ein anschauliches Beispiel zu geben, sei an dieser Stelle ein Rechnungsdurchlauf, wie er in jeder größeren Firma zu finden ist, erwähnt. Der Durchlauf einer Rechnung beschreibt hierbei die Abarbeitung der Rechnung innerhalb des Unternehmens durch verschiedene Abteilungen. Ein Durchlauf gliedert sich dabei meist in folgende grundlegende Tätigkeiten: Rechnungseingang, Rechnungsbearbeitung, Rechnungsweiterleitung, Rechnungsgenehmigung, Überweisung des Rechnungsbetrages und eine abschließende Archivierung. Durch eine Abstraktion können alle diese Tätigkeiten zu einem Geschäftsprozess aggregiert werden. Der einfache Vorgang der Rechnungsgenehmigung stellt demnach kein Geschäftsprozess dar, sondern würde nur als einzelner Prozess definiert werden.

Rohloff definiert, dass Geschäftsprozesse erfolgsrelevante Tätigkeiten sind, die der Sicherung des Unternehmenserfolges dienen. Der einfache Rechnungsdurchlauf ist hierbei innerhalb des Unternehmens ebenso relevant, wie in einem Produktionsbetrieb die Produktion der Produkte. Jedoch existiert eine Unterscheidung zwischen den beiden Geschäftsprozessen, die nicht durch die Definition Rohloffs ausgedrückt wird. Die Produktion der Produkte in einem Produktionsbetrieb ist der Kern der unternehmerischen Tätigkeit und

gehört zu den Kernkompetenzen. Wie sich Geschäftsprozesse und Kernkompetenz unterscheiden, wird in dem folgenden Abschnitt unterschieden.

2.1.3 Kernkompetenz als Untermenge von Geschäftsprozessen

Die Frage, ob Geschäftsprozesse auch automatisch mit in den Bereich der Kernkompetenzen eines Unternehmens fallen, bleibt unschlüssig. Nach der Definition Rohloffs sind alle Geschäftsprozesse für den Erfolg eines Unternehmens relevant und entscheidend für das erfolgreiche Fortbestehen des Unternehmens (siehe Definition). Kernkompetenz hingegen bezeichnet im engen Sinne, den Kern der Geschäftstätigkeit, die das Unternehmen sichert und deren Geschäftsfähigkeit untermauert. Hierunter fallen spezielle Fertigungstechniken in der Produktion eines Produktes bzw. deren Beherrschung durch ein Unternehmen. Die Kernkompetenzen stellen also die erfolgsrelevanten Tätigkeiten im Unternehmen dar und dienen somit der Sicherung des Unternehmenserfolgs. Sie sind somit Geschäftsprozesse.

Geschäftsprozesse sollten aber nicht nur als solche Tätigkeiten verstanden werden. Teilweise stellen sie auch eine Art der Unterstützung zum erfolgreichen Einsatz der Kernkompetenzen eines Unternehmens dar. Diese unterstützenden Tätigkeiten sind ebenso wichtig, wie die eigentlichen Kernkompetenzen. Sie ermöglichen es so den Mitarbeitern auf eine richtige Art und Weise in ihren Kerngeschäftsfeldern zu agieren und zu reagieren und das maximale Potenzial für die Kernarbeit einzusetzen.

Laut [Sohl96] sind Geschäftsprozesse unter anderem in Kern- und Supportprozesse zu unterscheiden. Kernprozesse können Kernkompetenzen darstellen bzw. deren aktiven Inhalt. Supportprozesse sorgen hingegen nur für einen erfolgreichen Ablauf der Kernprozesse. Dieser der Definition zugrunde liegenden Theorie wird hier gefolgt.

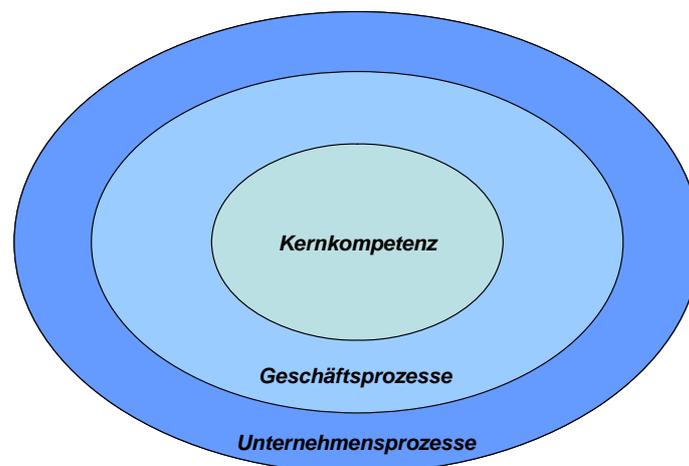


Abbildung 2-2: Einordnung der Kernkompetenz im Prozessumfeld

Obige Abbildung 2-2 zeigt auf eine einfache Weise das Verhältnis zwischen den Begriffen Kernkompetenz, Geschäftsprozesse und Unternehmensprozesse. In dieser Abbildung wurde

der Vollständigkeit halber die Menge aller in einem Unternehmen ablaufenden Prozesse als eine Menge mit dem Namen „Unternehmensprozesse“ mit aufgenommen. Als eine Teilmenge hiervon sind die Geschäftsprozesse zu nennen, die den unternehmerischen Erfolg sichern. Die Kernkompetenzen sind eine Teilmenge der Geschäftsprozesse, die den Kern der unternehmerischen Tätigkeit darstellen.

Innerhalb der Unternehmen wird versucht die Geschäftsprozesse einer Automatisierung zu unterziehen. Wie im obigen Beispiel des Rechnungsdurchlaufs, existieren bereits verschiedene Systeme in den Unternehmen, welche die einzelnen Schritte der Bearbeitung stark vereinfachen und sehr gut unterstützen. Oftmals werden die Rechnungen nur noch digital verarbeitet, anstatt sie händisch in Papierform weiterzureichen. Solche Systeme geben sich oft als Workflowmanagementsysteme zu erkennen, da sie die Abarbeitung der einzelnen Schritte koordinieren und die unterschiedlichen daran beteiligten humanen Ressourcen mit involvieren.

Im weiteren Verlauf dieses Kapitels soll daher zuerst geklärt werden, wie ein Workflow definiert ist und welche Eigenschaften er besitzt. Danach kann eine Erklärung zum Thema Workflowmanagementsystem erfolgen.

2.2 Theoretische Betrachtung eines Workflows

Die theoretische Betrachtung eines Workflows soll in diesem Abschnitt den Hauptaspekt bilden. Sie wird von der Definition der Workflow Management Coalition, kurz WfMC, abgeleitet und deren Definition eines Workflows genauer betrachtet. Danach wird ein Weg aufgezeigt, wie eine Workflowidentifikation laut der Object Management Group, kurz OMG, erfolgen kann. Eine abschließende Workflowntypisierung rundet die theoretische Betrachtung der Workflows ab.

2.2.1 Workflowdefinition der Workflow Management Coalition

Innerhalb dieser Arbeit soll sich strikt an den Definitionen der WfMC orientiert werden. Diese definiert einen Workflow, wie folgt:

„The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.“ [WfMC96, S.8]

Die Definition zeigt die Hauptaspekte eines Workflow genau auf:

- Es handelt sich um eine Teil- oder Vollautomatisierung eines Geschäftsprozesses.

- Der Fokus liegt auf Dokumenten, Informationen oder Aufgaben.
- Die Übermittlung von Dokumenten, Informationen oder Aufgaben an verschiedene Teilnehmer spielt eine wesentliche Rolle.
- Es existieren Regeln zur Abarbeitung.

An dieser Stelle ist zu bemerken, dass in der Definition keinerlei Anforderung oder Hinweise zu finden sind, inwieweit beispielsweise die Automatisierung vollzogen werden soll. Es gibt von Seiten der WfMC weiterhin aber Ergänzungen zu der Definition, welche im Folgenden mit erläutert werden sollen:

“ The automation of a business process is defined within a Process Definition, which identifies the various process activities, procedural rules and associated control data used to manage the workflow during process enactment” [WfMC96, S.8]*

Die Ergänzung beschreibt die Anforderung nach einer Prozessdefinition, welche die grundlegenden Prozessaktivitäten, Ablaufregeln und Kontrolldaten zur Laufzeit zur Verfügung stellt. Für die Anwendung eines regelkonformen Workflowmanagementsystems, kurz WFMS, muss eine genaue Definition der zugrunde liegenden Prozesse erfolgen.

“ Many individual process instances may be operational during process enactment, each associated with a specific set of data relevant to that individual process instance (or workflow "Case")” [WfMC96, S.8]*

Viele einzelne Prozessinstanzen interoperieren miteinander während der Laufzeit, wobei jede Prozessinstanz mit einem speziellen Satz von Daten versehen ist. Das bedeutet, es besteht eine einfache Interaktion zwischen verschieden vielen Prozessen eines oder mehrerer Workflows.

„ A loose distinction is sometimes drawn between production workflow, in which most of the procedural rules are defined in advance, and ad-hoc workflow, in which the procedural rules may be modified or created during the operation of the process.” [WfMC96, S.8]*

Unter Umständen kommt eine Unterscheidung zwischen Produktionsworkflows zustande, deren Ablaufregeln schon im Vorhinein definiert werden, und Ad-hoc-Workflows, deren Ablaufregeln erst während der Laufzeit erstellt werden.

Wichtig ist abschließend zu erwähnen, dass innerhalb der WfMC keine Unterscheidung zwischen technischen Prozessen und Geschäftsprozessen gemacht wird. Allgemein fokussieren sich aber alle Dokumente, rund um das Thema WFMS, auf Geschäftsprozesse. Daher wird in der weiteren Arbeit nur noch von Geschäftsprozessen gesprochen.

2.2.2 Vorgehen zur Bestimmung eines Workflows laut Object Management Group

In [OMG98] wird ein einfaches Vorgangsmodell durch die OMG vorgegeben, mit dessen Hilfe sich ein Workflow schnell und einfach definieren lässt. Zu beantworten sind drei grundlegende Fragen:

- Wer ist für die Ausführung einer bestimmten Aufgabe verantwortlich?
- Was sind die Aufgaben und welche Geschäftsregeln liegen dahinter?
- Mit welchen IT-Tools wird die Abarbeitung der Aufgaben unterstützt?

Mit Hilfe der drei Fragen lassen sich leicht drei grundlegende, orthogonale Dimensionen zur Beschreibung eines Workflows herausarbeiten: IT-Ressourcen, Prozesslogik und Organisation. Die folgende Abbildung 2-3 zeigt die durch die drei Fragen implizierten Dimensionen eines Workflows.

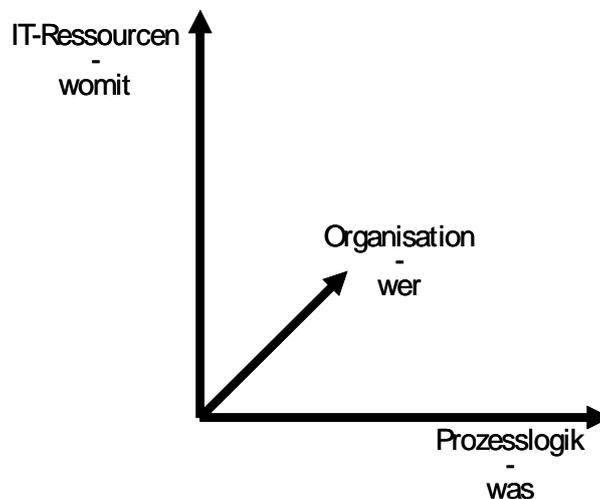


Abbildung 2-3: Workflowdimensionen

Sind diese drei Fragen beantwortet, so kann auf eine einfache Art und Weise eine kurze Workflowdefinition für einen spezifischen Fall erstellt werden.

Die Antworten auf diese drei Fragen beinhalten im engeren Sinne alle relevanten Informationen rund um die Workflowdefinition und im weiteren Sinne alle Informationen zur Definition des Prozesses, der diesem Workflow zugrunde liegt.

Folgende Aspekte werden durch eine Beantwortung geklärt:

- Die Aufgaben eines Workflows sind die Aktivitäten innerhalb der Prozessbeschreibung.
- Ressourcen sind von personeller Art und Weise – also wer führt den Workflow aus und welche IT-Systeme nutzt der Prozessverantwortliche bei der Abarbeitung der Aufgaben.
- Die Geschäftsregeln können Auskunft darüber geben, welche Auslöser zur Ausführung des Prozesses führen und welche Ergebnisse generiert werden müssen.
- Die IT-Tools können auf einfache Weise zeigen, welche Eingaben und Ergebnisse ein Prozess generiert.

Wie zu sehen ist, beinhaltet die Beantwortung der Fragen die nötigen Informationen zur Generierung eines Workflows, allerdings nicht alle Daten, die für eine Prozessdefinition notwendig sind. Es bleibt daher die Frage, wie Workflows und Prozesse konkret zusammenhängen. Dies soll im nächsten Unterabschnitt geklärt werden.

2.2.3 Von der Prozessdefinition zum Workflow

Eine Prozessdefinition findet immer kontextabhängig statt – je nachdem wofür eine Definition eines Prozess weiterverwendet werden soll bzw. für welchen Zweck eine umfangreiche Prozessbeschreibung gebraucht wird, sind die Daten unterschiedlich für die Erstellung eines WFMS zu gebrauchen.

Es existieren viele unterschiedliche Verfahren, welche die Überführung von Prozessbeschreibungen zu Abläufen in einem Workflowmanagementsystem behandeln. Innerhalb dieser Arbeit wird ein einfaches Verfahren unter der Nutzung der erweiterten ereignisgesteuerten Prozessketten, kurz eEPK's, genutzt, die alle notwendigen Informationen bereitstellen. Die Notation und die Semantik der eEPK's basiert innerhalb dieser Arbeit auf den internen Vorgaben der DCAG, die in dem Dokument [Glo00] festgehalten sind. Die wichtigsten Eigenschaften der eEPK's, welche auf den Ausführungen von [Sch91] basieren, sind:

- Ereignisse + Prozesse werden in einer Kette nacheinander folgend aufgelistet
- Hinterlegung einzelner Prozesse durch detaillierte eEPK's
- Hinterlegung der DV-Unterstützung eines Prozesses
- Hinterlegung der Eingangs- und Ausgangsinformationen für einen Prozess
- Hinterlegung der Organisationen/Prozessverantwortlichen
- Logische Konnektoren zur Verdeutlichung der Ablauflogik und Ausnahmeregelung

Diese Eigenschaften werden primär von Nutzen sein, wenn beispielsweise modellierte Soll-Prozesse in eine Workflowstruktur übernommen werden sollen. Eine Konvention wird jedoch im Vor herein gemacht: Verfeinerungen von Prozessen, wie sie grundsätzlich in eEPK's

möglich sind, werden nicht mit berücksichtigt, da sie einen zu hohen Detaillierungsgrad besitzen. Ausschließlich Prozesse auf der Grundebene werden für die Überführung in das WFMS vorgesehen.

Aus der Menge der Ein- und Ausgangsdaten der unterschiedlichen Prozesse kann leicht abgeschätzt werden, mit welchen Daten ein Workflow im späteren System konfrontiert wird bzw. welche Daten er innerhalb des Systems zur Abarbeitung benötigt. Dies wird im späteren Verlauf der Arbeit ein erster Anhaltspunkt für einen ersten Entwurf des Klassendiagramms sein.

Der generelle Ablauf einer eEPK's wird ausschlaggebend für den Ablauf der Workflows innerhalb des Gesamtsystems sein. Hierin sind explizit die Wege eines Workflows definiert, inklusive aller parallelen, alternativen Pfade und Schleifen, als auch implizit die Geschäftsregeln, die einem bestimmten Ablauf zugrunde liegen und die es zu extrahieren gilt.

Diese Geschäftsregeln geben beispielsweise eine bestimmte Reihenfolge für den Ablauf eines speziellen Workflowobjektes an. Da diese Regeln nur implizit in dem eEPK stecken, müssen sie entsprechend extrahiert werden. Dies geschieht durch die Extraktion des Ablaufwissens aus einer eEPK, durch das Wissen des Modellierers, der Kenntnis darüber besitzt, welche Regeln innerhalb des Modells gelten und durch externe Personen, die ebenfalls das Ablaufwissen über den Prozess besitzen, aber nicht bei der Modellierung der eEPK mitwirken. Besonders letztere Personengruppe ist entscheidend für eine konsistente Abbildung des Gesamtprozesses auf die eEPK, da nur sie Kenntnis über die Feinheiten des Ablaufs besitzen. Ein Instrument zur Offenlegung dieses Wissens, sind Interviews, die auch innerhalb des Projektes, in dem diese Diplomarbeit erarbeitet wurde, durchgeführt wurden. Die entsprechenden Personen sind die Disponenten für verschiedene Sortimente.

2.2.4 Workflow-Typisierung

In der Definition der Workflows der WfMC wurde eine Unterscheidung der Workflows in Produktionsworkflows und Ad-hoc-Workflows erwähnt. In [GIG03] werden jedoch noch zwei weitere Kategorien zur Unterscheidung von Workflows angegeben, sodass sich eine Auflistung von insgesamt vier verschiedenen Workflowtypen ergibt:

- Administrative Workflows
- Ad-hoc-Workflows
- Kollaborative Workflows
- Produktion-Workflows

In [GIG03] werden alle vier Gruppen anhand des Kriteriums Wiederholung und anhand des Geschäftswert unterschieden. Ordnet man beide Kriterien auf zwei Achsen an, so entsteht nach der Einordnung der vier Workflowtypen folgendes Diagramm:

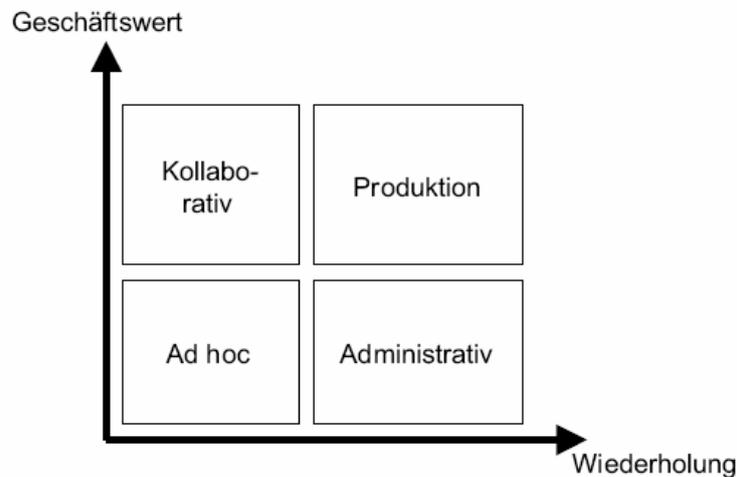


Abbildung 2-4: Workflow-Typisierung [HE03]

Die Abbildung 2-4 zeigt auf eine einfache Art und Weise die Typisierung der Workflowtypen. Beispielsweise zeichnen sich Ad-hoc-Workflows demnach durch eine geringe Wiederholung und geringen Geschäftswert aus.

Da weder Wiederholung, noch der Geschäftswert, innerhalb der Unterscheidung auf einen konkreten Wert normiert sind, bleibt hier nur eine Einteilung nach den eigenen Regeln eines jeden Unternehmens. Die in dieser Arbeit betrachteten Workflows können nicht in eine einzelne Kategorie eingeordnet werden. Obwohl alle Workflows über ein und dasselbe WFMS ablaufen, hat jeder Workflow seinen eigenen Wert, der durch viele verschiedene Faktoren bestimmt wird. Ebenso kann eine Unterteilung nach dem zweiten Kriterium, den Wiederholungen, nicht einfach vollzogen werden. Die tatsächliche Anzahl von Workflows, die in einem späteren WFMS behandelt werden sollen, kann nur grob geschätzt werden. Durch Schätzungen eines Mitarbeiters der DCAG kann mit ca. 500 Workflows (eine sehr grobe Schätzung) pro Jahr gerechnet werden. Dieser Wert kann als hoch in der Dimension „Wiederholungen“ angesehen werden.

Sicherlich kann die Typisierung von Workflows noch durch andere Kriterien erfolgen, wie z.B. den Grad einer Automatisierung. Wie hoch eine Automatisierung tatsächlich ist, ist natürlich abhängig von der Unterstützung im Ablauf eines Workflows durch geeignete Mittel. Eines dieser Mittel ist beispielsweise ein WFMS, dessen theoretische Betrachtung im folgenden Abschnitt erfolgt.

2.3 Theoretische Betrachtung eines Workflowmanagementsystems

Oftmals finden sich im Unternehmen viele unterschiedliche Prozesse, die sich stetig wiederholen und den Benutzer immer wieder mit denselben Anforderungen konfrontieren. Solche sich oft wiederholenden Prozesse sind dafür prädestiniert, eine Unterstützung durch ein Softwaresystem zu erfahren. Softwaresysteme, die Workflows unterstützen, werden als Workflowmanagementsysteme bezeichnet und werden innerhalb dieses Abschnitts durch die Definition im ersten Abschnitt, die allgemeinen Anforderungen und die Adaption der Systeme charakterisiert.

2.3.1 Definition eines Workflowmanagementsystems

Nachdem im Abschnitt 2.2 die theoretische Betrachtung eines Workflows abgeschlossen wurde, wird in diesem Abschnitt die Definition eines Workflowmanagementsystems aufgezeigt – es wird dabei auf die Definition der WfMC zurückgegriffen:

“A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.” [WfMC96, S.9]

Es handelt sich bei einem WFMS, der Definition folgend, um ein System, welches die Definition, die Erstellung und das Steuern von Workflows über eine Software zulässt. Dabei ist das System auf ein oder mehr Workflowengines verteilt und kann eine Prozessdefinition interpretieren, interagiert mit Workflowteilnehmern und involviert bei Bedarf IT-Tools und IT-Applikationen.

Die WfMC lässt die Definition nicht so abstrakt stehen, sondern erläutert diese noch durch eine Ergänzung:

„ A Workflow Management System consists of software components to store and interpret process definitions, create and manage workflow instances as they are executed, and control their interaction with workflow participants and applications.” [WfMC96, S.9]

Ein zukünftiges WFMS ist komponentenbasiert und muss die Funktionalität bieten, Prozessdefinitionen zu speichern und zu interpretieren. Darüber hinaus müssen Workflowinstanzen innerhalb des Systems erstellt und während ihrer Ausführung gesteuert und alle Interaktionen kontrolliert werden können.

Die WfMC fügt noch eine weitere Ergänzung hinzu:

“ Such systems also typically provide administrative and supervisory functions, for example to allow work reassignment or escalation, plus audit and management*

information on the system overall or relating to individual process instances.” [WfMC96, S.9]

Betrachtet man diese Ergänzung genauer, so werden hierin noch einige Eigenschaften eines zukünftigen WFMS beschrieben, die zur Unterstützung der Unternehmensleitung dienen können.

Aus diesen Punkten der Definition können leicht genaue Anforderungen an ein WFMS extrahiert werden, die sich in der Referenzarchitektur der WfMC zeigen. Hier sei auf den Abschnitt 6.1.1 verwiesen, in dem die Architektur mit allen daraus abgeleiteten Anforderungen an ein WfMC-konformes WFMS genau erklärt wird.

Im kommenden Unterabschnitt werden die generellen Anforderungen an ein WFMS analysiert und aufgestellt.

2.3.2 Anforderungen an ein Workflowmanagementsystems

In Abschnitt 2.1 wurde eine Untersuchung der Begriffe Prozess, Tätigkeit und Geschäftsprozess vorgenommen. Diese ist nicht irrelevant und hat einen praktischen Hintergrund, der hier erläutert werden soll.

Bei der ersten Konzeption einer Systemarchitektur für ein WFMS, ist entscheidend, in wie weit welche Tätigkeiten in dem System abgebildet werden sollen und bis zu welcher Granularität das System den Benutzer unterstützen soll. Da sich Workflows als ein teil- oder vollautomatisierter Geschäftsprozess innerhalb eines integrativen Softwaresystems darstellen, ist entscheidend, welche Aktivitäten des Geschäftsprozesses durch eine Systemkomponente unterstützt werden und welche nicht. Viele Tätigkeiten in einem Workflow können nur eingeschränkt durch das System gestützt abgearbeitet werden. Der Benutzer muss hier also einen Großteil der Arbeit unabhängig vom System vollbringen.

Es existieren darüber hinaus verschiedene Tätigkeiten, die gar nicht durch ein Softwaresystem unterstützt werden können bzw. bei denen es keinen Sinn macht eine Unterstützung durch ein IT-System zu gewähren. Daher ist schon zu Beginn der Konzepterstellung wichtig festzulegen, inwieweit die Implementierung erfolgen soll.

Eine sehr wichtige Anforderung an WFMS ist die Untersuchung der Interaktion zwischen WFMS und Benutzer. Die Hauptprämisse ist in jedem Fall die Benutzerfreundlichkeit des Systems. Der Benutzer soll mit dem System arbeiten, ohne viel darüber nachdenken zu müssen und sich viel mit dem System beschäftigen zu müssen. Steht der Benutzer vor dem Problem, mit einem nicht-intuitiven System zu arbeiten, wird das Endprodukt des Workflows entweder gar nicht oder aber nur sehr verzögert entstehen. Weiterhin kann ein WFMS auch die Kommunikation zwischen den Beteiligten eines Workflows erleichtern bzw. beschleunigen.

Erfahrungen des Autors zeigen, dass in einer Unternehmensleitung sehr oft unterschiedliche Ansichten über den Grad der Automatisierung eines Workflows existieren. Oftmals werden Argumente vorgebracht, deren Kern darin besteht, einem WFMS zu unterstellen, es würde dem Benutzer beim Handeln das Denken abnehmen – man ist hier strikt gegen einen hohen Automatisierungsgrad. Aus den Augen des Autors kann dieses Argument allerdings nicht gehalten werden. Heutige WFMS sind gar nicht in der Lage das Fachwissen eines Benutzers komplett zu ersetzen und ihm auf diesem Wege das Denken abzunehmen. Diese Substitution des Benutzers durch ein WFMS ist nicht real.

Viele leitende Angestellte sind der Meinung, dass WFM-Systeme primär für den Zweck existieren, den Benutzer zu leiten und zu lenken. Dies stellt auch, in den Augen des Autors, die Hauptaufgabe eines WFMS dar. Das Leiten des Benutzers unter Berücksichtigung der Bereitstellung aller relevanten Daten und Funktionen, ist die Hauptanforderung an ein WFMS.

2.3.3 Adaption eines Workflowmanagementsystems

Die Adaption eines WFMS besteht hauptsächlich in der Anpassung eines vorhandenen Workflows an neue Begebenheiten, die im Workflow Berücksichtigung finden sollen, also auch vollkommen neue Vorgänge, die komplett in ein System eingepflegt werden sollen.

Das geeignete Mittel für solch eine Interaktion stellt eine formale Sprache dar, mit deren Möglichkeiten der Benutzer das WFMS aktiv beeinflussen kann. Hierbei spielt die Mächtigkeit der Sprache eine zentrale Rolle, sprich die Nähe zur Objektorientierung und zu logischen Ausdrücken. Eine Objektorientierung wird dann nötig, wenn sich die Workflow-Objekte in dem System ‚bewegen‘ und ein neues Objekt zu hinerlegen ist; eine logische Mächtigkeit wird gebraucht.

Gegenwärtig existieren viele Sprachen, mit deren Sprachkonstrukten sich auf einfache Art und Weise Workflows neu definieren lassen. Eine allgemeine Unterscheidung kann durch die Syntax gemacht werden – während bei einigen Sprachen Petri-Netze als Eingabeformat dienen, sind bei anderen Sprachen textuelle Elemente als Eingabe vorgesehen. Neben vielen Workflowengine-abhängigen Sprachen, existieren auch einige frei verfügbare, abstrakte formale Sprachen. Durch diese Sprachen können WFMS, mit entsprechenden Schnittstellen für die entsprechenden Sprachen, angepasst werden.

Sofern die Erstellung eines relativ unflexiblen WFMS das Ziel in der Entwicklung ist, kann auf die Adaption des späteren WFMS wenig Wert gelegt werden. Anpassungen müssen dann durch eine Neuprogrammierung durchgeführt werden.

Vor der Neuerstellung eines WFMS kann es von Vorteil sein, die zugrunde liegenden Prozesse genauer zu betrachten und zu modellieren, um sich die Komplexität und die Kompliziertheit der Prozesse zu verdeutlichen. Ebenso kann die Modellierung dieser

Prozesse einen ersten Anhaltspunkt für die spätere Entwicklung und Struktur des WFMS geben. Im folgenden Abschnitt wird dieser Aspekt der Nutzung der Prozessmodellierung in der Softwareentwicklung genauer betrachtet.

2.4 Nutzung der Prozessmodellierung in der Softwareentwicklung

Dieser Abschnitt wird den theoretischen Hintergrund zu der Softwareentwicklung anhand des Sieben-Phasen-Modells erklären und aufzeigen, welche Ziele die allgemeine Prozessmodellierung hat. In einem weiteren Unterabschnitt wird erläutert, wie und wann die Prozessmodellierung innerhalb der Softwareentwicklung genutzt werden kann.

2.4.1 Die Sieben-Phasen-Softwareentwicklung

Die Unterteilung komplexer und komplizierter Sachverhalte in verschiedene Teilbereiche ist eine Hilfe, durch welche die Gesamtheit der gegebenen Problematik meist erst vollständig erkennbar wird. Die Unterteilung der Softwareentwicklung in verschiedene Phasen, ist eine weit verbreitete und allgemein bekannte Methode, den gesamten Prozess der Entwicklung eines Softwareproduktes zu unterteilen.

Man unterscheidet im Sieben-Phasen-Modell nach [Du05] die folgenden sieben aufeinander folgenden und auf den Ergebnissen der vorangegangenen Phase aufbauenden, Entwicklungsphasen:

- Problemdefinition
- Anforderungsanalyse
- Spezifikation
- Entwurf
- Implementation
- Erprobung
- Auslieferung

Das iterative Durchlaufen dieser Phasen ist nicht als gradliniger Prozess zu verstehen, sondern wird in vielen Instanzen durch einen Rücksprung begleitet, der in eine frühere Phase führen kann. Um die Bedeutung der Phasen genauer zu verstehen, wird an dieser Stelle eine Beschreibung der Phasen durchgeführt:

- *Problemdefinition*: Die Definition eines Problems ist die Grundlage zur Entwicklung eines Softwareproduktes. Hier hat der Anwender zum ersten Mal die Anforderung nach einem zukünftigen Softwareprodukt formuliert.
- *Anforderungsanalyse*: In der Phase der Anforderungsanalyse muss der Anwender seine Anforderungen an des Softwareprodukt formulieren und konkretisieren. Hier

werden durch die Entwickler alle nötigen Informationen gesammelt und alle Anforderungen genau protokolliert.

- *Spezifikation*: Die Phase der Spezifikation dient zur abschließenden Fixierung der Anforderungen. Hierbei wird eine vertragliche Grundlage (meist Pflichtenheft) zur Softwareentwicklung geschaffen, nach der das Entwicklerteam (intern oder extern) arbeiten kann und muss. Dabei sind alle machbaren Anforderungen zu übernehmen und durch Auftragnehmer und Auftraggeber abzugleichen.
- *Entwurf*: Der Entwurf eines Softwareproduktes beschreibt ein erstes Modell einer abstrakten Lösung. Hierbei wird noch kein coderelevanter Inhalt verwendet, sondern auf einer abstrakten Ebene mittels geeigneter Darstellungs- und Beschreibungsmittel das zu implementierende System erstmals in DV-nahe Strukturen gebracht.
- *Implementation*: Die Implementation des Entwurfs beschreibt die Umsetzung der Anforderungen und der im Entwurf festgehaltenen Strukturen in ein erstes prototypisches System, welches sich an der späteren Zielumgebung orientiert.
- *Erprobung*: Die Phase der Erprobung dient zur Verfeinerung des Prototypen aus der vorherigen Phase und dem Finden von Fehlern, struktureller als auch logischer Natur. Ziel der Phase ist die vollständige Umsetzung aller Lasten und Realisierung eines Produktes, welches frei von offensichtlichen Fehlern ist.
- *Auslieferung*: Die letzte Phase der Auslieferung beschreibt die Übergabe des fertigen Systems vom Auftragnehmer an den Auftraggeber, inklusiver jeglicher Art von Dokumentation.

Man erkennt an dieser Stelle, dass mit der Auslieferung der Software der Entstehungsprozess abgeschlossen ist, der Lebenszyklus aber gerade erst begonnen hat. Dem fertigen Produkt steht noch ein Zeitraum bevor, in dem es durch einige Aktualisierungen immer wieder verändert wird.

Die ersten Phasen der Softwareentwicklung (Problemdefinition und Anforderungsanalyse) werden oft in Kooperation mit den zukünftigen Anwendern gemacht. Hierbei spielt die Erklärung der Probleme durch den späteren Nutzer und die Abbildung der Probleme in einer entsprechenden Struktur eine wichtige Rolle. Dazu wird oft die Prozessmodellierung genutzt, um eine erste anschauliche Basis für die Kommunikation der Entwickler mit den Anwendern zu schaffen. In den nachfolgenden Abschnitten wird darauf eingegangen.

2.4.2 Ziele der Prozessmodellierung

Die Ziele der Prozessmodellierung sind so vielfältig, wie die einzusetzenden Mittel, diese zu erreichen. Es lassen sich aber leicht ein paar wenige Ziele herausfiltern, die wohl bei jeder Modellierung von Prozessen mit berücksichtigt werden:

- Prozessdokumentation
- Prozesssimulation
- Prozessverbesserung/-optimierung
- Transparenz schaffen
- Komplexität reduzieren
- Abläufe/Aktivitäten dokumentieren
- Performanzmessungen innerhalb von Abläufen
- Potenziale in Abläufen offen legen

Diese Liste ist natürlich unvollständig und gibt nur einen kleinen Ausschnitt aus den Zielen der Prozessmodellierung wieder. Entsprechend diesen Zielstellungen profitiert auch das Unternehmen durch den entstehenden Nutzen.

Am Beispiel der Wissensdokumentation kann dies schnell und einfach erkannt werden. Sind die Abläufe während der Entwicklung eines Produktes genau dokumentiert, so können diese Prozesse irreal und real schnell modifiziert werden, da die Auswirkungen schon im Voraus erkennbar sind. Es ist damit eine Zeitersparnis verbunden, die mit einer Kostenreduzierung einhergeht. Darüber hinaus kennt das Unternehmen die interne Funktionsweise der Prozesse. Dies eröffnet neue Potenziale zur Verbesserung, gegebenenfalls Optimierung, und Umgestaltung.

Die Prozessmodellierung dient oft als Grundlage zur Umgestaltung/Neugestaltung von internen Prozessen. Ziel hierbei ist immer die Verbesserung von internen Abläufen und bestehenden Strukturen durch eine Um- und Neustrukturierung.

2.4.3 Allgemeine Prozessmodellierung in den Softwareentwicklungsphasen

Mittels der Prozessmodellierung werden oft Vorgänge und Abläufe dokumentiert, wobei in den meisten Projekten diese Dokumentation erst im Nachhinein, sprich, nachdem die Prozesse etabliert wurden, erfolgt. Da sich die Softwareentwicklung allerdings nicht nach der Etablierung eines Produktes befindet, sondern definitiv davor, wird die Prozessmodellierung schon vor der Entstehung des Produktes eingesetzt und die Prozessaufnahme in der Softwareentwicklung so früh, wie möglich, angesetzt.

Viele Produkte der heute verfügbaren betrieblichen Anwendungssoftware orientieren sich an den betrieblichen Abläufen innerhalb der Unternehmen und setzen diese Abläufe als Grundlage der Abläufe in der Software um. Es ist daher nur verständlich die grundlegenden

Vorgänge zuerst in allgemeinen Modellen zu dokumentieren und danach in spezifischere Modelle zu überführen. Ausgehend von einem plattformunabhängigen Modell wird durch Verfeinerung und Überführung in andere Modelle ein immer feineres und letztlich plattformabhängiges Modell geschaffen.

Die Aufnahme der zukünftigen Anwendung zugrunde liegenden Prozesse, stellt den Beginn der Softwareentwicklung innerhalb der ersten Phasen dar. Neben der Aufnahme von Anforderungen, ist die Aufnahme der Prozesse eine der grundlegenden Aktivitäten. Hierbei wird mit wenigen Mitteln ein möglichst genaues Abbild der Realität geschaffen. Feinheiten der Prozesse müssen in einzelnen Modellen genau erfasst werden – für unterschiedliche Fokussierungen in der Prozessmodellierung können dann auch unterschiedliche Modelle herangezogen werden. Wichtig ist die hohe Detaillierung der Prozessmodelle – es gilt hier zwar nicht: „je mehr Informationen, desto besser“, allerdings ist es wichtig, möglichst alle nötigen Informationen über die Prozesse vollständig herauszufinden, denn umso besser können sie anschließend modelliert und in der Software umgesetzt werden.

Eine Prozessmodellierung muss sich immer an den bestehenden Prozessen und Strukturen orientieren und diese in das Zentrum des Betrachters setzen. Dazu sind die genaue Kenntnis und das Verständnis der Abläufe elementar. Der folgende Abschnitt wird deshalb dem Leser einen Einblick in die Vorgänge der Teileversorgung der DCAG geben.

2.5 Grundlegendes Verständnis zur Teileversorgung

Um das ganze Wesen der Teileversorgung in der DCAG zu verstehen, werden in diesem Abschnitt alle notwendigen Grundlagen zu dem Thema aufbereitet. Dazu erfährt der Leser, wie ein typischer Lebenszyklus eines Teils charakterisiert ist, wie die allgemeine Teileversorgung im GLC durchgeführt wird und welche Unterschiede sich zwischen der normalen Teileversorgung und einer Bevorratung ergeben. Danach werden beide Arten von Bevorratungsaktionen erläutert und unterschieden.

2.5.1 Lebenszyklus eines Teils im Teilewesen

Im Zuge einer Diplomarbeit [Lip04] wurde im GLC die Entwicklung des Absatzes einer Teilenummer untersucht und eine typische Lebenskurve entwickelt, die an dieser Stelle näher betrachtet werden soll.

In der folgenden Abbildung 2-5 wird die Absatzkurve gezeigt, deren Verlauf im Folgenden näher erklärt werden soll.

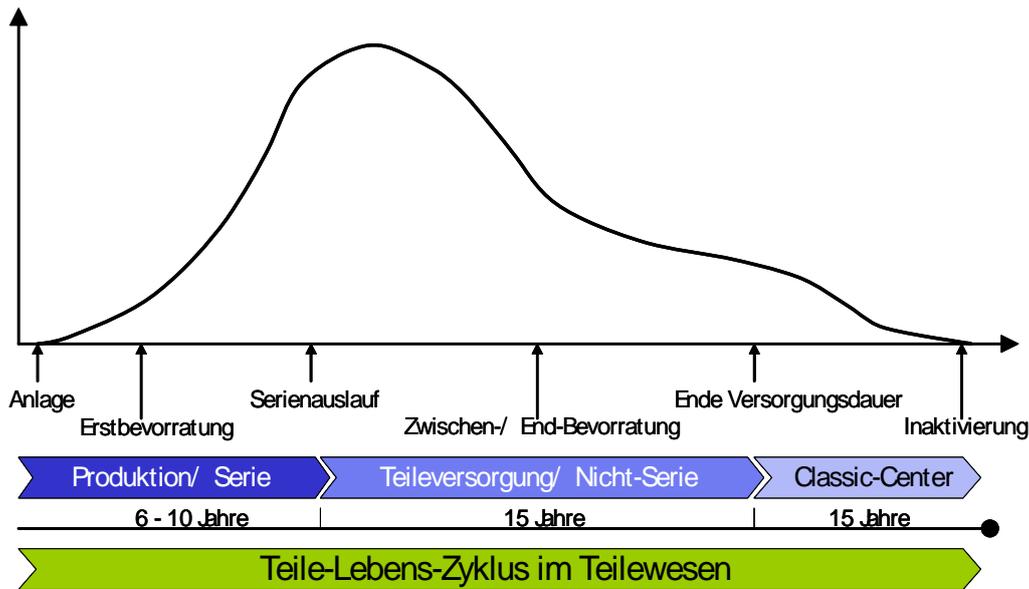


Abbildung 2-5: Typische Lebenszykluskurve eines Teils [Gaschler]

Innerhalb dieses Diagramms wird die Ordinate zur Indizierung der Menge verwendet, während die Abszisse zur Indizierung der Zeit verwendet wird. Beginnend bei einem Zeitpunkt 0, startet mit der Anlage der Teilenummer, auch die Produktion des Modells/der Modelle, in der/denen das Teil verbaut ist. Die Produktion, auch als Serie bezeichnet, läuft danach mit einer Dauer von ca. 6-10 Jahren. Innerhalb dieser Dauer steigt der Absatz des Teils stetig an und erreicht den Höhepunkt nach dem Serienauslauf, sprich dem Ende der Produktion. Dieses Phänomen ist leicht durch die zu diesem Zeitpunkt gegenwärtige höchste Anzahl von verbauten Teilen in den Fahrzeugen auf dem Markt zu erklären.

Nach dem Serienauslauf beginnt die Phase der Nicht-Serie bzw. Nachserie. Hier garantiert die DCAG ihren Kunden eine Teileverfügbarkeit von mindestens 15 Jahren, in einigen Sonderfällen auch eine Dauer von 30 Jahren, die sich in dem Diagramm aus den farblich gekennzeichneten Phasen „Teileversorgung/Nicht-Serie“ und „Classic-Center“ ergibt.

Das Classic-Center ist eine Einrichtung der DCAG, die für die Versorgung und die Betreuung von als besonders wertvoll eingestufteten Fahrzeugen verantwortlich ist. Das Classic-Center versorgt allerdings nur wenige, ausgesuchte Fahrzeuge mit Teilen - das jedoch bis zu einem Zeitpunkt von 30 Jahren nach Serienauslauf.

In dem Diagramm sind auch die Zeitpunkte, zu denen Zwischen- oder Endbevorratungen berücksichtigt werden müssen, zu finden. Allerdings ist an dieser Stelle zu bemerken, dass beide Zeitpunkte nicht fest sind oder durch irgendwelche Variablen bestimmt werden, sondern durch eine Vielfalt von Faktoren hervorgerufen werden können. In dem Abschnitt 2.6 werden die Gründe für diese Bevorratungen genau erläutert.

2.5.2 Regeldisposition und Bevorratungsaktionen allgemein

Die normale Teileversorgung im GLC läuft auf Basis einer so genannten Regeldisposition. Dabei beschreibt der Begriff die Tätigkeiten, die insgesamt für die Verplanung, Bestellung und Kontrolle der Teileversorgung nötig sind. Hierunter fallen zum Beispiel die Bestandskontrolle und die Überwachung der Bestellungen, die im Falle der DCAG automatisch durch ein Dispositions- und Bestellsystem (kurz: DuB) vorgenommen wird. Der Disponent hat hierbei die Aufgabe, autonome Tätigkeiten des Systems, wie die Bestellung eines Teils, zu überwachen und gegebenenfalls einzugreifen.

Neben diesen Tätigkeiten erhalten die Disponenten immer wieder die Aufgaben, Teile zu bevorraten. Die Besonderheit an einer Bevorratung ist, dass für einen gewissen Zeitraum die Versorgung des Teils nicht mehr auf Basis einer Regeldisposition ablaufen kann, sondern alle entstehenden Bedarfe aus dem aktuellen Bestand des GLC gedeckt werden müssen. Dies kann unterschiedliche Gründe haben, die im kommenden Abschnitt genau erklärt werden. Durch geeignete Maßnahmen muss der Disponent versuchen, den Zeitraum des Ausfalls der Regeldisposition zu überbrücken. Dies geschieht durch eine Bevorratungsaktion (kurz: BV-Aktion). Hierbei kann entweder der Fall auftreten, dass eine Disposition nur für eine Zeit lang nicht mehr möglich ist, als auch der Fall, dass eine Disposition gar nicht mehr möglich ist. Im ersten Fall wird in der DCAG von einer Zwischenbevorratungen (kurz: ZBV) und im zweiten Fall von einer Endbevorratungen (kurz: EBV) gesprochen. Beide BV-Aktionen werden im folgenden Unterabschnitt genauer unterschieden.

2.5.3 Zwischen- und Endbevorratungen

Die DCAG geht im Falle einer ZBV immer davon aus, dass nach einem überbrückten Zeitraum ein Teil, welches bevorratet war, wieder durch den Lieferanten zu beziehen ist bzw. generell wieder normal durch die Regeldisposition geplant werden kann. Bei einer Endbevorratung muss die DCAG hingegen davon ausgehen, dass nach dem Aufbau der gesamten Endbevorratungsmenge durch einen Lieferanten, dieser in Zukunft keine Teile dieser Art mehr liefern wird. Im Falle der ZBV ist der Zeitraum auf bis zu vier Jahre beschränkt. Bei einer EBV muss davon ausgegangen werden, dass alle zukünftigen Bedarfe für die nächsten 15 – 30 Jahre aus der Bevorratungsmenge gedeckt werden können. Eine Unterscheidung liegt demnach in dem Zeitraum, welche durch eine Bevorratungsaktion überbrückt werden muss.

Ein Lieferant ist im Falle einer EBV berechtigt, seine Werkzeuge und Maschinen für die Teilefertigung abzubauen und zu verschrotten. Die DCAG hätte in einem Fall einer eventuell notwendigen Nachlieferung keinen rechtlichen Anspruch mehr auf eine erneute Lieferung. Bei einer ZBV kann hingegen davon ausgegangen werden, dass nach dem überbrückten

Zeitraum eine Teilelieferung wieder einsetzt und das Teil in der normalen Regeldisposition wieder beschafft werden kann.

Tabelle 2-1: Unterschiede EBV - ZBV

<i>Attribute</i>	<i>EBV</i>	<i>ZBV</i>
Dauer	15/30 Jahre	4 Jahre
Nachlieferung	nein	ja

Die obige Tabelle 2-1 fasst einmal die Unterschiede zwischen EBV und ZBV als Übersicht zusammen.

Im folgenden Abschnitt werden die Gründe für die Bevorratungsaktionen besprochen und genau beschrieben. Dabei erfolgt eine Unterscheidung zwischen EBV- und ZBV-Vorgängen nur dann, wenn es sich um Gründe handelt, die explizit nur für eine der beiden BV-Aktionen gelten.

2.6 Gründe für Zwischenbevorratungen und Endbevorratungen

Die Gründe für die im vorherigen Abschnitt angesprochenen Bevorratungsaktionen sind sehr vielfältig. Damit der Leser einen Eindruck davon gewinnt, welche Gründe dies sind, wird in diesem Abschnitt zuerst die Zusammenarbeit der DCAG mit ihren Lieferanten erläutert und darauf aufbauend werden die Gründe sukzessive durchgegangen und jeweils genauer erläutert.

2.6.1 Zusammenarbeit zwischen Lieferanten und DaimlerChrysler

Die Zusammenarbeit der Lieferanten mit der DCAG basiert auf verschiedenen Grundlagen und Beziehungen. Nicht immer nimmt die DCAG hier die Position des dominanten, preisbestimmenden Abnehmers ein.

Die typische Beziehung, die für die weitere Betrachtung relevant ist, ist dadurch gekennzeichnet, dass Lieferanten als Teilelieferanten die DCAG während der Produktion und in der Nachserie versorgen. Ein typischer Lieferant versorgt die DCAG mit den entsprechenden Teilen während der gesamten Produktion eines Modells und darüber hinaus. Verträge sichern ihm somit die Lieferung zu und erlauben einen Lieferantenwechsel von Seiten der DCAG nur in speziellen Fällen. Teilweise werden in den Verträgen mit den Lieferanten spezielle Mengen festgeschrieben, die dem Lieferanten als Absatzmengen versprochen werden.

Dabei kann der Lieferant allerdings nur ökonomisch arbeiten, wenn genügend Teile von der DCAG in der Phase der Produktion abgefragt werden. Der festgeschriebene Preis der DCAG ist für einen Lieferanten nur dann wirtschaftlich sinnvoll, wenn er durch die entsprechenden

Absatzmengen Gewinn einfahren kann. Sinkt die Menge aber erheblich, verringert sich der Gewinn entsprechend oder aber der Lieferant macht Verluste. Damit der Lieferant mit der Lieferung eines Teiles beauftragt wird, muss er allerdings mit einem geringeren Absatzvolumen des Teils während der letzten Lebenszyklusphasen rechnen. Gerade in der Phase der Nachserie konnte im vorherigen Abschnitt anhand der Abbildung 2-5 gezeigt werden, wie der Absatz sich nach dem Serienauslauf verändert und stark sinkt.

Den betriebswirtschaftlichen Prinzipien folgend, müsste sich die stark negative Mengenveränderung natürlich im Preis niederschlagen. Eine sinkende Menge erzeugt einen höheren Preis. Bevor der Lieferant aber einen höheren, an der niedrigeren Menge orientierten, Preis festsetzen kann, reagiert die DCAG hier proaktiv und versucht den niedrigeren Preis gegenüber dem Lieferanten durchzusetzen.

Eine weitere Beziehung beschreibt die DCAG als einen kleinen Abnehmer eines Teiles bei einem größeren Lieferanten. Hier nimmt die DCAG nur eine untergeordnete Rolle beim Lieferanten ein und ist somit dem Druck eines Lieferanten ausgesetzt. Dies betrifft vor allem Interieur-Teile, die in so geringen Stückzahlen und mit einem so geringen Stückwert verbaut und bezogen werden, dass die DCAG daher beim Lieferanten nur als ein relativ unwichtiger Kunde registriert ist.

Nicht immer existiert eine direkte Beziehung zwischen der DCAG und den Lieferanten. Auch die Lieferanten besitzen ihrerseits wiederum Lieferanten, die sie mit Rohstoffen usw. versorgen. Auf diese Lieferanten hat die DCAG keinen Einfluss.

Diese grundlegenden, allgemeinen Beziehungen zwischen der DCAG und den Lieferanten dient als Grundlage zum Verständnis der nun folgenden Gründe, die BV-Aktionen verursachen.

2.6.2 Grund 1: Lieferanteninsolvenz

Die Insolvenz eines Lieferanten kann zu starken Problemen in der Teileversorgung führen. Dieses Szenario ist durch eine komplette Unfähigkeit des Lieferanten zur Lieferung weiterer Teile gekennzeichnet. Innerhalb einer kurzen Zeitspanne müssen somit alle notwendigen Entscheidungen zu den Mengen der zu bevorratenden Teile getroffen werden.

Eine Insolvenz eines Lieferanten kann ein oder mehrere Teilenummern betreffen. Wenn es möglich ist, noch einen Restbestand für bestimmte Teile vom Lieferanten zu beziehen, so wird dies natürlich gemacht und sich mit diesen restlichen Teilen bevorratet. Hier muss dann entschieden werden, wie lange dieser Restbestand in Bezug zu den prognostizierten, zukünftigen Verbräuchen reichen wird und ob eventuell eine Nachlieferung durch einen neuen Lieferanten nötig sein wird.

Sollte der insolvente Lieferant allerdings gar keine Teile mehr liefern können, muss zwangsläufig nach Alternativen gesucht werden. In diesem Szenario wird es allerdings

schwierig, einen neuen Lieferanten für das Teil zu finden, da gerade in der Phase der Nachserie die Preise für ein Teil von Seiten der DCAG sehr niedrig angesetzt werden und ein neuer Lieferant aus Mengenrunden einen viel höheren Preis ansetzen müsste.

Sofern ein neuer Lieferant gefunden wird, müssen eine Menge von nachträglichen Tätigkeiten durchgeführt werden. Darunter fallen beispielsweise die Überprüfung der Qualität der Teile oder aber die Validation der Produktionsprozesse beim Lieferanten. Durch den großen Umfang von nachgelagerten Tätigkeiten ist dieses Szenario sehr arbeitsintensiv.

2.6.3 Grund 2: Bauteilabkündigung vom Vorlieferanten

Nicht immer erfolgt die Abkündigung eines Bauteils vom Hauptlieferanten, der im direkten Kontakt zu der DCAG steht. Oftmals geschehen solche Abkündigungen auch von Seiten der Lieferanten eines Lieferanten, sprich den Vorlieferanten. Folgende Abbildung 2-6 verdeutlicht diese ‚Kette‘ von Lieferanten.



Abbildung 2-6: Lieferantenkette

Besonders häufig treten diese Fälle bei den so genannten Elektronik-Komponenten auf (bei der DCAG als EE-Komponenten bezeichnet). In diese Teilekategorie fallen alle Elektronikteile, gleich ob von komplexer oder einfacher Bauart. Es sind unter den EE-Komponenten einfache Chips, als auch komplexe Steuergeräte zu finden. In diesem Produktsortiment tritt das betrachtete Szenario einer Bauteilabkündigung (BT-Abkündigung) durch einen Vorlieferanten besonders häufig auf.

Betrachtet man die Lieferantenkette in der Abbildung 2-6, so würde im Falle einer Abkündigung eines Bauteils durch einen Vorlieferanten der EE-Lieferant vom Chiphersteller eine Abkündigung erhalten. Hierfür existieren viele Gründe, wie z.B. der Wechsel der Produktion beim Vorlieferanten von einer Chipserie auf eine nächste. Der EE-Lieferant wird dadurch gezwungen, alle entsprechenden in seinem Sortiment befindlichen Teile, die von dieser Abkündigung betroffen sind, bei der DCAG ebenfalls mit abzukündigen. Die zentrale Frage, die in diesem Fall zu klären ist, ist die Frage nach der Bevorratung der betroffenen Teile. Hier gibt es zwei Möglichkeiten: entweder der Lieferant bevorratet sich oder aber die DCAG führt eine Bevorratung durch und legt die Komponenten auf Lager. Die DCAG kann den Vorlieferanten nicht dazu zwingen sein Produktionsprogramm weiterhin aufrecht zu erhalten.

In diesem Fall der EE-Komponenten ergeben sich darüber hinaus noch andere Schwierigkeiten. Ausgehend von einer beispielhaften Abkündigung eines Chipherstellers, können Chips, die nun bevorratet werden müssten, nicht unendlich lange gelagert werden. Nach Aussagen der Disponenten bei der DCAG tritt nach ca. zwei Jahren eine Korrosion der Chipteile, speziell der Kontakte, ein, die dazu führen kann, dass die Chips unbrauchbar werden. Das heißt also, dass die Lagerfähigkeit der Teile äußerst schwierig ist und hier eine entscheidende Rolle bei der Entscheidungsfindung spielt. Entweder werden die vom Vorlieferanten gelieferten Chips beim Lieferanten durch spezielle Lagerungsmaßnahmen bevorratet oder aber die fertigen EE-Komponenten werden beim Lieferanten oder bei der DCAG bevorratet.

So muss zur Lösung dieser Problematik oftmals eine intensive Zusammenarbeit zwischen Lieferanten und DCAG stattfinden.

2.6.4 Grund 3: Auslauf der vertraglichen Versorgungsdauer

Im Abschnitt 2.5.1 wurde anhand der Lebenszykluskurve eines Teils die gesamte Dauer der Teileversorgung aufgezeigt. So beabsichtigt die DCAG Teile bis zu 15 Jahre nach Auslauf der Serie noch anbieten zu können. In einigen Fällen kann diese Versorgungsdauer sogar bis auf 30 Jahre nach der Serie anwachsen, sofern die Teile Classic Center-relevant sind.

Viele Lieferanten verfügen bei der DCAG allerdings über Verträge, die eine Teilleistung nur für insgesamt zehn Jahre vertraglich vorsehen, beispielsweise die Robert Bosch GmbH. Diese Differenz zwischen der vertraglich festgelegten Dauer zur Lieferung der Teile und der von der DCAG veranschlagten Lieferdauer nach Serienende muss durch eine Zwischen- bzw. Endbevorratung teilweise überbrückt werden.

In diesem Falle liegt die Bevorratung natürlich bei der DCAG. Hier müssen vor dem Ende der Versorgungsdauer alle notwendigen Maßnahmen getroffen werden, um die restliche Versorgung der Teile über die verbleibende Dauer zu gewährleisten.

2.6.5 Grund 4: Wirtschaftliche Aspekte – Auslaufmanagement

Die Aufgaben des Auslaufmanagements sind sehr vielfältig, konzentrieren sich aber auf ein Hauptaugenmerk, das auf der frühzeitigen Erkennung auslaufender Teile liegt. Ein Teil gilt dann als auslaufend, wenn es sich in Modellen der DCAG befindet, die in Kürze nicht mehr produziert werden. Dabei wird durch das Auslaufmanagement strategisch das Modellprogramm der DCAG beobachtet und schon einen langen Zeitraum vor einem bevorstehenden Modellauslauf werden alle die Teile bestimmt, die in keinem weiteren Modell verbaut werden und demnach mit auslaufen.

Der wirtschaftliche Aspekt beim Auslauf eines Teiles wurde bereits im Abschnitt 2.5.1 kurz angesprochen. Nach Ablauf der Serienproduktion muss der Preis eines Teiles theoretisch zwangsläufig steigen, da der Absatz stark einbricht. Der ursprüngliche Preis, der an die in der Serie produzierte Stückzahl gekoppelt war und auch nur in diesem Verhältnis für den Lieferanten Gewinn erwirtschaftet, kann in der Nachserie nicht mehr angesetzt werden. Das Auslaufmanagement ermittelt im Vorfeld eines Modellauslaufs alle relevanten Teilenummern und versucht dann auch auf einem proaktiven Weg den Lieferanten zu einer ZBV/EBV zu bewegen. Dabei ist die Proaktivität dadurch gekennzeichnet, dass man sehr früh versucht auf den Lieferanten zuzugehen und durch diesen Zeitvorsprung den Lieferanten zu einem niedrigen Preis bei einer ZBV/EBV zu bewegen. Es handelt sich hier also um rein betriebswirtschaftliche Gründe.

Im Jahr laufen ca. fünf bis sechs solcher Ausläufe bei der DCAG. Das Wichtigste bei diesen Vorgängen ist die rechtzeitige Erkennung, sodass, und das ist das oberste Ziel des Auslaufmanagements, eine Bevorratung zu aktuellen Serienkonditionen durchgeführt werden kann.

2.6.6 Grund 5: Verlagerungsmanagement

Ein weiterer Grund für Bevorratungsaktionen sind so genannte Verlagerungsaktionen. Sie sind typisch, wenn Maschinen/Fertigungsanlagen einen Standort wechseln.

Sehr typisch ist das Szenario der Verlagerung einer Maschine in dem Falle eines Produktionswechsels auf einer Produktionsstraße. In den heutigen Werken wird oft nicht mehr einfach nur ein Modell der DCAG gefertigt, vielmehr findet ein stetiger Modellwechsel und somit auch Wechsel der Produktionsanlagen statt. Die Lieferanten fertigen hierbei direkt auf ihren eigenen Produktionsstraßen neben den DCAG-eigenen Produktionsstraßen, um so die Transportkosten zu minimieren. Diese lieferanteneigenen Produktionsstraßen im Werk müssen beim Modellwechsel und dem damit verbundenen Umbau einer DC-Produktionsstraße ebenfalls abgebaut werden und an einem anderen Ort wieder aufgebaut werden, damit eine Teileproduktion weiter erfolgen kann. Den Abbau und Wiederaufbau der Produktionsanlagen nennt man Verlagerung. Die damit verbundene Ausfallzeit der Maschine beeinträchtigt somit auch die Produktion des Teils, die nach Serie auch weitergehen muss. Die Überbrückung dieser Ausfallzeit geschieht durch ZBV/EBV-Vorgänge.

Es existieren dabei verschiedene Szenarien, die sich dadurch unterscheiden, ob der Lieferant oder die DCAG eine Maschine verlagert und somit die Ausfallzeit verursacht. Je nachdem, wer der Verursacher ist, muss die Bevorratung dann bei der DCAG oder dem Lieferanten durchgeführt werden.

Tabelle 2-2: Szenarien bei Verlagerungen

Verlagerung	Von	Zu	Ort der Teilelagerung
Szenario 1	DaimlerChrysler	Lieferant	DaimlerChrysler
Szenario 2	Lieferant	Lieferant	Lieferant (DaimlerChrysler)
Szenario 3	intern beim Lieferanten		Lieferant (DaimlerChrysler)

Die obige Tabelle 2-2 unterscheidet nochmals drei Szenarien, in denen jeweils die Verlagerung genauer spezifiziert ist – wer veranlasst eine Verlagerung und wer muss dabei eine Bevorratung vornehmen.

Eine sehr typische Verlagerung ist hierbei das Szenario 1, in dem eine Produktionsstraße bzw. eine Maschine der DCAG zu einem Lieferanten verlagert wird. Dies geschieht beispielsweise bei Blechteilen sehr oft. Hier existieren strategische Partner zur Lieferung der Blechteile nach Serie. Diesen wird die gesamte Produktionsanlage übergeben. Damit entlastet sich die DCAG von der Produktion dieser Teile und kann den Platz für neue Maschinen und Fertigungsanlagen nutzen, welche für das neu zu produzierende Modell gebraucht werden. In diesem Szenario muss die DCAG die Kosten für die Bevorratung tragen und die Teile lagern.

Im Falle einer internen Verlagerung einer Produktionsanlage beim Lieferanten trägt natürlich dieser die Verantwortung für die Teilverfügbarkeit und muss daher selbst eine Bevorratung durchführen. Die DCAG sollte in diesem Falle eigentlich keine Auswirkungen spüren. Da jedoch auch hier eine gewisse Sicherheit herrschen muss, wird bei bestimmten Teilen auch von Seiten der DCAG eine Bevorratung durchgeführt. Daher erscheint hier in der Tabelle auch die DCAG als ein Ort der Teilelagerung.

2.6.7 Grund 6: Beschaffungskostenoptimierung

Ein wirtschaftlicher Grund zur Zwischen- bzw. Endbevorratung ist die so genannte Beschaffungskostenoptimierung, kurz BKO genannt. Sie versucht auf Basis einer Sortimentsbereinigung Teilenummern, die nur noch sehr selten bestellt werden, aus dem Sortiment zu nehmen. Dabei betrifft dies das Sortiment der DCAG und das Sortiment des Lieferanten. Auch der Lieferant kann bei einer eigenen Sortimentsbereinigung auf die DCAG zukommen und diese zu einer ZBV/EBV auffordern.

Nach der Überprüfung aller notwendigen Daten und der Berechnung, ob diese ZBV/EBV wirtschaftlich sinnvoll ist, wird meist nur eine einmalige Bestellung durchgeführt und damit alle notwendigen Teilebedarfe für die veranschlagte Zeitdauer gedeckt. Die BKO betrifft meist nur sehr wenige Teile in sehr geringen Stückzahlen.

2.6.8 Weitere Gründe

Nachdem in den vorherigen Unterabschnitten zahlreiche Gründe für ZBV/EBV-Vorgänge aufgeführt worden sind, existieren daneben noch weitere. Zwei davon sind beispielsweise:

- Die vertraglich festgelegte Menge in Serie konnte nicht eingehalten werden. Der Lieferant kündigt daraufhin den Versorgungsvertrag für ein Teil.
- Es erfolgen Konstruktionsänderungen in der Serie, bei denen ein Teil durch ein anderes Teil ersetzt wird. Das alte Teil muss daraufhin bevorratet werden.

Es gibt sicherlich noch andere Gründe, die zu einer ZBV/EBV führen können. Jedoch soll die einzelne Betrachtung der Gründe hier enden. Zur besseren Orientierung des Lesers sollen an dieser Stelle nochmals die Gründe konsolidiert dargestellt werden. Folgende Abbildung 2-7 illustriert eine Matrix:

Matrix Muss-Kann-EBV/ZBV		
	<i>Muss</i>	<i>Kann</i>
<i>EBV</i>	Insolvenz BT-Abkündigung	BKO wirtschaftlich
<i>ZBV</i>	Verlagerung	wirtschaftlich

Abbildung 2-7: Matrix Muss-Kann-EBV/ZBV

Zu sehen sind die vier verschiedenen Felder der Matrix, deren einzelne Abschnitte durch die Zuordnung über die Kriterien EBV und ZBV und den beiden Kriterien Muss und Kann erfolgt. In der Matrix wurde eine Unterscheidung zwischen den einzelnen Bevorratungsaktionen durchgeführt. Dabei wurden alle wirtschaftlich motivierten Aktionen unter dem Stichwort ‚wirtschaftlich‘ zusammengefasst. Die weiteren Fälle sind daran zu erkennen, dass sie in Klartext in der Matrix mit aufgenommen wurden, wie z.B. ‚Verlagerung‘.

Im Falle einer Muss-Bevorratung gibt es keine andere Möglichkeit, als sich zu bevorraten. Im Falle einer Kann-Bevorratung muss eine Bevorratung nicht unbedingt durchgeführt werden. Hier kann es aber vorteilhaft sein, sich zu bevorraten.

Die Unterteilung zeigt auch einen guten Überblick darüber, wann die DCAG proaktiv vorgehen muss/sollte. Alle Kann-Bevorratungen sind durch einen wirtschaftlichen Aspekt gekennzeichnet. Hier wird also versucht, durch die Erzielung eines besseren Preises in der Nachserie eine Reduzierung der Kosten für die Teilebeschaffung durchzuführen.

2.7 Kapitelabschluss

Dieses Kapitel hat dem Leser einen umfassenden Überblick über Prozesse, Geschäftsprozesse, Workflows und Workflowmanagementsysteme vermittelt. Weiterhin ist geklärt worden, inwieweit die Prozessmodellierung innerhalb der Softwareentwicklung eingesetzt und wie sie genutzt werden kann.

Durch einen Einblick in das allgemeine Verhalten von Teilen im Lebenszyklus und in die Zusammenarbeit der DCAG und den Lieferanten wurde ein kurzer Überblick geschaffen, der die Gründe, welche zu Bevorratungsaktionen führen können genauer erklärt.

Da die Vorgänge von Bevorratungen sehr problematisch sind und mit vielen Unsicherheiten behaftet sind, wurde im GLC die Entscheidung getroffen ein Projekt zu gründen, welches bestimmte Probleme hierbei lösen soll. Der Inhalt des Projektes wird im nächsten Abschnitt erklärt.

3 Beschreibung des Projektes “Integrated Process for Optimized Stocking”

Dieses Kapitel wird dem Leser eine umfassende Einführung zu dem Projekt IPOS geben. Ausgehend von einer allgemeinen Einführung zum Projekt, werden die Ziele des Projektes, die Strukturierung des Projektes und die Vorgängerprojekte erläutert.

3.1 Allgemeine Einführung

Das Projekt IPOS wurde durch die Beantragung mittels eines entsprechenden Projektantrages im Jahr 2004 begründet [Mau04]. Als Grundintention und als oberste Prämisse für das Projekt steht die Verbesserung der Abläufe für End- und Zwischenbevorratungen im Zuge der Harmonisierung der Arbeitsabläufe der Disponenten.

Bevor das Projekt IPOS beantragt wurde, konnten in mehreren Studien verschiedene Potenziale nachgewiesen werden [Kast04], die sich ergeben würden, wenn ein genormter Prozess zur Bevorratung von Teilen innerhalb des GLC's genutzt werden würde. Unter diesen Potenzialen findet sich in erster Linie die Reduzierung einer Kapitalbindung und einer Verringerung des Schrottrisikos. Dies wurde durch entsprechende Werte nachgewiesen und somit die Bedeutung aufgezeigt, die eine Verbesserung der Methoden haben wird.

Innerhalb der Studien wird ebenfalls gezeigt, dass die Disponenten in der DCAG die unterschiedlichen Bevorratungsprozesse äußerst unterschiedlich durchführen. Hier wird in der Studie davon ausgegangen, dass viele Disponenten nicht das gesamte Spektrum an Informationsquellen nutzen, um eine Bevorratung durchzuführen, sondern nur einen Teil davon. Durch dieses unterschiedliche Vorgehen werden die Prozesse, die hinter einer Bevorratungsaktion stehen, natürlich intransparent und es kann im Nachhinein nur sehr schwer nachvollzogen werden, warum welche Entscheidung getroffen wurden.

Um das gesamte Wesen einer Bevorratung transparent zu gestalten, die Disponenten besser zu unterstützen und ein einheitliches Softwaresystem bereit zu stellen, wurde das Projekt IPOS gegründet.

3.2 Ziele des Projektes

Die im vorherigen Unterabschnitt bereits erwähnte Studie hat gezeigt, dass sich ein hohes Potenzial an Kostenersparnissen und Ersparnissen bei der Kapitalbindung in Form von gebundener Fläche und an gelagerten Teilen ergeben kann, wenn die Disponenten dazu gebracht werden, in einem standardisierten Prozess alle nötigen und zusätzlichen

Informationen für eine Bevorratung zu berücksichtigen. Dies ist auch das Hauptziel, welches durch IPOS verfolgt wird. Daneben existieren natürlich noch weitere Ziele, die an dieser Stelle konsolidiert aufgeführt werden sollen und eine Auflistung in [Mau05] ergänzen:

- Ein durchgängiger, standardisierter Prozess soll für alle Disponenten gelten.
- Abwicklung einer Bevorratungsaktion in einem zentralen Softwaresystem, welches den Disponenten in einem Ablauf zu einer optimalen Bevorratungsmenge führt.
- Bündelung und Historisierung von allen Daten, die eine Bevorratungsaktion betreffen.
- Reduzierung von parallel ablaufenden Prozessen in unterschiedlichen Abteilungen.
- Lernen aus vergangenen Bevorratungen und Fehlern.
- Nachweis der Wirtschaftlichkeit des Gesamtprojektes IPOS.

Aus den oben genannten Zielen wird deutlich worin das Hauptaugenmerk des gesamten Projektes liegt: in einem integrativen Prozess zur Steuerung der Disponenten. Integrativ soll der Prozess sein, da er viele Softwaresysteme ansprechen soll und viele Informationsquellen integrieren soll. Weiterhin soll der Prozess die Disponenten steuern, aber nicht bevormunden. Die Disponenten sollen durch einen vorgegebenen Weg hin zu allen nötigen Entscheidungen geleitet werden und soweit es möglich ist, unterstützt werden. Er soll eine möglichst breite Informationsbasis erhalten, auf deren Grundlage er die Entscheidung treffen kann. Die letztendlichen Entscheidungen bleiben dem Disponenten vorbehalten – sie können und sollen nicht durch das System getroffen werden.

Neben diesen Zielen besteht ein weiteres Ziel in der Einführung eines Softwaresystems, welches als Umsetzung der oben genannten Ziele gilt. Dieses Softwaresystem wird äquivalent zum Projekt IPOS ebenfalls IPOS benannt werden. Innerhalb dieses Systems sollen die Disponenten alle Möglichkeiten für einen standardisierten Bevorratungsprozess haben, um eine möglichst gute, wenn möglich optimale, BV-Menge zu ermitteln. Das Ziel der Diplomarbeit ist die Entwicklung und Konzeptionierung eines solchen Softwaresystems.

Die oberste Prämisse des Projektes IPOS ist die Reduzierung der Risiken in der fehlerhaften Mengenbestimmung in den BV-Aktionen.

3.3 Strukturierung des Projektes

Generell kann das Projekt IPOS in zwei große Bereiche unterteilt werden. Einerseits wird sich mit den zu einer Bevorratungsaktion gehörenden Prozessen und der dahinter liegenden Theorie beschäftigt. Andererseits wird versucht ein großes und komplexes Softwaresystem aufzubauen, welches die an einer Bevorratungsaktion beteiligten Mitarbeiter von Beginn einer BV-Aktion an leitet.

Es ist klar, dass der erste Bereich vor dem zweiten Bereich behandelt werden sollte. Sofern eine theoretische Grundlage zur Schaffung einer neuen Anwendungssoftware unzureichend

oder nur teilweise erfasst wurde, kann das letztendliche Produkt nicht allen Anforderungen entsprechen. Daher wird im Projekt IPOS versucht, die theoretischen Grundlagen für BV-Aktionen mit zu erfassen, zu dokumentieren und die Grundlage zur Schaffung eines konsistenten Softwareproduktes zu legen, welches allen Anforderungen gerecht werden kann.

Das Projekt ist darüber hinaus in sechs große Module unterteilt. Dazu gehören nach [Mau04]:

- Modul Forecasting (Prognose)
- Modul Fragenkatalog
- Modul WIRE (Wirtschaftlichkeitsrechnung)
- Modul Eskalation
- Modul Dokumentation
- Modul Monitoring (Überwachung)

Das Module Forecasting (Prognose) umfasst die Bemühungen für den Aufbau eines integrierten Prognosemoduls, mit dessen Hilfe die Disponenten fähig sein sollen, sich ohne Zeitverzögerung Prognose zu generieren und diese entsprechend ihren Bedürfnissen verändern zu können. Innerhalb des Moduls ist auch die Weiterentwicklung eines Prognosemodells mit dem Forschungszentrum der DCAG in Ulm zu finden.

Das Modul Fragenkatalog beschreibt die Bemühungen, eine Sammlung aller notwendigen Fragen zu erstellen, welche von einem Disponenten während einer BV-Aktion zu beantworten sind. Ausgehend von verschiedenen Vorschlägen für den Fragenkatalog, sind hier schon erste Ansätze für die zur Verfügung zu stellenden Informationen zu finden.

Im Modul WIRE gilt die Zielstellung, die Wirtschaftlichkeitsrechnung völlig in das zukünftige Softwaresystem IPOS zu integrieren. Darüber hinaus sind weitere Ziel definiert, die beispielsweise die Realisierung einer Flächenberechnung umfassen. Hier soll anhand der einfachen Eckdaten einer BV-Aktion eine erste Rechnung stattfinden, wie viel Fläche von der BV-Menge belegt werden würde.

Das Modul Eskalation beschreibt die abschließende Findung einer Entscheidung für eine Bevorratungsaktion, wenn der wirtschaftliche Umfang eine gewisse Wertgrenze überschreitet. Dabei existieren verschiedene Gremien mit bestimmten Zuständigkeitsstufen, welche nach Begutachtung einer BV-Aktion entweder der Menge zustimmen oder die Menge ablehnen können.

Eine genaue Erstellung einer Dokumentation der im System IPOS hinterlegten Prozesse, als auch der theoretischen Hintergründe, ist das Ziel des Moduls Dokumentation.

Das letzte Modul beschreibt alle Funktionen, welche in einer Beziehung zu den Themen Kontrolle und Überwachung der BV-Aktionen steht. Es wird als Monitoring (Überwachung)

bezeichnet und soll in einem ersten Schritt eine Sammlung von zu kontrollierenden und überwachenden Daten erstellen, die beispielsweise für eine Eskalation wichtig sind.

Obwohl IPOS einen neuen Ansatz der Integration verfolgt, existieren im Vorfeld des Projektes noch einige Vorgängerprojekte, die immer nur sehr spezifische Aspekte einer BV-Aktion untersucht haben und Versuche unternommen haben, die Disponenten hier zu unterstützen. Diese Vorläuferprojekte sollen im kommenden Abschnitt erläutert werden.

3.4 Vorgängerprojekte

Schon im Vorfeld des Projektes IPOS wurden verschiedene Maßnahmen und Softwaresysteme eingeführt, die den Prozess einer Bevorratung erleichtern sollen und durch geeignete Mittel die Disponenten unterstützen sollen. Einige dieser Softwaresysteme werden durch die Einführung von dem zukünftigen Softwaresystem IPOS vollständig ersetzt, andere bleiben parallel dazu bestehen.

Das Softwaresystem IPOS soll Funktionalitäten bestimmter proprietärer Softwaresysteme mit übernehmen und Funktionen anderer Softwaresysteme nutzen und diese über verschiedene Schnittstellen nutzen.

Zu diesen Softwaresystemen gehören beispielsweise:

- Inventory Management Documentation System, kurz IMDS
- Wirtschaftlichkeitsrechnung, kurz WiRe
- **Javadatenänderung auf Unix**, kurz JDAX
- EBV-Workflow
- Prognosetool

Diese Liste von Systemen ist nicht komplett, enthält aber die wichtigsten Softwaresysteme, die im Umfeld von IPOS eine Rolle spielen. Damit der Leser einen Eindruck davon bekommt, welche Funktionalitäten die Systeme besitzen, sollen sie im Folgenden sukzessive kurz erläutert werden.

Die Funktionalität des IMDS wird im zukünftigen Softwaresystem IPOS vollständig hinterlegt werden. Es handelte sich beim IMDS um ein Softwaresystem zur Dokumentation von Bevorratungsaktionen. Das System bietet den Disponenten die Möglichkeit, alle Daten, welche die Entscheidungsgrundlage bilden, und die letztendliche Entscheidung in einem zentralen System zu hinterlegen. Das System wurde eingeführt, um durchgeführte BV-Aktionen und damit verbundene Entscheidungen nachvollziehbar zu machen.

Bei dem System WiRe handelt es sich um ein auf Basis von Microsoft Access erstelltes Lösung zur Berechnung eines Kapitalwerts, welcher aussagt, ob eine Bevorratungsaktion wirtschaftlich sinnvoll ist oder nicht. Dieser Wert dient dem Disponenten bei Bevorratungsaktionen mit wirtschaftlichem Hintergrund als Orientierung, ob eine Bevorratung

durchgeführt werden sollte oder nicht. Hierin wird beispielsweise auch das Schrottpotenzial monetär berücksichtigt. Als strategisches Ziel soll dieses System mit im Softwaresystem IPOS integriert werden.

Das Softwaresystem JDAX ist ein auf Basis von Java erstelltes Softwaresystem, welches zur Auswertung von unterschiedlichsten Sachverhalten dient. Im Umfeld des Projektes IPOS ist hier aber die Kommunikationsmöglichkeit des Systems von besonderem Interesse. Über das System JDAX können die unterschiedlichen Abteilungen mit der Abteilung der Technik kommunizieren. Die Technik ist für viele verschiedene Aspekte im GLC verantwortlich, wie z.B. die Verplanung und Planung der Fläche im GLC und überprüft jede zu einer EBV vorgesehene Teilenummer. Bestimmten Filterkriterien folgend, kann die Technik BV-Aktionen ablehnen oder bestätigen. Der EBV-Workflow beschreibt in diesem Zusammenhang den gesamten Prozess des Einpflegens der Daten in das JDAX-System und die Abstimmung zwischen Technik und Disponenten. Es handelt sich bei dem EBV-Workflow also nicht um ein Softwaresystem, sondern mehr um einen grundlegenden Prozess.

Das Prognosetool, welches wie das WiRe-System ein Softwareprodukt auf Basis von Microsoft Access ist, unterstützt den Disponenten durch die Erstellung einer Langfristprognose für eine Teilenummer. Grundlage für diese Erstellung ist ein Prognosemodell, welches in einer Kooperation mit dem DC-internen Forschungszentrum in Ulm entwickelt worden ist. Das Prognosetool soll im Softwaresystem IPOS vollständig integriert werden.

Weitere Maßnahmen, die ohne Unterstützung eines Softwaresystems durchgeführt werden, sind einfache Arbeitsanweisungen an die Disponenten. In einigen Anweisungen werden somit Ausschnitte aus dem komplexen Prozess beschrieben, wie Endbevorratungen durchzuführen sind [Korn05]. Allerdings fehlt hier eine umfassende Übersicht, wann welche Tätigkeiten in welchem Umfang durchzuführen sind und wo die spezifischen Schnittstellen zu anderen Bereichen liegen und welche Softwaresysteme hierbei zu gebrauchen sind. Es existieren keine umfassenden Darstellungen des Gesamtprozesses, welches daher als ein Ziel des Projektes IPOS definiert wird.

3.5 Zusammenfassung

Das Kapitel hat den Inhalt, die Struktur und die Ziele des Projektes IPOS dargestellt. Es hat ebenfalls einen Überblick zu den bisherigen Maßnahmen und Softwareprojekten vor der Gründung des Projektes IPOS verschafft. Durch eine genaue Auflistung der Schwachstellen in der Arbeit der Disponenten wird hier deutlich, warum das Projekt IPOS geschaffen wurde und mit welchen Anforderungen sich das Projekt zu beschäftigen hat.

4 Problemanalyse und Anforderungsanalyse

Dieser Abschnitt wird die grundsätzlichen Probleme, mit denen sich das Projekt IPOS beschäftigt erläutern. Dabei werden die Probleme der Mengenbestimmung, der Kapital- und Flächenbindung und des Schrottpotenzials zuerst erläutert. Danach folgt die Beschreibung von allgemeinen Problemen. Eine Auflistung der Anforderungen an ein zukünftiges Softwaresystem erfolgt im vorletzten Unterabschnitt, bevor mit der Abstraktion der Anforderungen das Kapitel abgeschlossen wird.

4.1 Problem der Mengenbestimmung

Die Bestimmung der Menge innerhalb einer jeden Bevorratungsaktion ist die wichtigste Entscheidung. Sie wird vom Disponenten getroffen und auch von diesem verantwortet. Dem Disponenten stehen zur Mengenbestimmung verschieden viele Tools zur Unterstützung bereit, darunter das Prognosetool, die WiRe und alle weiteren im Abschnitt 3.4 beschriebenen Softwaresysteme.

Um einen grundsätzlichen Eindruck darüber zu gewinnen, wie sich die Bedarfe eines Teils in der Zukunft entwickeln werden, setzt ein Disponent das Prognosetool ein. Dieses Tool kann dem Disponenten für eine Teilenummer, basierend auf einem Prognosemodell und den Verbräuchen der letzten Jahre, eine Langfristprognose aufstellen. Die sich aus dieser Langfristprognose ergebene kumulierte Menge wird als erste Grundlage für die weiteren Prozesse in den BV-Aktionen genutzt.

Die Problematik, die sich nun zur endgültigen Mengenbestimmungen ergibt, beruht auf dem Mitspracherecht von verschiedenen vielen Abteilungen der DCAG. Bevor eine endgültige Entscheidung über eine Bevorratungsmenge (kurz BV-Menge) getroffen wird, müssen unterschiedlich viele Abteilungen, wie z.B. das Marketing oder aber die Werksplanung, konsultiert werden. Diese können basierend auf eigenen Daten einen Mengenvorschlag für eine BV-Menge unterbreiten. Der Disponent muss letztendlich alle Vorschläge so gewichten und berücksichtigen, dass er zum Schluss eine BV-Menge festlegt, die allen Anforderungen der anderen Abteilungen gerecht wird. Letztendlich hat er als Bestandsverantwortlicher aber die Verantwortung über die BV-Menge und wird im Falle einer Unter- oder Überdeckung zur Verantwortung gezogen, während andere Abteilungen hier nicht belangt werden können.

Jede BV-Menge ist natürlich mit einem Grad der Unsicherheit behaftet. Obwohl Prognosen eingesetzt werden, kann nicht garantiert werden, dass der Absatz eines Teiles nicht einbricht und damit die BV-Menge völlig überdimensioniert ist oder im anderen Falle der Absatz plötzlich unerwartet steigt und somit zu wenig Teile auf Lager liegen. Nicht immer kann dabei mit künstlichen Mitteln eine Absatzregulierung (beispielsweise eine Preiserhöhung bei

plötzlich steigendem Absatz) herbeigeführt werden. Zwar existieren einige Instrumente, aber diese sind nicht immer wirksam. Oft wird auch der entscheidende Zeitpunkt bei solchen Problematiken nicht rechtzeitig erkannt, sodass Gegenmaßnahmen nur noch verzögert und mit verminderter Wirkung eingesetzt werden können.

Mit jeder BV-Menge, welche auf Lager liegt, ist eine Kapitalbindung verbunden, die im folgenden Unterabschnitt genauer erläutert wird.

4.2 Problem der Kapitalbindung und Flächenproblematik

Die Kapitalbindung findet bei den Zwischen- und Endbevorratungen in zwei Formen statt: einerseits wird das Kapital in Form von Teilen, welche auf Lager liegen, gebunden, andererseits wird mit den auf Lager liegenden Teilen auch Fläche gebunden, die ebenfalls Kapital darstellt. Diese Problematik wird weiterhin durch das Ziel verschärft, dass man versucht eine möglichst hohe Verfügbarkeit der Teile zu erreichen.

Es ergeben sich somit vier verschiedene Faktoren, die jeweils paarweise konträr wirken:

1. Teilemenge
2. Teilverfügbarkeit
3. Gebundene Fläche
4. Restliche Fläche

Die beiden Punkte eins und zwei bilden hierbei ein Paar und die beiden Punkte drei und vier ein zweites Paar in der folgenden Betrachtung.

Würde in einem Vorgang der ZBV/EBV lediglich die Teilverfügbarkeit als oberste Prämisse/Zielvorgabe gelten, so könnte man mit einer hohen Bevorratungsmenge an Teilen eine sehr hohe Teilverfügbarkeit gewährleisten. Das Ziel einer hohen Teilverfügbarkeit ist natürlich erstrebenswert, aber dennoch nicht auf diesem Weg zu erreichen. Denn hier greifen die Restriktionen, die aus dem Paar der Punkte drei und vier entstehen. Eine bestimmte Teilemenge bindet demnach auch eine gewisse Fläche im Lagerwerk. In Abhängigkeit der Teilegröße kann auch bei einer niedrigen Teilemenge eine große Fläche gebraucht werden, so z.B. bei Karosserieteilen. Diese Fläche muss in der Betrachtung der Bevorratungsmenge mit beachtet werden, denn die gebundene Fläche ist natürlich ein großer Kostenfaktor. Je Quadratmeter existieren lagerortabhängige, teilweise auch länderabhängige, Kostensätze, die bei der Betrachtung einer Bevorratung nicht außer Acht gelassen werden. In einer Wirtschaftlichkeitsrechnung werden solche Faktoren mit einbezogen.

Weiterhin verfügt das GLC nur über eine beschränkte Lagerfläche, bei welcher zu der operativen Lagerhaltung, die strategische Lagerhaltung der bevorrateten Teile mit

hinzukommt. Da die verfügbare Fläche auf absehbare Zeit auch nicht erweitert wird, muss hier mit einer starken Restriktion gerechnet werden.

Große Bevorratungsaktionen, z.B. im Millionen-Euro-Bereich, werden bei der DCAG durch verschiedene Instanzen validiert. Allerdings ist auch hier immer wieder der Disponent in der Pflicht, die veranschlagte Menge zu untermauern bzw. zu rechtfertigen, selbst wenn ein Gremium einer BV-Menge zugestimmt hat.

Ein weiteres Problem, welches auch monetär ausgedrückt werden kann, ist das Schrottrisiko, welches im kommenden Abschnitt separat betrachtet werden soll.

4.3 Problem des Schrottpotenzials

Als Schrottpotenzial wird bei der DCAG die Teilemenge einer BV-Menge verstanden, die nach dem Ablauf des Zeitraumes einer Teileversorgung nicht mehr abgesetzt werden kann und daher nach dem Ablauf einer Versorgungsdauer verschrottet werden muss. Das bedeutet, dass bei diesem Teil eine Überdeckung existiert und damit Kapital in Form von Teilen und gebundener Fläche vorliegt. Hier kann von „sunken costs“, also versunkenen Kosten, gesprochen werden. Die Teile, die durch die DCAG beschafft worden sind, liegen zwar auf Lager, aber es ist hierbei abzusehen und davon auszugehen, dass sie nicht mehr abgesetzt werden können.

Wenn dieser Fall eintritt, werden bei im GLC die notwendige Konsequenz gezogen und eine Verschrottung geplant. Dabei wird das Teil durch eine Verschrottungsaktion aus dem Lager geräumt und danach verschrottet. Damit wird die gebundene Fläche wieder frei gegeben.

Für diese Verschrottungsaktion muss weiteres Budget zur Verfügung stehen, welches bei der DCAG pro Jahr und je Abteilung beschränkt ist. Ziel einer jeden Abteilung ist natürlich jegliche Verschrottungsaktionen zu vermeiden.

Die einzige Möglichkeit eine Verschrottung innerhalb von BV-Aktionen zu vermeiden ist die möglichst beste Bestimmung der BV-Menge. Je ungenauer die Bestimmung der Menge erfolgt, desto höher ist das Schrottrisiko.

Dieses Risiko wird innerhalb der DCAG mit einem konstanten Satz in der Berechnung der Wirtschaftlichkeit einer BV-Aktion mit berücksichtigt. Damit wird die Unsicherheit bei der Bestimmung der BV-Menge auch monetär berücksichtigt. Je nach Länge einer BV-Aktion wird der sogenannte Schrottkostensatz immer weiter angehoben. Je länger eine BV-Aktion dauern soll, desto unsicherer ist der Absatz in der Zukunft und desto höher ist dementsprechend auch das Schrottrisiko.

4.4 Allgemeine Probleme

Neben den bisher aufgeführten theoretischen Problemen, existieren vor allem auch die Probleme in der Arbeit der Disponenten, die eine BV-Aktion durchzuführen haben. Letztlich ist es das Hauptziel des Projektes IPOS durch eine Standardisierung der Abläufe und die Unterstützung der Arbeit bei den Aktivitäten für eine BV-Aktion, den Disponenten so gut wie möglich zu unterstützen und ihn zu leiten.

Die grundsätzlichen Probleme, die sich momentan ergeben und damit das Entstehen des Projektes IPOS erst bedingen, sollen an dieser Stelle nochmals konsolidiert werden. Sie sind die Grundlage, auf der eine Lösungsfindung stattfinden kann [Mau06].

- Es existiert kein umfassendes System, welches den Disponenten alle notwendigen und zusätzlichen Informationen zur Verfügung stellt.
- Disponenten verfügen nicht über das Wissen, welcher Umfang von Softwaresystemen ihnen zur Unterstützung in einer BV-Aktion zur Verfügung steht.
- Disponenten nutzen kein standardisiertes Vorgehen zur Bestimmung einer Bevorratungsmenge. Oftmals werden Maßnahmen, die zu einer besseren Mengenbestimmung führen können, nicht berücksichtigt.
- Disponenten müssen zur Bestimmung der optimalen Menge viele unterschiedliche Softwaresysteme konsultieren. Es herrscht eine heterogene Datenbasis durch die Verwendung von unterschiedlichen Systemen. Viele Disponenten nutzen daher viele unterschiedliche Dokumente mit sehr unterschiedlichen Formaten zur Ablage von Informationen.
- Viele Abteilungen bevorzugen kleine, sehr spezielle Softwareprodukte. Die Einführung eines großen Softwareproduktes wird schwierig werden, muss aber erfolgen, um die Heterogenität zu beseitigen.
- Entscheidungen können wenig oder gar nicht nachvollzogen werden. Die Gründe, warum eine Entscheidung wie gefällt wurde, sind nicht immer dokumentiert.
- Es existiert keine Automatisierung. Bestimmte Arbeitsabläufe müssen vom Disponenten händisch durchgeführt werden. Eine Arbeitserleichterung existiert in keinem Bereich.
- Viele Prozesse sind kompliziert, da viele Prozessteilnehmer aus unterschiedlichen Abteilungen daran beteiligt sind und die Disponenten die Kommunikation, als auch die Zusammenarbeit koordinieren müssen.
- Die BV-Aktionen können unter Umständen sehr lange dauern.
- Es existiert keine zentrale Instanz zur Leitung und Kontrolle der BV-Aktionen. Erst wenn schwerwiegende Fehler auftreten, wird eine Überprüfung durchgeführt. Einfach zu verhindernde Fehler werden durch keine Instanz überprüft und frühzeitig erkannt.

- Nur sehr wenige Prozesse sind in irgendeiner Form dokumentiert. Es existiert zwar eine Dokumentation zur Vorgehensweise beim Ende einer Regeldisposition, allerdings fehlen darin sehr viele Informationen.
- Die Ermittlung der optimalen BV-Menge ist sehr schwierig und sehr langwierig.

Diese einfache Sammlung von allgemeinen Problemen beschreibt die Situation, in der das Projekt IPOS entstanden ist. Aus den theoretischen und den allgemeinen Problemen sind die Anforderungen an das zukünftige Softwaresystem herauszuarbeiten, was im nächsten Unterabschnitt passieren soll.

4.5 Anforderungen an das zukünftige Softwaresystem

Zu den aufgeführten generellen Problemen im vorherigen Unterabschnitt kommen im Projekt IPOS noch die Anforderungen an das zukünftige Softwaresystem aus vielen unterschiedlichen Quellen und Instanzen der DCAG hinzu. In Gesprächen mit den Disponenten hat der Autor leicht unterschiedliche Anforderungen an ein System herausfinden können. Darunter sind verschiedene Anforderungen zu finden, die entweder die Benutzerfreundlichkeit betreffen, oder aber die Anforderungen nach Informationen beschreiben. Weiterhin sind durch unterschiedliche Teamleiter der DCAG unterschiedliche funktionale Anforderungen an das spätere Softwaresystem IPOS gestellt worden.

Die Anforderungen sollen hier gegliedert nach theoretischen, funktionalen, informellen Anforderungen und Anforderungen an die Benutzerfreundlichkeit gegliedert aufgelistet werden. Die Auflistung orientiert sich teilweise an der Quelle [IPOS05], wird aber durch die Arbeit des Autors noch stark ergänzt und vervollständigt.

4.5.1 Theoretische Anforderungen

Die theoretischen Anforderungen an das Projekt IPOS und an das Softwaresystem sind nahezu identisch.

- Es muss ein Gesamtprozess für alle Arten von BV-Aktionen definiert werden. Dazu müssen alle notwendigen Informationen und Schnittstellen in dem Prozess hinterlegt und gesammelt werden.
- Die Arbeitsabläufe des Disponenten müssen so genau, wie möglich beschrieben werden. Interaktionen und Kommunikation mit anderen Abteilungen müssen genau definiert werden.
- Eine logische Abhängigkeit zwischen den einzelnen Schritten bis zu einer vollständigen Ermittlung der BV-Menge muss erstellt werden.

- Eine Abfolge aller notwendigen Schritte, bis eine konsistente Entscheidung durch den Disponenten getroffen werden kann, muss festgelegt werden.
- Eine genaue Dokumentation der Prozesse muss erfolgen. Dies soll in Form eines Prozessmodells und verschiedener Dokumente geschehen.
- Die Zusammenarbeit der einzelnen Abteilungen muss dokumentiert werden. Den Abteilungen müssen die Schnittstellen zu dem Prozess einer Bevorratungen aufgezeigt werden.

Die Anforderungen an das Softwaresystem werden durch die im nun folgenden Abschnitt genannten funktionalen Anforderungen konkretisiert.

4.5.2 Funktionale Anforderungen

Die funktionalen Anforderungen beziehen sich auf einfache, generelle Funktionen der Software, als auch auf sehr spezielle Funktionen, die aus der Arbeit der Disponenten abgeleitet wurden.

- Prognosen, auf denen Bevorratungen basieren, müssen durch Korrekturreihen verändert werden können. Dazu muss ein Disponent bestimmte Bedarfe aus der Vergangenheit korrigieren können, um somit Werte, die sich stark von den sonstigen Werten unterscheiden, kompensieren zu können. Die Werte aus den Korrekturreihen korrigieren die Vergangenheitswerte.
- Die Änderung von Daten, die einer BV-Aktion zugrunde liegen, ist nach verschiedenen Zeitpunkten nicht mehr zulässig. Die Speicherung der Daten garantiert, dass die Entscheidungen in den BV-Aktionen nachvollzogen werden können und nachträglich keine Änderungen der Daten geschieht.
- Zwischen sehr ähnlichen BV-Aktionen werden Beziehungen hergestellt. Damit wird der Disponent auf in der Vergangenheit liegende, ähnliche BV-Aktionen hingewiesen und kann sich anhand der Daten einerseits orientieren und andererseits Fehler frühzeitig erkennen und vermeiden.
- In der Software IPOS bestehende BV-Aktionen werden ständig überwacht. Alle Werte werden dazu aufgezeichnet und anhand von Grenzwerten wird bestimmt, wann ein Wert eine Grenze überschreitet. Wenn dies geschieht, wird ein Disponent darauf aufmerksam gemacht. Dadurch können Über- oder Unterbeständen frühzeitig erkannt werden.
- Eine Alarmfunktion zur rechtzeitigen Alarmierung des Disponenten soll integriert werden. Hierbei werden Abweichungen, die einen vorgeschriebenen Wert überschreiten, sofort gemeldet. Diese Funktion geht mit der Überwachung verschiedener Werte in den BV-Aktionen einher und ergänzt diese.

- Die Einbindung von einer Wirtschaftlichkeitsrechnung zur sofortigen, unverzögerten Berechnung von einem monetären Nutzen der BV-Aktion bzw. eines Break-even-Preises soll innerhalb der Software IPOS realisiert werden. Der Benutzer erhält dadurch eine schnelle Vorstellung davon, ob die BV-Aktion wirtschaftlich rentabel ist oder nicht.
- Durch die Einbindung einer Flächenberechnung kann innerhalb einer BV-Aktion schnell festgestellt werden, wie viel Fläche die Anzahl der bevorrateten Teile im Endeffekt belegen wird.
- Die Überwachung von bereits abgeschlossenen BV-Aktionen soll ebenfalls implementiert werden. Durch diese Funktion werden alle BV-Aktionen, die innerhalb des Softwaresystems IPOS abgeschlossen sind, überwacht und hierbei das prognostizierte Absatzverhalten dem tatsächlich eingetretenen Absatzverhalten gegenübergestellt werden.
- Eine Überführung aller notwendigen Daten aus dem System IPOS heraus an das Disposition und Beschaffung-System soll realisiert werden. Damit kann die BV-Menge sofort zur Disposition übergeben werden.
- Das System IPOS soll eine Datenbasis nutzen, die darauf ausgelegt ist, große Datenmengen für lange Zeit zu speichern. Es müssen alle relevanten Daten, die innerhalb einer BV-Aktion entstehen, gespeichert werden können.
- Das System IPOS soll ein umfassendes System werden, welches alle notwendigen Funktionen für den gesamten Prozess einer ZBV/EBV bereitstellt. Es ist nicht geplant, das System modular aufzubauen bzw. Funktionen durch andere Softwaresysteme ausführen zu lassen, außer es gibt keine andere Lösungsmöglichkeit.
- Das System IPOS soll alle proprietären Softwareprodukte, die im Vorfeld des Projektes IPOS bereits entstanden sind, ersetzen und dazu die Funktionalitäten der Produkte ebenfalls anbieten.

Die funktionalen Anforderungen werden im folgenden Abschnitt durch die informellen Anforderungen noch weiter ergänzt.

4.5.3 Informelle Anforderungen

Die informellen Anforderungen beschreiben vorrangig die Notwendigkeiten zur Integration von weiteren Daten in das System IPOS.

- Um das Absatzverhalten der Teile in den kommenden Jahren besser prognostizieren zu können, sollen neben den bestehenden Daten noch weitere Daten in das System IPOS integriert werden.

- Durch die Integration von Garantie- und Kulanzfällen (kurz: GuK) in das bisherige Absatzverhalten soll gezeigt werden, welcher Anteil des Absatzes tatsächlich abgesetzt worden ist und welcher Anteil des Absatzes ‚nur‘ an die Werkstätten zur Abwicklung von GuK-Fällen zu zählen ist. Der Disponent kann dadurch besser unterscheiden, wie viel Bedarf in der Vergangenheit tatsächlich bestanden hat.
- Das Hinzufügen von genauen Informationen zu den zu bevorratenden Teile soll dem Disponenten genauer Auskunft über die Ladungsträger, den Lagerort und der durch eine BV-Aktion belegten Fläche geben.
- Da die Reparatur von manchen Teile sehr teuer sein kann, sollen die Disponenten durch die entsprechenden Kostensätze für die Reparatur eines Teiles und den Austausch eines Teiles unterstützt werden. Somit soll ein Disponent besser entscheiden können, ob die Lagerung eines Teils mit einem sehr hohen Kostensatz bei der Reparatur für 15 Jahre tatsächlich sinnvoll ist.
- Das Hinzufügen von Informationen zu dem vergangenen Absatzverhalten von sehr ähnlichen Teilen soll einem Disponenten eine bessere Orientierung ermöglichen. Er kann somit das Verhalten besser einschätzen.
- Eine umfangreiche Darstellung und Überwachung von großen BV-Aktionen, die einen festgesetzten Wert überschreiten, soll es der DCAG ermöglichen, schneller zu reagieren.
- Die Weitergabe der im System IPOS erstellten Absatzprognosen in weitere Datenquellen innerhalb der DCAG soll die Nutzung der Daten in weiteren Bereichen ermöglichen.

Die Anforderungen nach den informellen Aspekten des Systems IPOS werden nun durch die Anforderungen nach der Benutzerfreundlichkeit ergänzt.

4.5.4 Anforderungen an die Benutzerfreundlichkeit

Folgende Anforderungen wurden innerhalb des Projektes IPOS an die Software IPOS definiert:

- Bestimmte Aktionen müssen automatisiert werden, wie z.B. die automatische Übergabe von Daten in Sekundärsysteme.
- Eine Liste von BV-Aktionen, die bestimmte Werte überschritten haben, soll dem Disponenten zum Systemstart bereit stehen, damit er Probleme rechtzeitig erkennen kann.
- Das System muss dem Benutzer angepasste Masken zur Darstellung der Daten für BV-Aktionen zur Verfügung stellen. Je nachdem, welcher Benutzer eine BV-Aktion betrachtet, dürfen nur spezifische Daten zur Ansicht erscheinen.

- Umfangreiche Hilfsfunktionen sollen den Disponenten zu jedem Zeitpunkt unterstützen.
- Kleinere Überwachungsfunktionen prüfen ständig, ob vorher festgelegte Werte eingehalten werden. Ist zum Beispiel die zulässige Gesamtdauer für eine BV-Aktion bereits überschritten?
- Eine Aufgaben- und eine Prioritätenliste soll den Disponenten in der täglichen Arbeit mit dem System unterstützen.

Aus den oben genannten Anforderungen ist leicht eine große Anforderung nach einem umfassenden Softwaresystem herauszulesen. Trotzdem bleibt die Frage offen, wie dieses Softwaresystem umgesetzt werden kann und welche Möglichkeiten sich dazu bieten oder ob es noch Alternativen zum Einsatz eines Softwaresystems gibt. Mit diesen Aspekten setzt sich das folgende Kapitel auseinander.

4.6 Abstraktion der Anforderungen

Im vorherigen Abschnitt wurde eine Auflistung von Anforderungen vorgenommen, die in dem Abschnitt lediglich durch den Inhalt unterschieden wurden. Da die Übernahme aller Anforderungen in ein Lastenheft für die Konzeptionierung eines WFMS aus Sicht des Autors nicht sinnvoll erscheint, soll an dieser Stelle ein Großteil der Anforderungen zusammen gefasst und, sofern möglich, abstrahiert werden. Damit lassen sich leichter Anforderungen und Aussagen zur Funktionalität des Gesamtsystems treffen, als es durch die vielen, kleinen Anforderungen geschehen kann.

Die wichtigsten, abstrakten Anforderungen an das System IPOS sind:

- Das System soll die Benutzer vom Beginn eines Prozesses einer Bevorratung bis zum Ende des Prozesses leiten. Der Benutzer kann alle notwendigen Tätigkeiten für die Abwicklung einer BV-Aktion innerhalb eines Systems durchführen. Dabei ist die Abfolge der notwendigen Aktivitäten vorgeschrieben.
- Das System unterstützt den Benutzer bei der Abwicklung von Tätigkeiten und automatisiert einen Teil der Tätigkeiten selbstständig.
- Der System bindet auf Anforderung hin notwendige externe Anwendungen in die Arbeit mit ein.
- Das System IPOS soll die Funktionen der vorangegangenen Softwaresysteme übernehmen.

Diese vier grundsätzlichen Anforderung an das System IPOS lassen den Grundcharakter der Software erkennen: es handelt sich um ein WFMS.

Gleichwohl im kommenden Kapitel zuerst noch eine Betrachtung der möglichen Lösungsansätze diskutiert wird, wird in dem späteren Kapitel 6 die Entwicklung eines konzeptionellen WFMS durchgeführt. Hierbei bilden diese vier grundsätzlichen Anforderungen den Ausschlag gebenden Kern der Entwicklung. An diesen Anforderungen wird sich bei der Lösung und Entwicklung orientiert.

4.7 Zusammenfassung

Das Kapitel hat die Probleme des Projektes IPOS aufgezeigt und dem Leser die Wichtigkeit einer strategischen und koordinierten Bevorratung aufgezeigt. Die entscheidende Größe ist die vom Disponenten bestimmte Menge, welche die weiteren Faktoren, wie das gebundene Kapital und die gebundene Fläche beeinflusst. Neben diesen Einflussgrößen existieren noch eine Menge von weiteren Problemen, die hauptsächlich die Arbeit der Disponenten betrifft.

Das Projekt IPOS hat sich demnach als Ziel gesetzt, durch ein Softwaresystem die Arbeit der Disponenten zu vereinheitlichen und somit die Probleme zu verringern.

Da es neben dem Einsatz von einem individuell entwickelten Softwaresystemen auch weitere Lösungsmöglichkeiten gibt, wird in dem kommenden Abschnitt dies diskutiert.

5 Betrachtung von Lösungsmöglichkeiten

Durch den Projektantrag [Mau04] ist das Ziel des Projektes IPOS nur global und sehr grob definiert worden. Hier ist noch keine Rede von einem Softwaresystem zur Unterstützung der Disponenten. Erst in verschiedenen Steuerkreissitzungen, wobei ein Steuerkreis ein Gremium zur Steuerung von Projekten darstellt, wurde die Schaffung eines neuen Softwaresystems beschlossen (siehe [Mau05]). Hierbei ist noch keine Sprache davon, wie das System aussehen soll oder auf welcher Basis es arbeiten soll. Ausgehend von den im vorherigen Kapitel dargestellten Probleme und Anforderungen, soll in diesem Kapitel eine Betrachtung zu den Lösungsmöglichkeiten durchgeführt werden.

5.1 Lösung ohne Softwaresystem

Die Fragestellung, ob sich die Probleme der Bevorratungsaktionen auch ohne Unterstützung eines Softwareproduktes lösen lassen, ist nicht uninteressant, schließlich sollte bedacht werden, dass die Schaffung eines komplett neuen Systems eine große Investition ist.

5.1.1.1 Grundcharakter

Eine Alternative zu einem neuen Softwareprodukt ist darin zu sehen, die Disponenten durch geeignete Maßnahmen, die keine neue Software erfordern zu unterstützen. Solche Maßnahmen sind zum Beispiel Arbeitsanweisungen, Prozessbeschreibungen und Schulungen zum besseren Verständnis der bisher schon verfügbaren Systeme. Durch detaillierte Prozessbeschreibungen und gegebenenfalls Soll-Prozessbeschreibungen kann den Disponenten vor Augen geführt werden, wie solche BV-Aktionen effizienter durchzuführen sind. Umfassende Arbeitsanweisungen, die alle notwendigen Hinweise und alle notwendigen Schritte für BV-Aktionen enthalten, können die Soll-Prozesse noch untermauern und den Disponenten genaue Beschreibungen zu den BV-Aktionen liefern. Die dritte Maßnahme der Schulungen, integriert alle vorherigen Maßnahmen und liefert dem Disponenten damit eine breite Wissensbasis für zukünftige Bevorratungsaktionen.

5.1.1.2 Vorteile

Der Vorteil dieser Lösung ist einfach zu erkennen. Die Schaffung eines zusätzlichen Softwaresystems wird damit nicht notwendig und die Kosten, welche eine Neuschaffung hervorrufen würde, wären somit vermeidbar. Die Projektmitglieder können sich damit auf die Stärkung der theoretischen Hintergründe beschränken und die Erstellung der

Arbeitsanweisungen und Beschreibungen zum Vorgehen fokussieren. Eine genaue Auflistung und Dokumentation der gesamten Prozesslandschaft wäre in diesem Szenario vorteilhaft. Durch die Aufnahme der Prozesse können eventuell redundante Vorgänge erkannt und beseitigt werden. Die Arbeit der Disponenten wird durch diese Maßnahmen vereinheitlicht und Entscheidungen dadurch schneller nachvollziehbar. Fehler könnten vermieden werden.

5.1.1.3 Nachteile

In dieser Lösung müssen viele personelle Ressourcen für die Erstellung von detaillierten Prozessbeschreibungen und umfangreichen Arbeitsanweisungen investiert werden. Die damit nötige intensive Zusammenarbeit zwischen Projektmitgliedern und Disponenten bindet Arbeitskräfte. Weiterhin sind Schulungen natürlich ebenfalls kostspielig – deren Kostenfaktor darf nicht unterschätzt werden.

Darüber hinaus hat sich in der Vergangenheit gezeigt, dass manche bereits durchgeführte Schulungen, grade auf dem Gebiet des bisherigen Prognosetools, nur sehr wenig Anklang finden. Nur sehr wenig Disponenten wenden darüber hinaus die Fülle von kleinen Softwaretools zur Unterstützung wirklich an. Viele verlassen sich eher auf ihre Erfahrungswerte und ignorieren die zur Verfügung stehende Software.

Das Szenario ohne ein neues Softwaresystem ist zwar denkbar, wäre aber nach Einschätzungen der Projektmitglieder wenig Erfolg versprechend. Das bisherige Verhalten der Disponenten würde sich nicht soweit ändern, dass die Maßnahmen wirklich zu einer Verbesserung der Situation führen würden.

Aus diesem Grunde ist entschieden worden, ein umfassendes Softwaresystem zu schaffen, welches zur Bearbeitung von BV-Aktionen genutzt werden muss und zwingend vorgeschrieben ist. Die Frage, welche Möglichkeiten sich zur Schaffung eines solchen Systems ergeben, wird in den folgenden Unterabschnitten geklärt.

5.2 Einsatz eines kommerziellen Softwaresystems

Gleichwohl der Einsatz eines kommerziellen, auf dem Markt verfügbaren Produktes zur Unterstützung der Anwender im Projekt IPOS nie vorgesehen wurde, soll an dieser Stelle trotzdem eine Betrachtung der Vor- und Nachteile des Einsatzes erfolgen.

5.2.1.1 Grundcharakter

Der Einsatz von kommerziellen Systemen zur Unterstützung von Workflows in den Firmen ist durch den Kauf eines kommerziellen Produktes gekennzeichnet. Nach dem Kauf kann sich

ein langwieriger Prozess des Anpassens der Software anschließen, auch Customizing genannt. Nachdem dieser Prozess abgeschlossen ist, werden sukzessive die verschiedenen theoretische ausgearbeiteten Workflows in das Softwaresystem eingegeben.

Hierbei ist man auf die vorgegebene Notation und die vorgegebenen Semantik des WFMS angewiesen und muss versuchen alle Aspekte in den Systemen abzubilden, die vorher als wichtig eingestuft wurden. Können Aspekte eventuell nicht abgebildet werden, muss entweder auf diese Aspekte verzichtet werden oder aber ein Prozess der Anpassung der Software gestartet werden. Eigene Erfahrungen des Autors zeigen, dass solche Prozesse sehr langwierig und kostspielig werden können.

Im Projekt IPOS ist auf die Möglichkeit des Einsatzes von Standardsoftware nie eingegangen wurden. Der Grund hierfür ist darin zu suchen, dass innerhalb des GLC's schon immer die Auffassung vertreten wurde, einzelne kleiner Softwarelösungen für den internen Gebrauch selber zu entwickeln. So sind das Prognosetool, als auch die WiRe, ebenfalls in Eigenregie entwickelt wurden – meist im Rahmen einer Diplomarbeit.

5.2.1.2 Vorteile

Die Vorteile bei der Nutzung von Standardsoftware sind vor allem in den niedrigeren Kosten bei der Anschaffung eines Produktes zu sehen, als auch bei einer schnelleren Einsatzfähigkeit des Produktes und somit einer schnelleren Erfüllung seiner Aufgaben im Unternehmen. Ob der Einsatz im Projekt IPOS sinnvoll gewesen wäre und zu niedrigeren Kosten geführt hätte, lässt sich nicht nachvollziehen. Bisher wurde keine Zusammenfassung der Kosten im Projekt durchgeführt und ein monetärer Wert lässt sich nur schwer beziffern.

5.2.1.3 Nachteile

Anpassungen an Standardsoftware können, aus der Erfahrung des Autors heraus, sehr langwierig sein. Gleichwohl sie nicht immer notwendig sind, ist die Anwendung von Standardsoftware auf einen sehr speziellen Themenkomplex nur dann sinnvoll, wenn die Grundfunktionalitäten der Software genutzt werden können und durch eigene Module die speziellen Funktionen implementiert werden können. Hierzu sind aber spezielle Kenntnisse über die Schnittstellen der Standardsoftware von Voraussetzung, die im Projektteam nicht gegeben waren. Der zeitliche Aufwand für die Aneignung der Kenntnisse auf dem Gebiet der Software wäre somit dem Aufwand zur Aneignung der Kenntnisse für das .NET-Framework gleichzusetzen.

5.3 Einsatz einer Individualsoftware

Der Einsatz einer Software, welche als Eigenprojekt und als völlige Neuentwicklung abgewickelt werden soll, ist die bevorzugte Lösung der Projektmitglieder für die Probleme der BV-Aktionen. Im Steuerkreis in der Mitte des Jahres 2005 wurde die Festlegung durch das Gremium getroffen, die Software in der Eigenverantwortung zu erstellen, anstatt durch einen externen Dienstleister [Mau05].

5.3.1.1 Grundcharakter

Aus der Festlegung heraus, ein neues System zur Unterstützung bei BV-Aktionen zu schaffen, ist das Ziel eines webbasierten Systems formuliert worden [Mau05].

Die Programmierung des gesamten Softwaresystems erfolgt durch Eigenleistung innerhalb des Projektteams, in dem jeder Mitarbeiter bestimmte Module (siehe Projektstruktur) zu programmieren hat. Die Einhaltung von globalen Vorgaben der IT-Abteilung bei der Programmierung ist Aufgabe eines jeden Projektmitgliedes. Die Umsetzung der Anforderungen erfolgt jeweils punktweise, wobei die Modulstruktur beachtet wird und die Anforderungen entsprechend vorher zugeordnet werden. Da in jedem Steuerkreis immer wieder neue Anforderungen hinzukommen, ist auch nach der Fertigstellung eines ersten Prototypen davon auszugehen, dass ständig neue Anforderungen an das System entstehen und diese einzupflegen sind. Das System wird demnach mehrere Versionen durchlaufen, bevor es in einen abschließenden Produktivbetrieb wechselt.

5.3.1.2 Vorteile

Die Entwicklung von einer Individualsoftware garantiert die höchst mögliche Anpassung und Unterstützung der Disponenten in ihrer Arbeit. Durch die Sammlung der nötigen Informationen im Vorfeld der Entwicklung können die grundsätzlichen und die erweiterten Funktionalitäten genau bestimmt werden.

Jegliche Anpassungen an verschiedene Problematiken können ohne große Verzögerung durchgeführt werden. Weiterhin sind alle nötigen Aspekte in der Software vorhanden, alle Funktionalitäten sind eingearbeitet, alle notwendigen Datenquellen werden an das System per Schnittstelle angeschlossen. Durch die Kenntnis über den internen Aufbau des gesamten Systems können neue Module ohne große Verzögerung neu hinzugefügt werden. Änderungen, die durch die Disponenten oder den Steuerkreis vorgegeben werden, können ohne große zeitliche Verzögerung realisiert werden.

Die Entwicklung ohne Lastenheft, ohne Pflichtenheft, sprich ohne vertragliche Grundlage, die bei einer Entwicklung mit einem externen Partner nötig wären, garantiert eine zügige

Umsetzung in das System. Da die Projektmitglieder im Projekt IPOS ständig örtlich zusammen sind, sind kurze Kommunikationswege gegeben.

5.3.1.3 Nachteile

Es ergeben sich durch die Entscheidung für die Entwicklung von einem individuellen Softwareprodukt Nachteile. Hierunter fällt beispielsweise, dass im Projekt IPOS ein Mangel an Programmierern herrscht und die Programmierung von neuen Modulen viel Zeit in Anspruch nimmt. Daneben hat jeder Mitarbeiter im Projekt IPOS noch weitere Aufgabe aus dem Tagesgeschäft. Diese nehmen oft einen Großteil der Arbeitszeit in Anspruch, sodass im Endeffekt die Kapazitäten für die Entwicklung der Software fehlen.

Die interne IT-Abteilung des GLC's hat für das Projekt IPOS die Vorgabe gemacht, das .NET-Framework von Microsoft zu nutzen, wodurch sich ein weiterer Nachteil in dieser Lösung ergibt: das fehlende Wissen bei den Projektmitgliedern zu dem .NET-Framework. Die Projektmitglieder haben wenig Erfahrung im Umgang mit dem .NET-Framework und kennen die technischen Möglichkeiten wenig. Ebenfalls problematisch ist die .NET-Framework-Vorgabe, da selbst die IT-Abteilung innerhalb des GLC's wenig Erfahrungen mit dem Framework aufzuweisen hat. Zu speziellen Techniken des Frameworks existieren keinerlei Erfahrungen. Dadurch müssen oft Nachforschungen aufgenommen werden, um Lösungen und Erklärungen, als auch Programmbeispiele zu finden. Diese Aktivitäten kosten enorm viel Zeit.

5.3.1.4 Erkennung der Softwarephasen im Projekt

Die Phasen einer Softwareentwicklung wurden in der Arbeit bereits theoretisch abgehandelt. Ausgehend von dieser Theorie, soll in diesem Abschnitt nun zu jeder Phase der Softwareentwicklung der praktische Bezug im Projekt IPOS hergestellt werden. Dazu wird iterativ jede Phase betrachtet:

- *Problemdefinition:* Durch eine hohe Unter-/Überdeckung einer Teilenummer nach einer End- bzw. Zwischenbevorratung entsteht eine hohe Kapitalbindung in Form von gelagerten Material und auch Flächenbindung. Diese Kapitalbindung muss so gering wie möglich gehalten werden.
- *Anforderungsanalyse:* Die Disponenten erarbeiten in Zusammenarbeit mit dem Entwicklungsteam im Projekt IPOS die Anforderungen an das zukünftige System. Hierbei werden kleine Details der Benutzerführung geklärt, als auch große grundlegende Strukturen erläutert. Innerhalb des Entwicklerteams werden weitere Anforderungen geklärt und die Machbarkeit aller Anforderungen diskutiert und

evaluiert. Diese Evaluierung erfordert teilweise eine sehr genaue Betrachtung der zugrunde liegenden Technologie.

- *Spezifikation:* Innerhalb des Projektes IPOS vollzieht sich die Entwicklung ohne eine sonst übliche vertragliche Basis. Da die gesamte strategische Richtung der Entwicklung dem Entwicklungsteam klar ist und sehr oft neue Anforderungen an das kommende System entstehen, wäre der Aufwand zur Aktualisierung der notwendigen Dokumente zu hoch, als dass man dies in dem kleinen Team realisieren könnte. Da sich darüber hinaus auch Auftragnehmer und –geber in diesem Fall gleichen, entfällt in dem Projekt eine vertragliche Grundlage.
- *Entwurf:* Der Entwurf des zukünftigen Systems IPOS ist das zentrale Arbeitsobjekt innerhalb der Diplomarbeit. Hierbei wird durch die Analyse der Anforderungen und das Abstimmen mit den Beteiligten ein erster Entwurf erstellt. Dabei wird unter der Verwendung von UML-nahen Strukturen ein erstes strukturelles Abbild erstellt.
- *Implementation:* In ersten Schritten erfolgte eine Implementierung einzelner kleiner Teilkomponenten. Da der Fokus der Arbeit nicht auf der Implementierung lag, wurde dieser Punkt nur in einigen Ausnahmen weiter verfolgt.
- *Erprobung:* Die Erprobung kann im Projekt IPOS mangels eines noch nicht implementierten Systems noch nicht durchgeführt werden und wird daher hier vernachlässigt. Partielle Tests wurden im Rahmen fertig gestellter kleinerer Komponenten mit durchgeführt.
- *Auslieferung:* Auch diese Phase kann mangels eines noch nicht komplettierten Systems nicht durchgeführt werden und kann daher an dieser Stelle vernachlässigt werden.

Das voraussichtliche Ende der gesamten Entwicklung des zukünftigen Systems IPOS soll noch innerhalb des Jahres 2006 liegen und mit der Auslieferung des Systems und der Schulung der Disponenten enden.

5.4 Kapitelabschluss

Das Kapitel hat die möglichen Lösungsmöglichkeiten für die Probleme im Projekt IPOS aufgezeigt und Vor- und Nachteile genannt. Gleichwohl die Entscheidung über die Entwicklung eines individuellen Softwareproduktes nie in Frage gestellt wurde, zeigt der Abschnitt, dass es neben dieser Lösungsmöglichkeit noch andere Wege geben würde, die zu einer Verbesserung der Gesamtsituation im Projekt IPOS führen könnten.

Die Lösung mittels einer Individualsoftware wird im weiteren Verlauf der Arbeit weiterverfolgt. Im folgenden Kapitel wird dazu eine Architektur für das zukünftige System IPOS entwickelt.

6 Entwicklung einer konzeptionellen Applikationsschicht

Bevor mit der Entwicklung begonnen wird, sollen in diesem Kapitel die Referenzarchitektur der WfMC und der Einsatz von Patterns erklärt werden. Danach wird ein Soll-Prozess erstellt und entsprechend den Anforderungen der Umsetzung in ein Softwaresystem überführt. Danach beginnt die Entwicklung des Workflow Enactment Services und der zusätzlichen Komponenten für das WFMS. Danach werden diese Komponenten in Klassen überführt. Eine letzte Betrachtung der fünf typischen Anwendungsfälle der WfMC beenden das Kapitel.

6.1 Referenzen der Workflow Management Coalition

Die WfMC existiert bereits seit 1993 und hat als eine lose Organisation von verschiedenen Unternehmen, unterschiedliche Standards für Workflowanwendungen herausgegeben. Sie definierte den Begriff Workflow und schaffte eine Referenz für die zukünftigen Implementierungen von Workflowmanagementsystemen. In diesem Abschnitt wird ein grundlegendes Verständnis zur Theorie der WfMC durch die Erläuterung der Referenzarchitektur geschaffen. Der Abschnitt schließt mit einer Erklärung zu dem Workflow Management Application Programming Interface.

6.1.1 Referenzarchitektur

Innerhalb dieses Abschnitts soll die Referenzarchitektur der WfMC für WFMS erklärt werden. Folgende Abbildung zeigt die von der WfMC herausgegebene Referenzarchitektur für WFMS als schematische Darstellung.

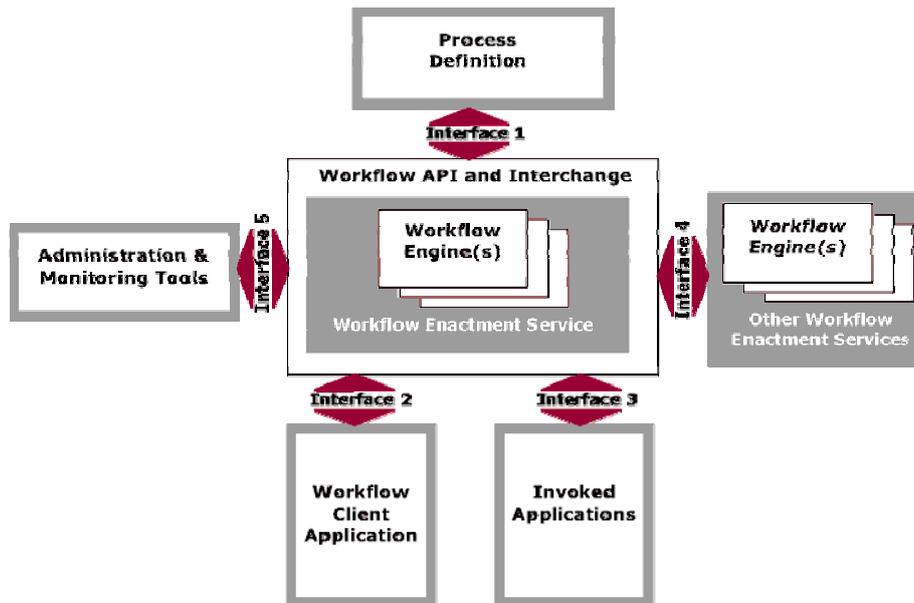


Abbildung 6-1: Schematische Darstellung der Referenzarchitektur [WfMC97]

Als zentraler Bestandteil zeigt sich hier der Workflow Enactment Service. Er umfasst die verschiedenen Workflow Engines, welche dafür zuständig sind die entsprechenden Workflowinstanzen zu verwalten, zu steuern und auszuführen.

Jegliche Client-Anwendungen greifen über ein Workflow Application Programming Interface, kurz WAPI, auf diesen Service zu. Im folgenden Abschnitt (6.1.2) wird dieses Interface noch genau erklärt.

Die WfMC definiert für Client-Anwendungen nicht etwa explizit fünf Interfaces, die dann jeglichen Anfrage gerecht werden können, sondern beruft sich auf fünf verschiedene Anwendungsszenarien. Diese Szenarien sollen hier nur kurz erwähnt werden:

- *Process definition:* Werden weitere Prozesse in ein WFMS eingepflegt, so geschieht dies durch dieses Interface. Eine Änderung bestehender Prozesse und Prozesselemente ist damit ebenso möglich.
- *Workflow Client Application:* Hiermit werden alle Fälle abgedeckt, bei denen eine Interaktion mit dem menschlichen Benutzer über eine Client Applikation zustande kommt.
- *Invoked Applications:* Ein Interface, welches das Aufrufen einer Applikation erlaubt, ohne genau zu wissen, welche Daten dafür nötig sind – das Interface stellt alle erforderlichen Informationen darüber zur Verfügung.
- *Workflow Engines:* Als eines der großen Ziele der WfMC bei der Entwicklung der Referenzarchitektur galt die Herausforderung die Interoperabilität zwischen verschiedenen WFMS zu steigern. Dafür dient dieses Interface, welches ein Zusammenspiel zwischen verschiedenen Workflow Enactment Services ermöglicht.
- *Administration & Monitoring Tools:* Das Interface sieht das Zusammenspiel des WFMS mit verschiedenen externen Tools vor, welche die Informationen aus dem

Workflowmanagementsystem aufbereiten können. Hierfür sieht die WfMC verschiedene Protokolle vor.

Wie leicht zu sehen ist, wird durch die WfMC keine konkrete Architektur vorgegeben, sondern vielmehr eine Referenz, nach derer man ein neu aufzubauendes WFMS konstruieren kann. Hierbei sind die fünf Interfaces weniger Schnittstellen im eigentlichen Sinne, sondern vielmehr typische Interaktionsfälle, die durch die Komponenten und Berücksichtigung in der Architektur abgedeckt werden. Dem Entwickler eines neuen WFMS steht es vollkommen frei, inwieweit diese Vorgaben der WfMC eingehalten/realisiert werden oder an welchen Stellen der Entwickler von der Referenz abweicht. Die WfMC lässt genügend Freiheiten.

6.1.1.1 Vorteile

Durch die Berücksichtigung der verschiedenen Anwendungsszenarien besitzt ein WFMS, welches nach den Vorgaben und Standards der WfMC entsteht, alle notwendigen Voraussetzungen zur Interaktion mit den Benutzern, anderen Applikationen und anderen WFMS. Dieser Aspekt wird Interoperabilität genannt und gilt als ein großes Ziel der WfMC, siehe [WfMC00]. In der Quelle wird für eine Standardisierung argumentiert, da auf diesem Wege die investierten Mittel effizient eingesetzt werden und Risiken bei der Entwicklung minimiert werden. Besonders im Falle des Einsatzes eines WFMS über die Unternehmensgrenzen hinaus, ist der Aspekt eines standardisierten WFMS höchst wichtig.

Der Autor sieht den Vorteil bei der Verwendung eines Referenzmodells in der Abdeckung aller notwendigen Anwendungsfälle. Schon in der Vorentwicklung eines späteren WFMS kann somit erheblich Zeit gespart werden. Durch die Nutzung der Standards kann die Interoperabilität mit weiteren standardisierten WFMS sichergestellt werden.

Zur Zeit ist das Referenzmodell der WfMC bereits über zehn Jahre alt. Die Interface-Spezifikationen werden zwar immer wieder weiterentwickelt, jedoch verändern sie sich nicht grundlegend. Somit ist ein WFMS, sofern einmal etabliert, zukunftssicher.

6.1.1.2 Nachteil

Das Referenzmodell der WfMC ist sehr umfassend. Die Spezifikationen der fünf Interfaces sind sehr umfassend gestaltet worden, da mit ihnen eine Vielzahl von Anwendungsfällen abgedeckt werden sollen. Es ist damit fast unmöglich in annehmbarer Zeit ein WfMC-konformes WFMS zu entwickeln. Sollen auf diesem Wege schnelle Ergebnisse produziert werden, so muss entweder auf die Realisierung eines oder mehrerer Interfaces verzichtet werden, oder aber die Entwicklungszeit verlängert werden.

6.1.2 Workflow Management Application Programming Interface

Sinn und Zweck des WAPI ist die einheitliche Programmierung von komplexen Applikationsschnittstellen mit vereinheitlichten Methodenaufrufen. Dazu hat die WfMC eine Spezifikation herausgebracht, welche die Application Programming Interfaces zukünftiger Workflowmanagementsysteme so vereinheitlicht, dass auch andere Workflow Engines auf ein und demselben Weg angesprochen werden und keine Funktionalitäten durch sich unterscheidende Methodenaufrufe verloren gehen. Die WAPI gehört zum Interface 2. Die aktuelle Version der WAPI-Spezifikation liegt als Stand 2.0 vom Juli 1997 vor.

Innerhalb der Spezifikation werden die Funktionen der Workflow Engines in acht unterschiedliche Klassen unterschieden [WfMC98]:

- *WAPI Connection Functions*: Die Funktionen zur Herstellung einer Verbindung von einer Applikation zu einer Workflow Engine wird über die erste Klasse von Methoden gesteuert. Sie koordinieren und steuern den Zugriff auf die Engine.
- *WAPI Process Control Functions*: Mittels der Process Control Functions kann der Status einer Prozesses bzw. mehrerer Prozesse verändert werden. Hierbei ist wichtig zu erwähnen, dass bestimmte Methoden durchaus mehrere Prozesse und damit mehrere Benutzer betreffen können und demnach vorsichtig behandelt werden sollten bzw. einer Restriktion anhand eines Benutzermanagements unterliegen sollten.
- *WAPI Activity Control Functions*: Die Funktionen der dritten Klasse werden zur Statusänderung einer bzw. mehrerer Activites verwendet. Auch hier können einige Methoden einen oder mehrere Benutzer betreffen, sodass diese Funktionen ebenfalls vorsichtig eingesetzt werden sollten bzw. einer Restriktion anhand eines Benutzermanagements unterliegen sollten.
- *WAPI Process Status Functions*: Alle Status-Funktionen dienen zur Abfrage der einzelnen Status aller Prozesse, die sich zu einem Zeitpunkt in der Engine eines WFMS befinden.
- *WAPI Activity Status Functions*: Alle Status-Funktionen in dieser Klasse dienen zur Abfrage der Status einzelner Aktivitäten. Diese Klasse ist äquivalent der vorherigen Klasse, nur auf Basis der Aktivitäten.
- *WAPI Worklist Functions*: Die Funktionen dieser Klasse sind für die Bereitstellung von Informationen zu einzelnen Items aus einer Worklist bzw. zu der gesamten Worklist zuständig.
- *WAPI Application Invocation Functions*: Diese Klasse von Funktionen dient zur Steuerung der Tool Agents, welche ihrerseits externe Geräte und Applikationen, die Informationen für das WFMS bereit stellen, steuern. Über das genormte Interface

zwischen WFMS und Tool Agents kann das WFMS die Geräte/Applikationen ansteuern.

- *WAPI Administration Functions*: Die letzte Funktionsklasse stellt eine Menge von Funktionen zur Verfügung, die zur Administration und zur Wartung des gesamten Funktionsumfangs des WFMS dienen.

All diese Klassen werden im weiteren Verlauf der WAPI-Spezifikation weiter unterteilt und in konkrete Prozedurbeschreibungen verfeinert. Zu jeder dieser Prozeduren gibt es spezifizierte Ein- und Ausgabeparameter und entsprechende Fehlerrückgabewerte. Da es wenig sinnvoll wäre, an dieser Stelle jede einzelne Prozedur zu erläutern, soll die Betrachtung an dieser Stelle enden. Zu einem späteren Zeitpunkt der Arbeit aber, wird wieder auf diesen Punkt zurückgekommen und anhand einiger verwendeter Methoden die partiell konforme Entwicklung des neuen WFMS „IPOS“ gezeigt.

Abschließend bleibt zu sagen, dass es sich bei der WAPI um ein abstraktes Interface handelt, welches alle Möglichkeiten und Freiheiten bietet, ein neu zu entwickelndes WFMS WfMC-konform zu gestalten.

6.2 Einsatz von Patterns

Da zur Konzepterstellung innerhalb dieser Arbeit mehrmals auf Patterns zurückgegriffen wird, soll der generelle Einsatz dieser Patterns in diesem Kapitel erklärt werden.

Die Grundlagen zum Einsatz von Patterns soll im ersten Abschnitt kurz erklärt werden, um darauf hin mit der Beschreibung der beiden eingesetzten Patterns fortzufahren und mit der Einordnung der Patterns in die Referenzarchitektur der WfMC den Abschnitt zu beenden.

6.2.1 Der Ansatz von Pattern

Christopher Alexander schuf in den 1970'er Jahren eine erste Patternlanguage zur Lösung von verschiedenen Architekturproblemen. Innerhalb dieser Patternsprache verwandte er verschieden viele Muster zur Beschreibung und Lösung eines spezifischen Problems aus dem Bereich des Städtebaus. Dabei waren die Patterns nahezu gleich aufgebaut. Auf eine Beschreibung eines Problems, folgte die Beschreibung der Einflussgrößen und abschließend eine Lösungsbeschreibung. Jedes dieser Patterns abstrahierte eine Problemstellung und eröffnete so dem Nutzer der Patternlanguage eine Lösung zu einem Problem. Patterns stellen sich demnach als „allgemein bewährte Lösungen für Standardprobleme, also Problem/Lösungs-Paare, die in ähnlicher Form immer wieder angewendet werden können.“ [Sch96] dar.

Hierbei lassen sich schon die Grundgedanken einer Patternlanguage erkennen. Es werden verschiedene, in der Grundstruktur und dem Kontext übereinstimmende, Probleme abstrahiert und aus den Erfahrungen heraus eine Lösung bereitgestellt, die einem unerfahrenen Anwender hilfreich sein kann. Hier ist wichtig festzuhalten, dass meist nur Experten Patterns erstellen. Sie bringen die nötige Erfahrung und die nötigen Kenntnisse für die Problemlösungen mit sich, da sie die Vielfältigkeit eines Problems kennen und das gleiche Problem in verschiedenen Varianten bereits gelöst haben.

Die Verwendung von strukturierten Informationen zur Problemanalyse und –lösung ist kein neuer Ansatz, der erst durch das Pattern entstand. Vielmehr wird dieser Ansatz durch eine Abstraktion der Probleme mit der gleichzeitigen Kopplung mit strukturierten Lösungsinformationen gekennzeichnet. Entscheidend ist an Patterns, dass sie der Erkennung von Problemen, der Skizzierung der Problematik und der Hilfe bei der Lösung des Problems dienen. Patterns stellen sich also als eine Art von Hilfestellung bei einer vorhandenen Problem Instanz dar.

6.2.1.1 Vorteile

Durch die Verwendung von Patterns kann zu einer Problem Instanz schnell eine entsprechende Lösung gefunden werden. Dieses Zeitersparnis, die durch die Verwendung von Patterns entsteht, trifft besonders in Softwareprojekten zu. Hier können zu vielen unterschiedlichen Problemen in den unterschiedlichen Phasen der Entwicklung Patterns gefunden werden, die mit Standardlösungen eine schnelle Lösung bereit stellen.

Weiterhin können Patterns als Ideengebung dienen. So kann auf Basis eines Patterns die Lösungsidee des Patterns für ein gegebenes Problem einfach umstrukturiert werden und mittels Erweiterungen zu einer sehr spezifischen Lösung transformiert werden.

6.2.1.2 Nachteile

Problembeschreibungen, Einflussgrößen und Lösungsbeschreibungen der Pattern sind sehr allgemein gehalten. In diesen Punkten besteht ein Nachteil in der Verwendung von Patterns, denn die Problembeschreibung muss nicht unbedingt mit dem betrachteten Problem vollständig übereinstimmen, genauso wie die Lösung nicht sofort für ein gegebenes Problem zutreffen muss und sofort anwendbar sein muss. Hier muss vor der Verwendung solcher Lösungen eine Anpassung stattfinden, welche die Lösung soweit verändert, dass sie auf das gegebene Problem zutrifft.

6.2.2 Zwei Patterns zur Erstellung eines Workflowmanagementsystems

Um eine Grundstruktur für ein kommendes WFMS zu erstellen, ist es notwendig sich einerseits an den bestehenden Vorgaben/Richtlinien für ein WFMS zu orientieren (siehe hierzu Abschnitt 2.3), als auch an den Anforderungen, die in diesem Kontext als höher priorisiert anzusehen sind. Die Vorgaben der WfMC zu einer Architektur eines WFMS sind abstrakt und auch nur sehr grob gegliedert. Sie sind daher als Richtlinie zu verstehen, nach denen ein WfMC-konformes WFMS zu entwickeln wäre. In der Design-Phase des WFMS kommen zwei Patterns zum Einsatz, deren Kontext im Folgenden beschrieben werden soll. Es handelt sich um das Pattern „A Pattern for Managing Distributed Workflows“ von [Seu00] und die Pattern Language „A Pattern Language for Workflow Systems“ von [Mes97]. Beide Patterns unterstützen die Design-Phase durch die Lösungsgebung zu verschiedenen Problematiken, wobei das erstere Pattern zur Entwicklung der grundlegenden Architektur des umgebenen Systems genutzt wird und die Language zur Entscheidung über die grundlegende Workflowarchitektur.

6.2.2.1 „A Pattern for Managing Distributed Workflows“

Im Pattern von Seung et al. beginnen die Ausführungen über die interne Architektur des zukünftigen WFMS durch die Entwicklung einer Gesamtarchitektur aus den Anforderungen an ein WFMS heraus. Sie verfeinern die Architektur in den folgenden Themen immer weiter bis zu einer Übersicht über alle Klassen, welche in der kommenden Abbildung 6-2 zu sehen ist (nach [Seu00]).

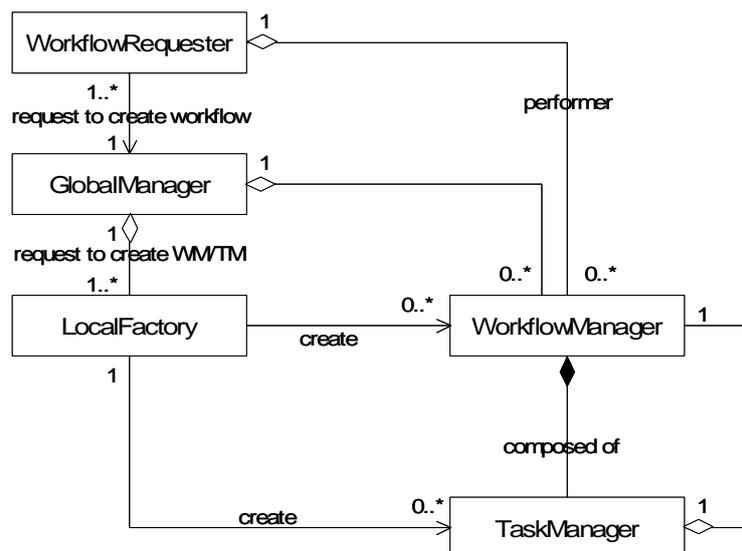


Abbildung 6-2: Klassendiagramm nach [Seu00]

Die obige Abbildung orientiert sich nach der Übersicht, die Seung et al. in ihrem Pattern aufgeführt haben. Nachfolgend sollen die wesentlichen Komponenten kurz erläutert werden [Seu00]:

- WorkflowRequester(WR): Der WR stellt die Anfrage zur Instanziierung eines Workflows. Darüber hinaus betreut er eine laufende Instanz eines Workflows.
- GlobalManager(GM): Der GM stellt den zentralen Anlaufpunkt zur Instanziierung eines Workflows dar; er steuert die LocalFactory. Des Weiteren beinhaltet er alle Informationen über alle im System verfügbaren WorkflowManager.
- LocalFactory(LF): Die LocalFactory instanziiert TaskManager oder WorkflowManager, sobald eine Aufforderung dazu vom GM kommt. Je Workflowserver sollte eine LocalFactory existieren.
- WorkflowManager(WM): Die WorkflowManager sind für die TaskManager verantwortlich, die wiederum für jeden Task innerhalb eines Workflows verantwortlich sind. Die WM hält alle notwendigen zu einem Workflow gehörigen Attribute bereit. Dazu zählen beispielsweise: Name, Ausführungsdatum usw. Task-spezifische Informationen werden hierin nicht bereitgehalten.
- TaskManager(TM): Sie sind für die Ausführung eines Tasks in einem Workflow Prozess zuständig und halten alle erforderlichen Daten für die Ausführung eines Tasks bereit.

Das Zusammenspiel der einzelnen Komponenten wird in [Seu00] durch die Erläuterung von verschiedenen Szenarien kurz dargestellt, soll allerdings an dieser Stelle nicht weiter von Interesse sein.

Wichtig ist an dieser Stelle zu erwähnen, dass sich durch die Architektur eine Hierarchie der einzelnen Komponenten ergibt, welche sich in der nachfolgenden Abbildung 6-3 nochmals verdeutlicht.

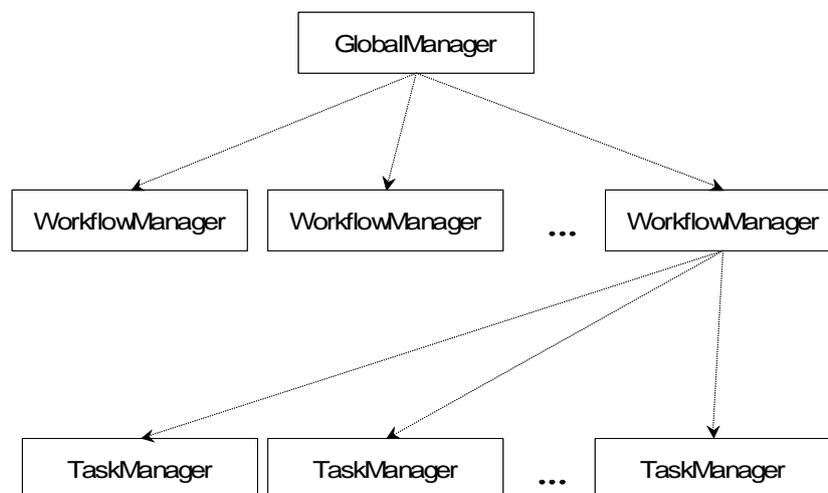


Abbildung 6-3: Komponentenhierarchie

Diese Komponentenhierarchie ergibt sich aus der Steuerung des gesamten WFMS über den GlobalManager hinweg. Dieser kontrolliert die WorkflowManager, welche dann die Kontrolle über die TaskManager besitzen. Somit ergibt sich hiermit auch ein Einblick in das grundlegende Verständnis eines Workflows, welches einen Workflow als eine Abfolge von Tasks darstellt. Diese Philosophie und die umgebende Komponentenhierarchie werden sich in dem Konzept des zu erstellenden WFMS „IPOS“ wieder finden.

6.2.2.2 „A Pattern Language for Workflow Systems“

Die Pattern Language von Meszaros und Brown nimmt nur einen kleinen Teil bei der Konzeption des WFMS „IPOS“ ein. Hierin lassen sich zu vielen unterschiedlichen Problemen bei der Erstellung zukünftiger Workflowmanagementsysteme Lösungsansätze finden. Eines dieser Probleme ist die Entscheidung über die interne Funktionsweise des WFMS. Meszaros und Brown machen zwei Vorschläge, wie die Workflowobjekte durch das Gesamtsystem geleitet werden kann. Dazu unterscheiden sie zwei Architekturen: die Push- und die Pull-Architektur.

Während der Design-Phase muss entschieden werden, wie das interne Routing der einzelnen Workflowobjekte geschehen soll. Dabei ist das Routing der Vorgang der Weitergabe eines Objektes von einem Punkt zum nächsten. Das Objekt muss entlang eines bestimmten Pfades weitergeleitet werden und darf keine der auf dem Pfad notierten Punkte auslassen/überspringen/ignorieren. An jedem Punkt des Pfades wird das Objekt verändert. Es muss daher sichergestellt werden, dass jedes Objekt durch eine geeignete Routing-Strategie jeweils zum richtigen Zeitpunkt an dem richtigen Punkt ist. Hierfür sehen die beiden Autoren zwei unterschiedliche Strategien anhand der beiden Architekturen Push und Pull vor, deren Inhalt im Folgenden erklärt werden soll.

Die Push-Architektur sieht zum Routing eines Objektes die Weitergabe des Workflowobjektes durch einen „single central workflowproduct router“ vor. Dieser ist für die Weitergabe eines Objektes von einem Punkt zum nächsten zuständig. Der Router organisiert dementsprechend die Weitergabe, ohne dass ein Ablaufwissen innerhalb der einzelnen Punkte vorhanden ist. Die unterschiedlichen Punkte besitzen ebenfalls kein Wissen über die anderen im System verfügbaren Punkte und kennen nur ihre Schnittstelle zu den Workflowobjekten. Der „workflowproduct router“ ist eine einzelne Komponente, die als zentrale Instanz die Kommunikation und das Routing der Objekte durch das System übernimmt.

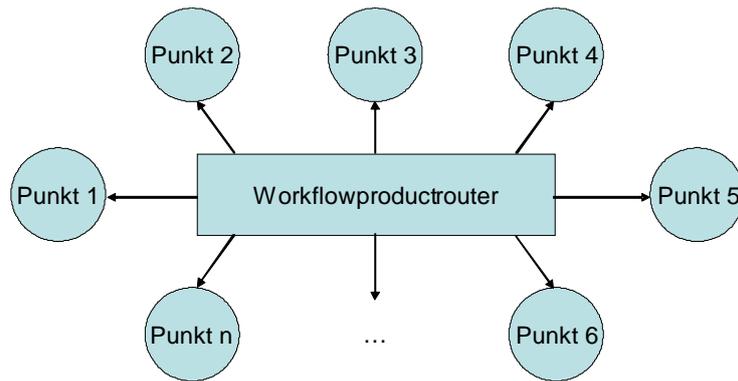


Abbildung 6-4: Single central workflowproduct router, nach [Mes97]

In der obigen Abbildung 6-4 kann das grundlegende Prinzip nochmals kurz erfasst werden. Einzig und alleine die zentrale Instanz des „workflowproduct router“ ist für die Kommunikation zuständig. Somit sind das gesamte Management des Ablaufs und das gesamte Wissen über die Vorgänge im System an einer Stelle zentralisiert.

Diese Architektur kann, wagt man einen Blick auf ein zukünftiges System, schnell zu einer Engstelle werden, so [Mes97]. Ausschlaggebend hierfür ist eine Überlastung des zentralen Routers durch zu viele Anfragen. Der Autor ist in diesem Falle der gleichen Meinung wie die beiden Autoren, würde aber hierfür unterschiedliche Lösungsansätze erkennen, welche jedoch hier nicht weiter aufgegriffen werden sollen.

Innerhalb der Pull-Architektur hingegen, wird eine strikte Dezentralisierung vorgenommen. Hierbei existiert lediglich ein Repository, welches die entsprechenden Informationen über eine Workflowobjekt bereithält. Sofern ein Punkt im Workflowmanagementsystem hinzu kommt bzw. wieder entfernt wird, werden die entsprechenden Informationen, wann die ein Workflowobjekt am diesem Punkt bearbeitet wird, hinzugefügt respektive wieder gelöscht. Damit organisieren sich die Punkte quasi selber, da sie jeweils zu einem bestimmten Zeitpunkt die Kenntnis darüber besitzen, welchen Status ein Objekt hat. Stimmt dieser Status mit den Vorgaben für diesen Punkt überein, wird das Objekt zu diesem Punkt „gezogen“ (daher auch die Namensgebung der Pull-Architektur).

Die Pull-Architektur ist somit natürlich sehr flexibel, da die Punkte das Wissen selber besitzen und somit ein Hinzufügen bzw. Entfernen eines Punktes aus dem Gesamtsystem sehr einfach ist. Allerdings läuft in einem auf der Pull-Architektur basierenden System ein Objekt die Gefahr an einem Punkt für ewig zu verweilen, wenn an keinem anderen Punkt des Pfades Interesse an dem Objekt gezeigt wird. Damit ist ein intensives Testen der implementierten Struktur immer eine Voraussetzung zur Implementierung der Pull-Architektur, so Meszaros und Brown.

Die beiden eben besprochenen Patterns reihen sich in eine Reihe von vielen unterschiedlichen, verfügbaren Patterns zum Thema Workflowmanagementsysteme ein. Der Autor hat sich allerdings als Ziel gesetzt, das zukünftige System IPOS angelehnt an die

Referenzarchitektur der WfMC zu konzipieren. Auf welchem Weg und wie weit sich die Patterns in diese Architektur einfügen, klärt der kommende Abschnitt.

6.2.3 Einordnung der beiden Patterns in die Referenzarchitektur der WFMC

Da die Referenzarchitektur der WfMC dem Leser bereits bekannt ist, soll an dieser Stelle die Einordnung der eben besprochenen Patterninhalte in das Verständnis der Referenzarchitektur der WfMC vorgenommen werden.

Die Patterns eröffnen verschiedene Möglichkeiten sie mit der Referenzarchitektur der WfMC zu koppeln bzw. die Referenzarchitektur weiter zu verfeinern. Da die WfMC genügend Freiheiten zur Umsetzung der Architektur lässt, können die Patterns in dieser Umgebung sehr gut eingearbeitet werden. Beginnend bei dem ersten Pattern von Seung et al. lässt sich hier festhalten, dass dieses Pattern sehr gut zum Design einer Workflowengine geeignet ist. Betrachtet man die einzelnen Bestandteile genauer, so offenbart sich der Teil einer internen Struktur einer Workflowengine. Dabei enthalten sind die grundlegenden Steuerungs-, Kontroll- und Instanziierungsaspekte für einen Workflow. Fehlende Bestandteile sind beispielsweise die Monitoring-Aspekte und Aussagen zu der Prozessdefinition und Arbeitsverteilung. Jedoch kann aus diesem Pattern ein erstes Extrakt gebildet werden, welches für eine zukünftige Workflowengine hervorragend geeignet ist.

Das zweite Pattern von Meszaros und Brown kann als eine Grundausrichtung des späteren WFMS gesehen werden. Hierbei sollte die generelle Funktionsweise des Gesamtsystems im Vordergrund stehen. Die Push- und Pull-Architekturen ergänzen dabei die funktionale Struktur der Workflowengine, die schon durch das vorherige Pattern von Seung et al. vorgegeben wurde. Sie beschreiben die eigentliche globale Steuerung und Funktionsweise innerhalb des gesamten Systems und ergänzen mit einigen Komponenten, wie z.B. das Repository, die bisherige Struktur.

Es bleibt zu sagen, dass die Patterns gut mit der Referenzarchitektur gekoppelt werden können und auch untereinander gut verbunden werden können. Sie ergänzen die Referenzarchitektur und geben einen ersten Vorschlag zur Realisierung.

Auf diesem Wege kann das erste Ziel eines WfMC-Referenz-nahen neuen Workflowmanagementsystem weiter verfeinert werden und ein erster Vorschlag zu einer internen Funktionsweise erarbeitet werden. Darüber werden in der Referenz der WfMC keinerlei Vorgaben gemacht, sodass es durchaus legitim erscheint, die Referenzarchitektur durch die Nutzung der Patterns zu detaillieren.

6.3 Erstellung und Überführung des Soll-Prozesses

Im vorherigen Abschnitt wurde bei der Pattern Language von Meszaros und Brown bereits von einem Pfad für die Workflowobjekte durch das WFMS gesprochen. Solch ein Pfad muss vor der Überführung in ein Softwaresystem zunächst abgebildet werden und so aufgebaut sein, wie der solotypische Ablauf im späteren WFMS sein soll. Dies geschieht in diesem Abschnitt durch Erstellung eines Soll-Prozesses für das Durchführen von BV-Aktionen.

6.3.1 Gründe, Ziele und Vorgehen zur Soll-Prozesserstellung

Im Zuge der Erstellung eines Soll-Prozesses für eine BV-Aktion hat sich der Autor mit zwei Disponenten auseinander gesetzt und mit deren Hilfe in Form von Interviews einen ersten Überblick über die Abläufe während eines Bevorratungsvorgangs verschafft. Es wird im Zuge dieser Interviews schnell klar, dass hier ein hoher Grad an unterschiedlichem Vorgehen bei den einzelnen Disponenten herrscht. Jeder einzelne Disponent geht bei solch einem Vorgang anders vor und agiert nicht immer gleich. Um diesen Vorgang für alle Disponenten zu harmonisieren wurde im Projekt IPOS die Erstellung eines Soll-Prozesses für eine ZBV/EBV vorangetrieben.

Das verfolgte Ziel ist eine Standardisierung der Vorgänge einer ZBV/EBV, um die Disponenten so zu leiten, dass fehlerhafte Bevorratungsaktionen nicht mehr durchgeführt werden. Durch den Soll-Prozess wird es darüber hinaus auch möglich sein, die Disponenten an den wichtigsten Schnittstellen zu unterstützen und soweit mit Daten zu versorgen, dass ihm zu vielen Themen mehr Informationen sofort bei Bedarf zur Verfügung stehen werden, als es zur Zeit möglich ist. Dem Disponenten wird demnach in IPOS eine Fülle von Daten zur Verfügung stehen, die er nutzen kann und daher seine Bevorratungsmenge besser abschätzen kann. Weiterhin kann durch einen genau definierten Prozess eine erste Vorstellung davon gewonnen werden, wie ein ‚Pfad‘ eines Workflowobjektes durch ein zukünftiges WFMS aussehen kann.

Zur Erstellung des Soll-Prozesses wird zuerst projektintern ein erster Entwurf eines Prozessmodells auf Basis einer eEPK erstellt. Hierbei sind vor allem die einzelnen Prozesse wichtig, die ein Disponent durchzuführen hat und die Abfolge dieser Prozesse. Danach werden die Prozesse in eine sachlogisch richtige Reihenfolge gebracht und so geordnet, dass ein Prozess auf den Ergebnissen des vorherigen Prozesses aufbaut. Dies ist wichtig für die spätere Implementierung und Reglementierung im WFMS. So wird sichergestellt, dass Disponenten alle Tätigkeiten erfüllen und kein Prozess vor einem anderen ausgeführt wird, obwohl dieser zeitlich nachgelagert ist.

Weiterhin wird im Soll-Prozess eine erste Analyse durchgeführt, welche Daten zwischen dem System IPOS und den anderen DC-internen Systemen ausgetauscht werden. Darüber

hinaus wird auch festgehalten, welche Daten überhaupt während des gesamten Prozesses einer ZBV/EBV entstehen werden und gespeichert werden müssen.

Eine weitere wichtige Fixierung ist die Abbildung der Teilnahme der unterschiedlichen DC-internen Abteilungen an dem Bevorratungsprozess. Hierbei wird durch die Soll-Prozess-Erstellung protokolliert, wann welche Abteilung auf welchem Weg und in welchem Umfang in die BV-Aktion eines Disponenten eingreift.

Im Folgenden wird der Soll-Prozess iterativ durchlaufen. Da der Prozess sehr groß ist, wird er immer segmentweise durchlaufen. Anhaltspunkte geben die Konnektoren. Zur Modellierung wurden die von IDS Scheers Produkt ARIS bekannten eEPK eingesetzt (siehe [Sch91]), hier allerdings in einer abgewandelten, eigenen Notation, internen DC-Vorgaben folgend (siehe [Glo00]).

6.3.2 Segment 1: Datenkollektion

Nachfolgende Abbildung 6-5 zeigt das erste Segment des Soll-Prozesses. Es handelt sich hier um die Datenkollektion bzw. den Abgleich der Daten mit den vorher festgelegten benötigten Daten für den Soll-Prozess.

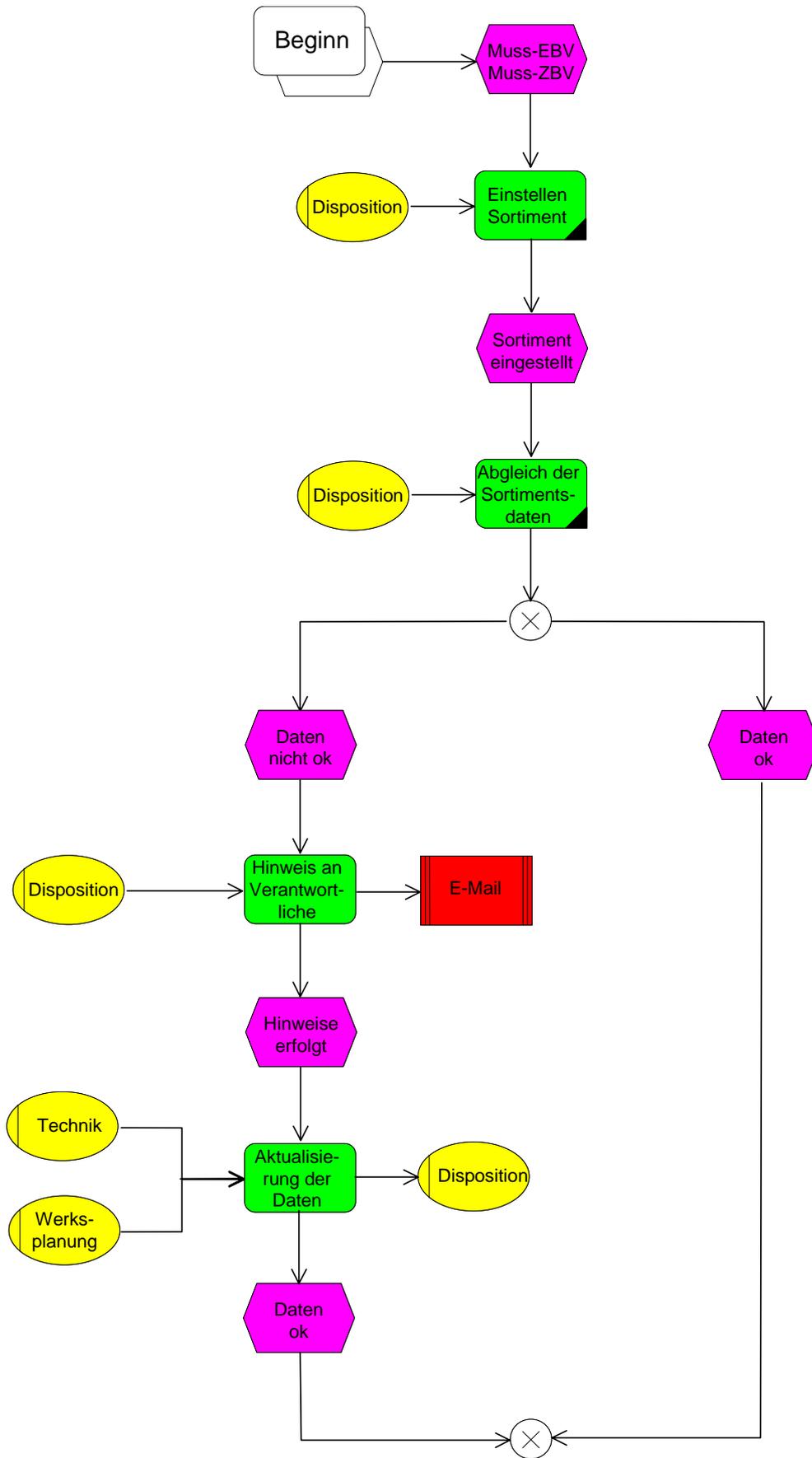


Abbildung 6-5: Segment 1: Datenkollektion

Bei der Betrachtung des Beginns des Soll-Prozesses startet der gesamte Prozess mit einer Schnittstelle und dem Ereignis „Muss-EBV Muss-ZBV“. Danach folgt eine Funktion zum Einstellen des Sortiments. Hierbei ist ein Sortiment eine Menge von zur Bevorratung vorgeschlagenen Teilenummern. An der Funktion ist ein kleines schwarzes Dreieck zu erkennen, welches eine Komposition/Verfeinerung andeutet.

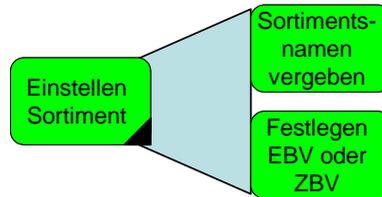


Abbildung 6-6: Einstellen des Sortiments

Um den Hauptablauf im Soll-Prozess frei von vielen kleinen Aktivitäten zu halten, hat sich der Autor entschieden auf diesem Wege eine Verfeinerung darzustellen. Die obige Abbildung 6-6 zeigt hierbei die elementaren Tätigkeiten, die ein Disponent durchzuführen hat.

Sofern die Tätigkeit des Einstellens eines Sortiments komplettiert ist, muss ein Disponent der Abbildung 6-5 weiterfolgend einen Abgleich der Daten durchführen. Dazu wird er die Tätigkeiten, wie sie in der kommenden Abbildung 6-7 zu sehen sind, durchführen müssen.

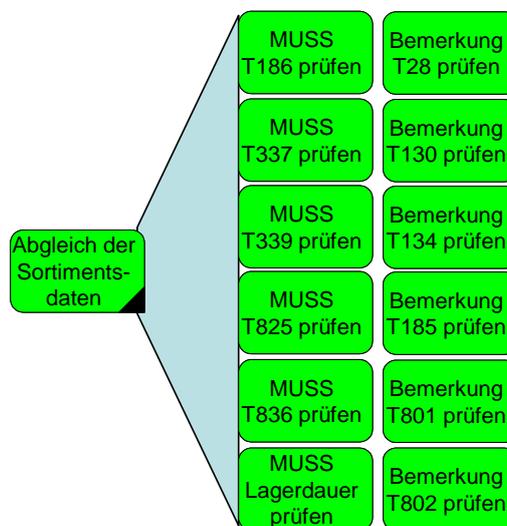


Abbildung 6-7: Abgleich Sortimentsdaten

Dazu gehört vor Allem die Überprüfung von Muss-Feldern in den Daten zu den einzelnen Teilenummern. Je Teilenummer existieren innerhalb der DCAG verschieden viele Parameter, die intern zur Codierung von vielen unterschiedlichen Informationen zu einer Teilenummer genutzt werden. Diese Informationen werden in Feldern, die mit ‚T‘ beginnend und einer Zahl endend, kodiert. Diese Felder hat der Disponent für jede einzelne Teilenummer zu überprüfen. Bemerkungsfelder sind mit zu überprüfen, müssen aber nicht gefüllt werden.

Wenn der Abgleich der Daten erfolgt ist, wird der Disponent entweder alle Daten komplettiert vorliegen haben, oder aber einige Daten werden fehlen. Je nachdem welche Daten fehlen, kann der Disponent entweder selber die Daten einpflegen oder muss per E-Mail an die Technik die Aktualisierung der Daten veranlassen.

Die beiden Pfade der eEPK werden durch einen XOR-Konnektor wieder vereint. Nach dem Konnektor beginnt das nächste Segment.

6.3.3 Segment 2: Prognoseerstellung – Schleifenbeginn

Die Prognoseerstellung stellt den Beginn einer Schleife dar, die in einem Prozess der Bevorratung immer wieder durchlaufen werden soll.

Durch die Prognose kann der Disponent eine erste Mengenschätzung für die zu bevorratenden Teilenummern vornehmen. Dies geschieht auf Basis eines langfristigen Prognosemodells. Dabei übergibt der Disponent aus dem System IPOS heraus die entsprechenden Daten an die externe Prognosekomponente. Die Komponente ermöglicht es, Prognosewerte ohne zeitliche Verzögerung für eine Teilenummer zu erhalten, die Prognose durch das Hinzufügen von Korrekturreihen zu verändern und die Auswirkungen auf eine Prognose unmittelbar zu verfolgen. Dem Disponenten wird hiermit ein sehr starkes Tool zur Verfügung gestellt, welches ihn mit einer hohen Genauigkeit gute Prognosewerte erstellen lässt.

Die kommende Abbildung 6-8 zeigt das zweite Segment des Soll-Prozesses:

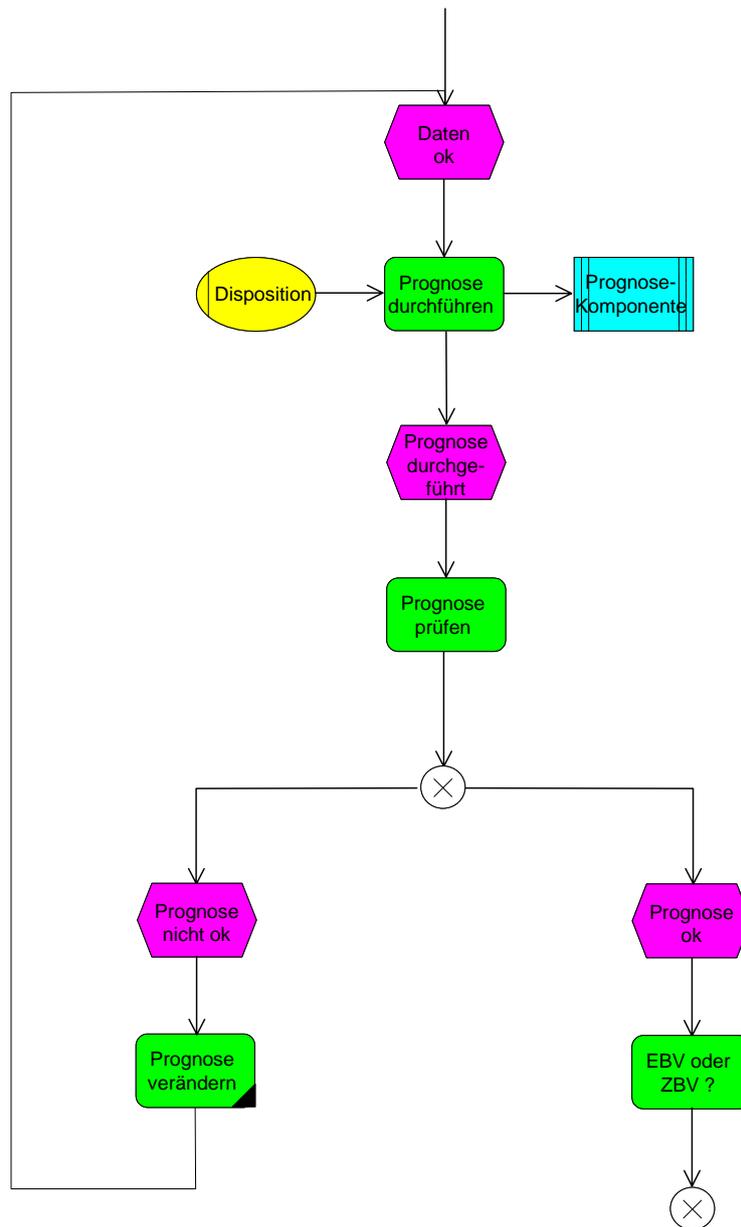


Abbildung 6-8: Segment 2: Prognoseerstellung

Nachdem eine Prognose erstellt wurde, prüft ein Disponent diese und entscheidet dann, ob die Prognose in Ordnung ist oder nicht. Wenn die erhaltenen Daten aus der Prognose nach Meinung des Disponenten nicht korrekt sind bzw. er mit dem Prognoseverlauf nicht zufrieden ist, muss er die Prognose verändern.

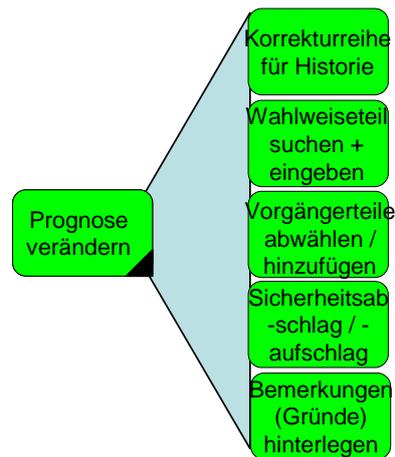


Abbildung 6-9: Tätigkeiten zur Veränderung der Prognose

Die Abbildung 6-9 zeigt auf einen Blick alle Tätigkeiten, die der Disponent vorzunehmen hat, wenn die Prognose nicht seinen Erwartungen entspricht. Dazu gehören beispielsweise das Hinzufügen oder Entfernen von Korrekturreihen, welche die Historie einer Teilenummer beeinflussen. Weiterhin können Vorgängerteile berücksichtigt werden oder ein globaler Sicherheitsaufschlag/-abschlag berücksichtigt werden. Alle diese Tätigkeiten verändern den Prognoseverlauf.

Nur wenn der Disponent mit der gesamten Prognose zufrieden ist, verfährt er in dem Prozess der EBV/ZBV weiter und geht weiter zum nächsten Segment.

6.3.4 Segment 3: EBV-Workflow und Einkauf

Nachdem der Disponent eine Prognose für jede zu bevorratende Teilenummer erstellt hat, muss er in IPOS alle Teilenummern über eine Schnittstelle in das System JDAX eingeben.

Dazu werden per vordefinierter Schnittstelle die Daten in das System übertragen.

Die Abbildung 6-10 zeigt diesen Schritt.

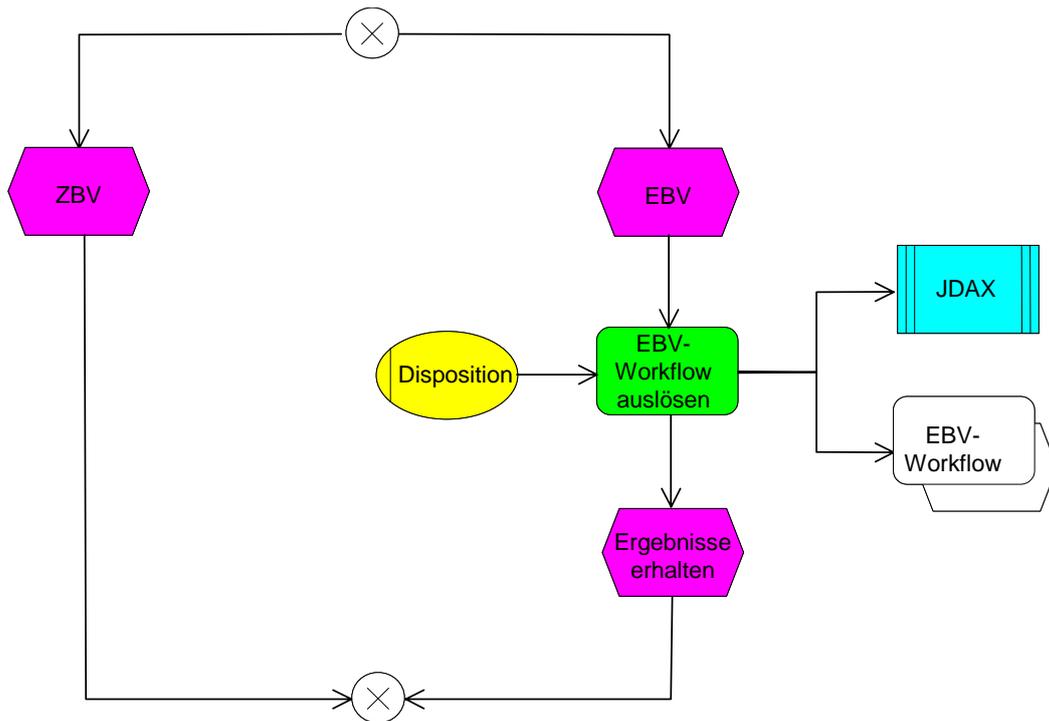


Abbildung 6-10: Segment 3-1: EBV-Workflow

Die Übertragung der Datei ist nur notwendig, wenn es sich bei der Aktion um eine Endbevorratung handelt.

Nachdem die verantwortliche Abteilung innerhalb des Systems JDAX die Daten überprüft hat, wird dem Disponenten ein Mengenvorschlag unterbreitet, wie hoch die BV-Menge sein sollte.

Im Anschluss an den EBV-Workflow schließt sich eine Aktivität des Einkaufs an, der in der folgenden Abbildung 6-11 zu sehen ist.

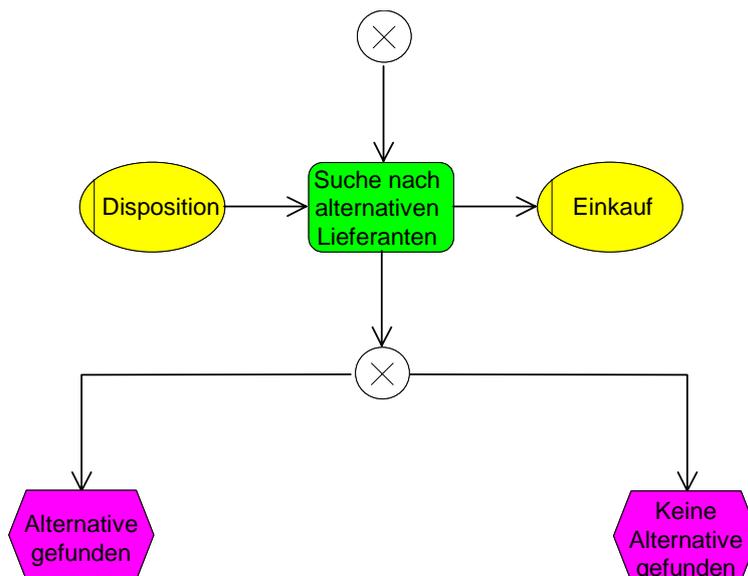


Abbildung 6-11: Segment 3-2: Suche nach alternativem Lieferanten

Die Abteilung Einkauf versucht einen alternativen Lieferanten zu finden, der zu einem den Serienkonditionen ähnlichen Preis eine weitere Teileversorgung vornehmen könnte oder aber zu einem günstigeren Preis eine Bevorratungsaktion durchführen kann. Sofern der Einkauf hier erfolgreich ist, muss wiederum eine neue Menge prognostiziert werden. Wenn kein neuer Lieferant gefunden werden kann, kann mit dem nächsten Segment fortgefahren werden.

6.3.5 Segment 4: Classic Center, Wirtschaftlichkeitsrechnung und Eskalation

Das folgende Segment umfasst die Tätigkeiten des Disponenten zur Bestimmung einer zusätzlichen Bevorratungsmenge, wenn eine Teilenummer Classic Center-relevant ist. Ein Teil wird dann Classic Center-relevant. Der Disponent muss beim Classic Center die zusätzlichen Bedarfe erfragen und auf die bisher prognostizierten Werte aufaddieren. Die kommende Abbildung 6-12 zeigt das Segment 4:

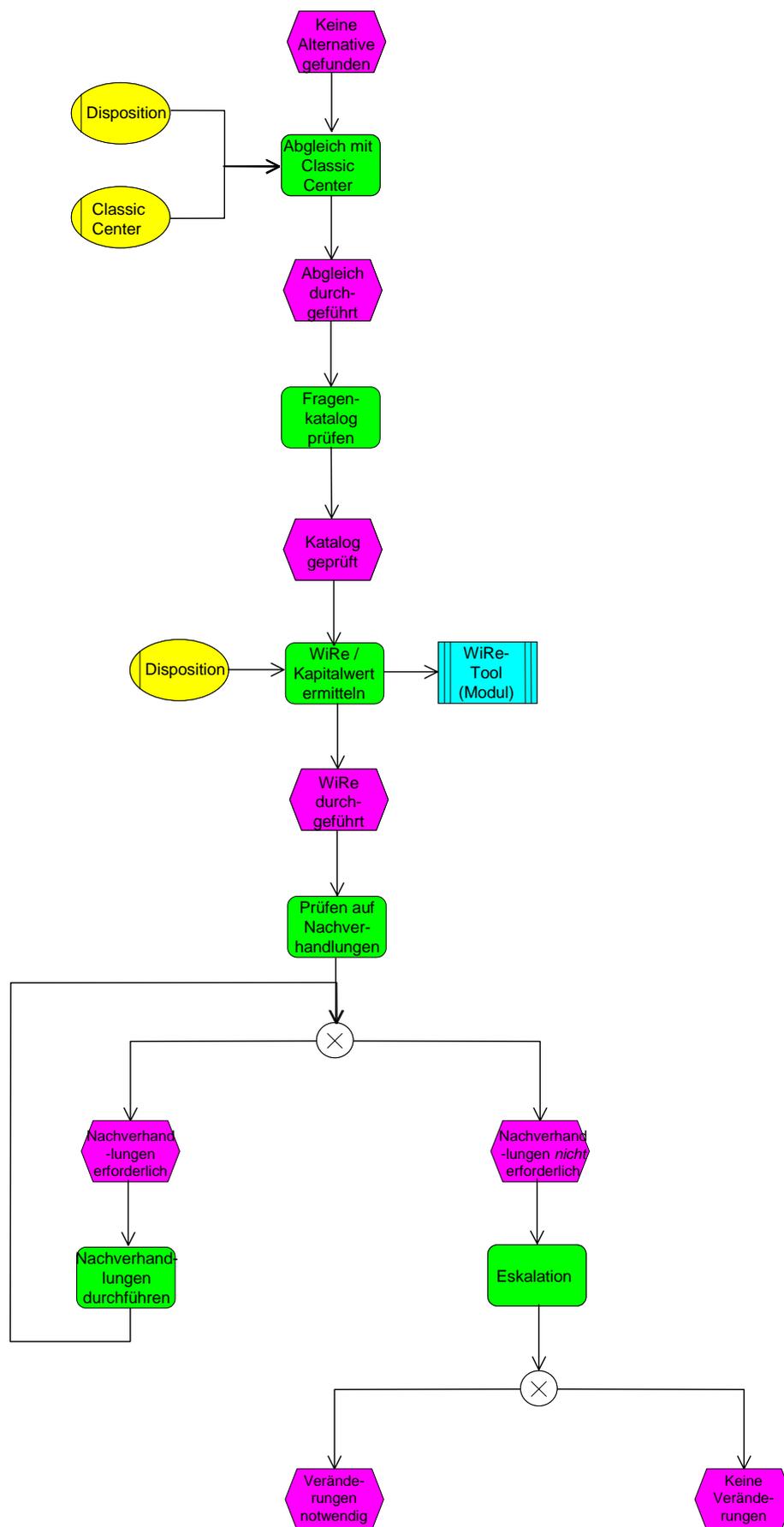


Abbildung 6-12: Segment 4: Classic Center, WiRe und Eskalation

Nach dem Abgleich mit dem Classic Center muss der Disponent eine Menge von Fragen in einem Fragenkatalog beantworten. Dies ist zu seiner Unterstützung und dient dazu, durch die Beantwortung der Fragen einfache Fehler zu verhindern.

Nach dieser Beantwortung, wird eine Wirtschaftlichkeitsrechnung auf der Basis der veranschlagten Bevorratungsmenge und der Verbrauchshistorie durchgeführt. Das Ziel dieser Berechnung ist die Ausweisung eines Kapitalwerts zu einer Teilenummer. Der Kapitalwert, kurz KW, gibt Aufschluss darüber, ob die Bevorratung wirtschaftlich ist oder nicht.

Der Output der regulären WiRe umfasst den optimalen Bevorratungszeitraum, den Kapitalwert und die durch die Bevorratungsmenge belegte Fläche. Ist der Kapitalwert zu niedrig oder kleiner Null, so wird in Nachverhandlungen mit dem Lieferanten versucht einen niedrigeren Preis zu erzielen, da der KW damit steigen würde. Wurde dies geschafft, so muss die WiRe erneut gerechnet werden. Kann kein niedrigerer Preis erzielt werden, muss der Disponent in die anschließende Eskalation.

Wird im Zuge einer Eskalation festgelegt, dass Veränderungen an Menge oder anderen Kriterien bei einer Bevorratungsaktion notwendig sind, muss der Disponent wieder zu Beginn der Schleife einsteigen und mit einer neuen Prognose beginnen. Im anderen Fall kann er den Prozess weiter verfolgen und sich der Werksplanung widmen.

6.3.6 Segment 5: Werksplanung, Ende der Schleife und des Prozesses

Das letzte Segment des Soll-Prozesses ist durch die Werksplanung beschrieben. In der Abbildung 6-13 werden die Tätigkeiten aufgezeigt, die der Disponent nach der Freigabe durch die Eskalation noch durchzuführen hat.

Sofern keine Überprüfung durch die Werksplanung erfolgen muss, kann der Disponent alle notwendigen Schritte zur Endbevorratung einleiten. Dazu werden alle Werte so eingefroren, wie sie aktuell im System verzeichnet sind. Sie werden entsprechend dokumentiert und die gesamte Bevorratungsaktion als abgeschlossen gekennzeichnet. Der Disponent führt danach alle Aktionen durch, die für eine Bevorratung notwendig sind – hierzu gehört z.B. das Einpflegen der Bevorratungsmengen als Bestellmengen das DuB.

Muss eine Überprüfung durch die Werksplanung erfolgen, so werden alle an die Werksplanung übertragen. Die Werksplanung gibt im Falle einer Überprüfung zu der bereits anvisierten BV-Menge noch einen eigenen Vorschlag zu einer Menge. Differieren die Menge zu stark, muss der Disponent noch mal zum Beginn der Schleife zurückgehen und erneut mit einer angepassten Prognose den gesamten Prozess durchlaufen. Ziel dieser Schleifen ist die Reduzierung von potenziellen Fehlentscheidungen, die damit zu besser angepassten Bevorratungsmengen führen. In diesem Falle kann natürlich auch ein nicht unerheblicher Aufwand an Arbeitskraft und Zeit beim Disponenten verloren gehen. Dieser Mehraufwand

kann sich aber schon in einer einzigen Bevorratungsaktion wieder auszahlen, wenn mit den ständigen Mengenanpassungen z.B. die Kapitalbindung innerhalb einer BV-Aktion stark reduziert werden kann.

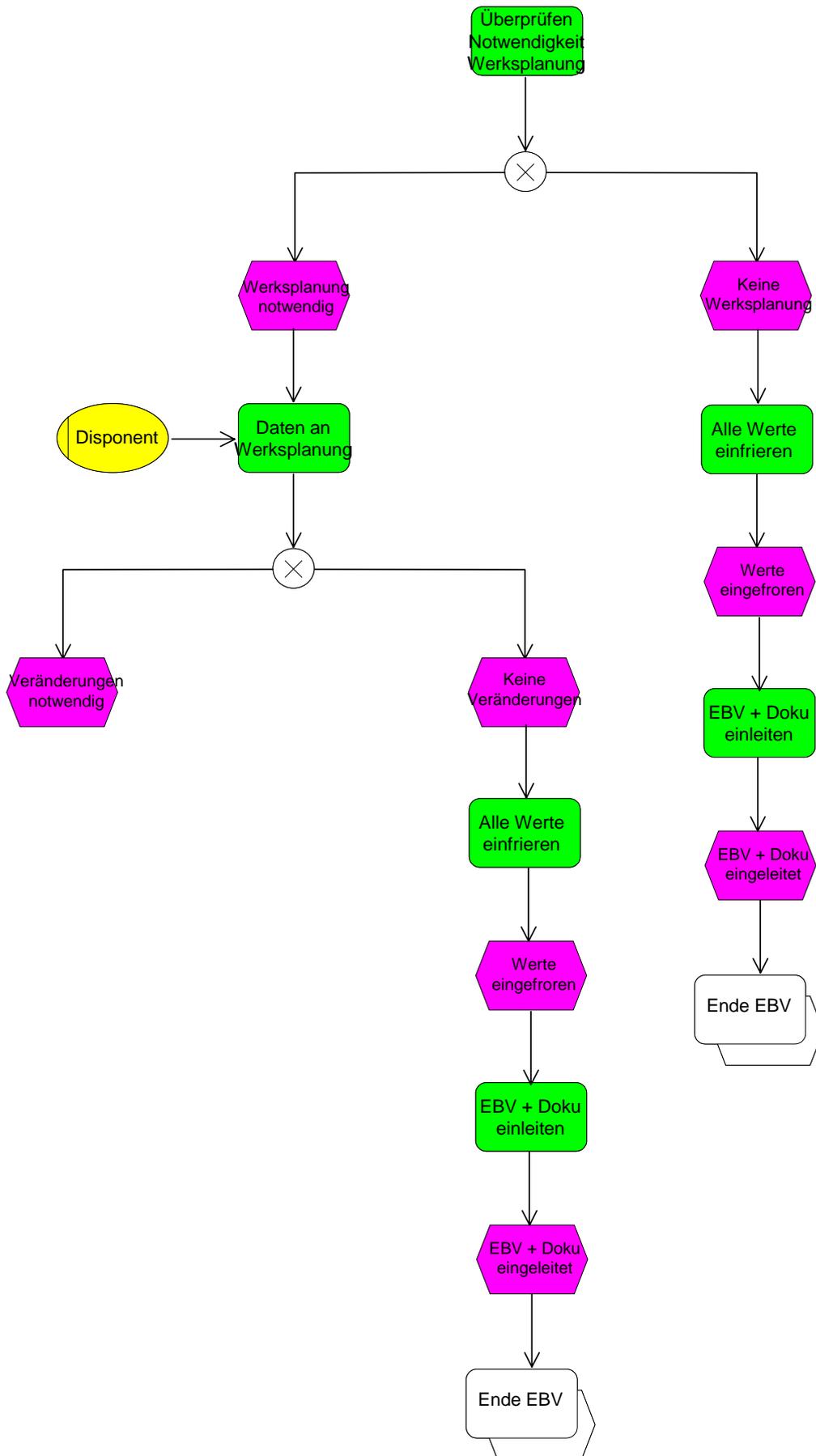


Abbildung 6-13: Segment 5: Werksplanung und Ende des Prozesses

Sofern keine weiteren Veränderungen notwendig sind, kann auch hier der Disponent alle Werte einfrieren und dann mit den notwendigen Aktivitäten zur Bevorratung eines Teils beginnen.

Der gesamte Soll-Prozess endet an dieser Stelle und hat das Ziel, die Planung für eine komplette End- oder Zwischenbevorratung durchzuführen, erreicht. Damit der Leser noch einen Einblick über die gesamte Länge der Prozesskette bekommt, sei an dieser Stelle noch eine stark verkleinerte Abbildung 6-14 aufgeführt, in welcher der Autor zur besseren Orientierung die einzelnen Segmente farblich und textuell hervorgehoben hat.

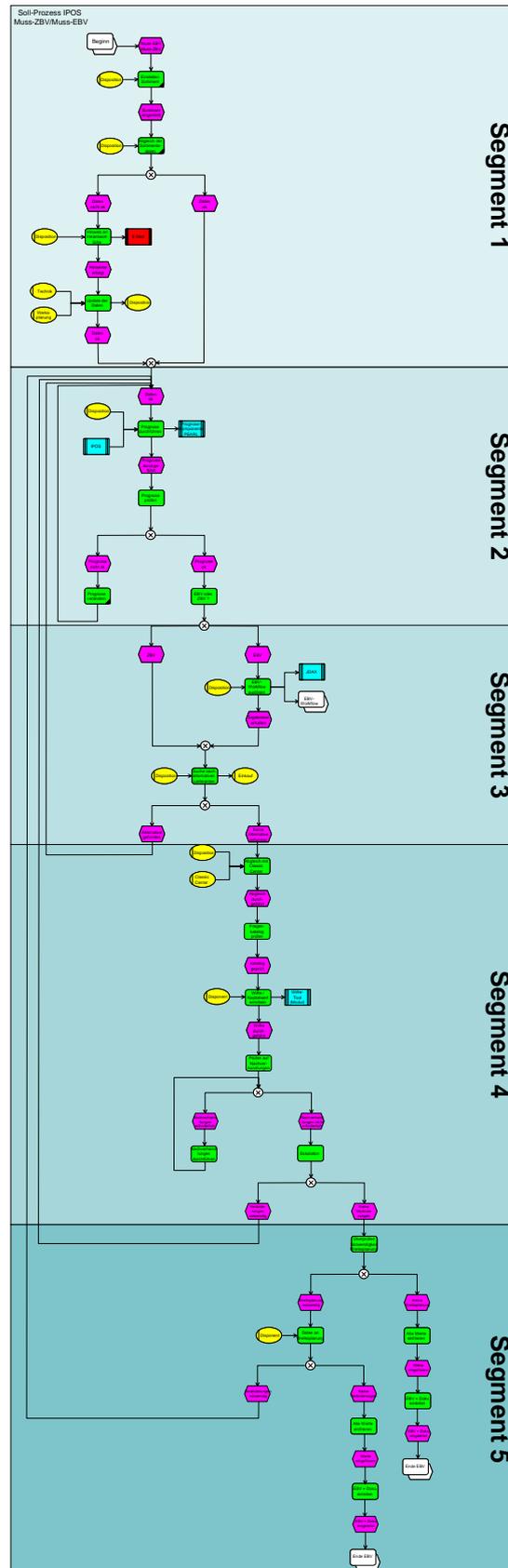


Abbildung 6-14: Überblick zum gesamten Soll-Prozess

In der Abbildung sind gut die großen Schleifen zu erkennen, die ein Disponent unter Umständen mehrmals durchlaufen muss. Im nächsten Kapitel wird geklärt, wie aus dem Soll-

Prozess heraus geeignete Informationen zur Abbildung in einem späteren System extrahieren werden können.

6.3.7 Extraktion der Stationen – Schnittstellendefinition

Eine Station beschreibt eine Menge von inhaltlich zusammenhängenden Ereignissen und Prozessen innerhalb einer eEPK, die zusammen zu betrachten sind und daher zusammengefasst werden können. Eine Station beschreibt auf diesem Wege einen speziellen Themenbereich des gesamten Soll-Prozesses. Extrahierte Stationen aus dem Soll-Prozess bilden die Punkte, an denen später die Workflowobjekte verändert werden. Eine Station ist immer gegenüber anderen Stationen abgegrenzt – es existieren keine Prozesse oder Ereignisse, die zu mehr als einer Station gehören.

Die Extraktion der Stationen erfolgt auf Basis eines Fachverständnisses heraus. Prozessteilnehmer müssen bei der Extraktion der Stationen teilnehmen und somit die fachliche Korrektheit sichern. Sie können mit dem theoretischen Hintergrund die entsprechenden Prozesse zu einer Station bündeln und zusammenfassen.

Im Folgenden werden die einzelnen Stationen des Soll-Prozesses extrahiert, wobei hierbei von Beginn des Soll-Prozesses bis zum Ende hin vorgegangen wird und sukzessive die Stationen nacheinander extrahiert werden.

6.3.7.1 Station 1 – „Daten in das System einspielen“

Im Falle der ersten Station ist die Extraktion einfach. Es handelt sich hier um die Station „Daten in das System einspielen“. Sie beschreibt exakt die Tätigkeiten, die ein Disponent durchzuführen hat, wenn ein neues Sortiment für eine BV-Aktion vorgesehen ist. Hierzu muss er alle Tätigkeiten durchführen, die im Segment 1 des Soll-Prozesses aufgeführt wurden sind.

Alle Tätigkeiten werden im System IPOS durchgeführt. Der Disponent hinterlegt zuerst das Sortiment im System, vergibt einen Namen und überprüft danach alle erforderlichen Daten. Falls Daten im System fehlerhaft sind bzw. nicht vorhanden sind, wird durch den Disponenten eine Benachrichtigung der technischen Abteilung per E-Mail durchgeführt. Diese hat die Aufgabe die Daten auf den neusten Stand zu bringen, fehlerhafte Daten zu ergänzen. Erst wenn alle Daten in Ordnung sind, kann der Disponent mit den Arbeiten der nächsten Station beginnen.

Innerhalb der ersten Station existieren keine Schnittstellen zu externen Softwaresystemen, die es zu beachten gilt.

6.3.7.2 Station 2 – „Prognose erstellen“

Die zweite Station beschreibt die Tätigkeiten, welche nötig sind, um für alle Teilenummern in einem Sortiment jeweils eine Prognose zu erstellen. Der Disponent muss hierzu von einer externen Prognose-Komponente eine Prognose für jede Teilenummer abfragen und begutachten. Befindet der Disponent eine Prognose für unzureichend, kann er sie durch verschiedene Maßnahmen weiter verändern. Dies muss er solange machen, bis er die Prognose als korrekt empfindet.

In dieser Station existiert eine Schnittstelle zu der externen Prognose-Komponente, die nicht im System IPOS entwickelt wird und daher mit einer vordefinierten Schnittstelle anzusprechen ist.

6.3.7.3 Station 3 – „JDAX“

Im Falle einer Endbevorratung einer Teilenummer aus einem Sortiment, müssen die entsprechenden Daten aus dem System IPOS in das System JDAX eingespielt werden. Dies geschieht per Schnittstelle. Die Übertragung erfolgt dabei ohne Interaktion mit dem Benutzer, der lediglich einer Übertragung zustimmen muss. Die technische Abteilung des GLC hat danach die Aufgabe im System JDAX alle Daten zu überprüfen und die Ergebnisse in das System IPOS zu übertragen.

Der Disponent hat in dieser Station noch eine Prozessschnittstelle zum EBV-Workflow, dessen Prozesse er ebenfalls auszuführen hat.

Handelt es sich bei einer BV-Aktion um eine ZBV, so wird diese Station übersprungen.

6.3.7.4 Station 4 – „Einkauf“

Die Abteilung Einkauf hat die Aufgabe bei einer bevorstehenden BV-Aktion die Suche nach einem günstigeren bzw. alternativen Lieferanten zu beginnen und die Ergebnisse dieser Suche in das System IPOS einzugeben.

6.3.7.5 Station 5 – „Classic Center und Fragenkatalog“

Die fünfte Station beinhaltet den Mengeabgleich mit dem Classic Center. Hierbei müssen die Verantwortlichen aus dem Classic Center ihre Mengevorschläge im System IPOS mit hinterlegen.

Anhand eines Fragenkataloges hat der Disponent nun die Aufgabe möglichst viele Fragen zu beantworten – dies geschieht im System IPOS. Die Antworten auf die Fragen werden mitprotokolliert und können somit später den Disponenten zugeordnet werden.

Nachdem dies geschehen ist, kann der Disponent mit der nächsten Station fortfahren.

6.3.7.6 Station 6 – „WiRe“

Eine BV-Aktion besitzt nach Auffassung der DCAG immer einen Kapitalwert, dessen Bestimmung die Aufgabe des Disponenten ist. Dies führt er mit Hilfe der WiRe durch.

Gleichwohl es sich bei der WiRe um eine externe Komponente handelt, wird sie per Schnittstelle in das System IPOS integriert. Die Übertragung der Daten und die Errechnung des wirtschaftlichen Wertes geschehen nach Betätigung durch den Disponenten vollautomatisch. Er hat die Aufgabe in dieser Station weiterhin zu prüfen, ob bei einem errechneten negativen oder zu geringen Kapitalwert Nachverhandlungen mit einem Lieferanten nötig sind. Dies kann generell zwar möglich sein, allerdings entfallen solche Verhandlungen natürlich bei insolventen Unternehmen.

6.3.7.7 Station 7 – „Eskalation“

Die Station der Eskalation kann nicht durch den Disponenten durchgeführt werden. Hierbei wird durch die Vorgesetzten eine Entscheidung getroffen und diese im System IPOS hinterlegt. Erst nachdem diese Entscheidung gefallen ist, kann der Disponent mit der Bearbeitung der nächsten Station beginnen.

6.3.7.8 Station 8 – „Werksplanung“

Die Werksplanung, sofern sie für eine BV-Aktion von Nöten ist, kann eine BV-Aktion noch nach einer Eskalation verhindern, wenn Fehler in der bisherigen Planung des Disponenten festgestellt werden.

Die Werksplanung führt hierzu auf Basis der im System IPOS hinterlegten Daten eigene Überprüfungen durch und entscheidet dann, ob eine BV-Aktion durchgeführt werden kann oder nicht.

Der Disponenten hat hier nur die Aufgabe bei einer erneuten Überprüfung der BV-Aktion mit einer erneuten Prognose nochmals den Prozess zu durchlaufen oder aber die Fixierung aller Werte im System IPOS zu veranlassen, die BV-Aktion zu dokumentieren und die letztendliche Bevorratung im System DuB einzuleiten.

Weitere Stationen existieren nicht.

Die Stationenextraktion war notwendig, um bei der nun folgenden Konzeptionierung des Systems IPOS auf die Gegebenheiten des Projektes einzugehen.

6.4 Konzeption des Workflow Enactment Services

Bevor mit der Entwicklung des Workflow Enactment Service begonnen wird, soll im ersten Abschnitt eine Erläuterung zu der grundlegenden Theorie der Beziehung der Komponenten im Projekt IPOS gegeben werden. Danach erfolgt die Entwicklung der Workflow Engine und danach des Workflow Enactment Service. Die Entwicklung des WAPI beendet den Abschnitt.

6.4.1 Das Konstrukt „Teilenummer-Workflow-Station-Prozess-Aktivität“

Die grundlegende Idee, die hinter dem gesamten Aufbau des Systems IPOS steckt, wird durch das in der Überschrift genannten Konstrukt „Teilenummer-Workflow-Station-Prozess-Aktivität“ gut charakterisiert. Der Leser erkennt hier schon die Hauptkomponenten, die dem gesamten Workflow zugrunde liegen und an ihm beteiligt sind.

Ausschlaggebend für alle im System IPOS stattfindenden Aktivitäten der Benutzer sind die Teilenummern, die zwar in einem Sortiment gruppiert und gekapselt werden können, jedoch die atomare Einheit darstellen, auf der das Hauptaugenmerk im gesamten System liegt. Letztlich geht es um die Zwischen-/Endbevorratung einer einzelnen Teilenummer, nicht um eine Bevorratung eines Sortiments. Daher bilden die Teilenummern den Ausgangspunkt der Betrachtung.

Pro Teilenummer wird nun ein Workflow zugeordnet, der die Bevorratungsaktion beschreibt. Je nach der Unterscheidung, ob es sich um eine Art der Zwischenbevorratung oder eine Art der Endbevorratung handelt, wird für den Workflow eine individuelle Stationskette aufgebaut, die dem Workflow zugeordnet wird. Innerhalb dieser Kette sind alle für diese Bevorratungsaktion relevanten Stationen aufgeführt. Die Stationsreihenfolge ist dabei bindend, sodass ein Workflow keine Station überspringen kann, auslassen kann oder die Reihenfolge verändern kann. Diese Stationskette wird im System IPOS als Stationsliste bezeichnet.

Nach dem Aufbau der Stationsliste wird eine Prozessliste aufgebaut, nach welcher der Workflow abgearbeitet werden muss. Diese Prozesskette ist eine logische Aneinanderreihung von Prozessen, die nacheinander abgearbeitet werden müssen und die voneinander abhängig sind. Dabei stellt jeder Prozess in dieser Kette eine Bearbeitung eines speziellen Tasks dar. Da es zu jeder Station mindestens eine Prozessbeschreibung geben wird, werden in der Liste mindestens so viele Prozesse zu finden sein, wie Stationen in der Stationsliste. Darüber hinaus sind in der Prozessliste noch weitere Prozesse zu finden, wie z.B. solche, die keiner Station zugeordnet werden. Sie laufen unabhängig vom Benutzer bzw. von Fremdsystemen ab. Prozessliste und Stationsliste korrespondieren miteinander und können nicht losgelöst voneinander existieren.

Jeder Prozess besteht aus einer oder mehreren Aktivitäten. Zur Erreichung eines Prozessziels müssen hierbei verschiedene Handlungen durchgeführt werden, die in der Philosophie von IPOS als Aktivitäten bezeichnet werden. Jede Aktivität ist hierbei eine atomare Handlung des Nutzers oder der Applikation. Hier offenbart sich der Bezug zur WfMC, die in ihrer Philosophie dieselbe Hierarchie zwischen Prozessen und Aktivitäten sieht. Diese Konformität zwischen IPOS und dem Referenzmodell der WfMC ermöglicht es, viele der Prinzipien der WfMC konsistent zu übernehmen und nur in einigen Punkten dem gesamten Projekt anpassen zu müssen.

Dieses ganze Konstrukt kann leicht an einem Beispiel verdeutlicht werden. Folgende Abbildung 6-15 zeigt das Konstrukt auf einen Blick mit allen relevanten Informationen:

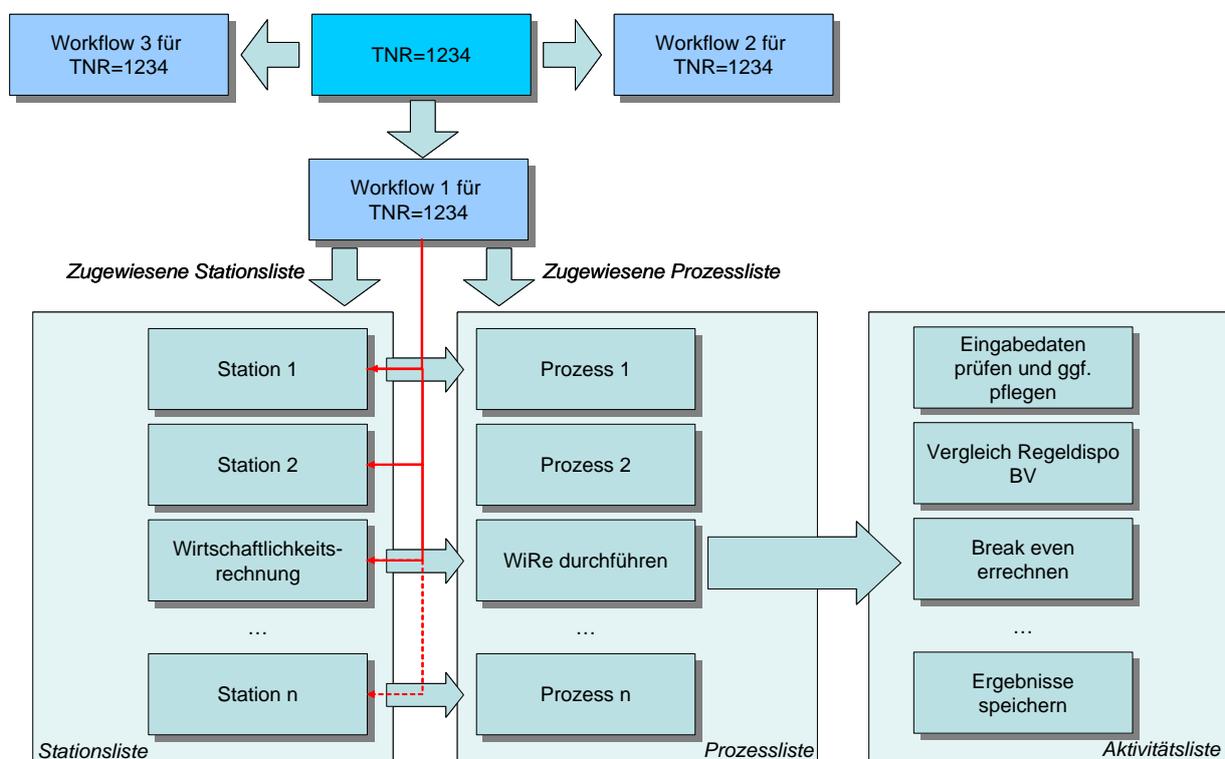


Abbildung 6-15: Konstrukt Teilenummer-Workflow-Station-Prozess-Aktivität

Der Ausgangspunkt der Betrachtung stellt in diesem Fall die Teilenummer (in der Abbildung mit der Abkürzung TNR versehen) dar. Hier wird eine beispielhafte Teilenummer verwendet: 1234. Dieser Teilenummer sind nun insgesamt drei Workflows zugeordnet, die alle rund um die TNR angeordnet sind. Letztlich geschieht eine Zuordnung eines neuen Workflows zu einer TNR dann, wenn diese Teilenummer in einem anderen Sortiment nochmals verwendet wird und durch einen Disponenten im späteren System IPOS hinterlegt wird.

In der Abbildung wird die Betrachtung für den Workflow 1 weitergeführt, welchem an dieser Stelle zwei Listen zugeordnet sind: eine Stationsliste und eine Prozessliste. In der Stationsliste sind alle Stationen aufgeführt, die durch den Workflow nacheinander durchlaufen werden müssen. Der Pfad des Workflows, symbolisiert durch die roten Pfeile in

der Zeichnung, ist somit vorgeschrieben. Eine Beispielstation ist in dieser Abbildung mit aufgeführt: die Station der Wirtschaftlichkeitsrechnung. Anhand dieser Station soll das weitere Prinzip erklärt werden.

Im Abschnitt 5 wurde der Soll-Prozess einer EBV diskutiert und dort die Prozesse, die für eine EBV ausgeführt werden müssen, aufgezeigt – die Wirtschaftlichkeitsrechnung war einer davon. Sie erschien erst in einem späteren Abschnitt des gesamten Bevorratungsprozesses, spielt aber eine wichtige Rolle. Sie unterteilt sich in ihrer Struktur in verschiedene Aktivitäten. Um nun diese Aktivitäten alle zu erfassen und im System IPOS zu hinterlegen, wird der Station „WiRe“ ein Prozess in der Prozessliste zugeordnet. Die Bezeichnung des korrespondierenden Prozesses lautet „WiRe durchführen“. In der Abbildung ist dabei auch zu sehen, wie sich dieser Prozess noch weiter in Aktivitäten unterteilt, welche in einer Aktivitätsliste zu sehen sind. Hierzu gehört beispielsweise „Eingabedaten prüfen und ggf. pflegen“. Diese elementaren Aktivitäten werden in dem System IPOS hinterlegt und können dann einem Benutzer bzw. einer externen Applikation zugeordnet werden – die Zuordnung erfolgt ebenfalls auf Basis von Worklists, wird aber intern vom WorkflowManager geregelt bzw. interne implizit in den Komponenten hinterlegt.

Konnten alle Aktivitäten durch den Benutzer und die Fremdapplikationen ausgeführt werden, so ist der Prozess insgesamt beendet und ein nächster Prozess kann ausgeführt werden. Mit dem Abschluss eines Stations-gekoppelten Prozesses wird danach auch die Bearbeitung der Station für beendet erklärt. Ist der letzte Prozess in der Prozessliste beendet, so endet damit auch die gesamte Bearbeitung des Workflows und alle notwendigen Ziele sind damit erreicht.

6.4.2 Entwicklung der Workflow Engine

Die Workflow Engine steuert alle Prozesse innerhalb des Workflow Enactment Services. Zur Entwicklung wird sich an dieser Stelle wieder auf das Verhältnis von Teilenummer und Sortiment bezogen. Wie schon erwähnt, bleibt die Verbindung zwischen beiden Komponenten für den gesamten Lebenszyklus einer ZBV/EBV bestehen und soll somit auch als zentrale Komponente im System übernommen werden.

Die folgende Abbildung 6-16 zeigt die Teilenummer und das Sortiment als eigenständige Bestandteile der Workflow Engine des Systems IPOS. Beide Komponenten werden solange miteinander verbunden bleiben, wie das Sortiment oder die Teilenummer existieren und der Prozess nicht vollständig abgeschlossen ist.

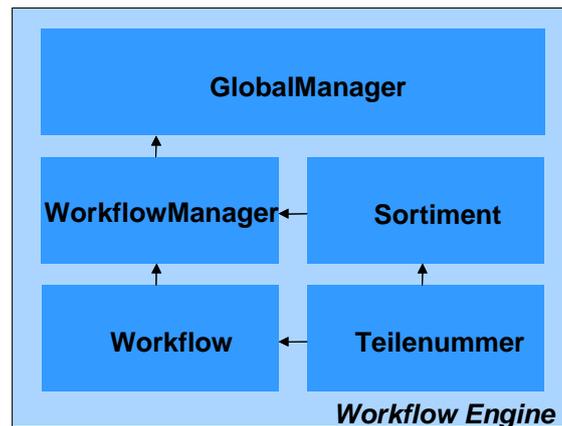


Abbildung 6-16: Workflow Engine des Systems IPOS

Obige Abbildung 6-16 zeigt aber noch mehr, als nur das simple Verhältnis der beiden ersten Komponenten. Es werden zum ersten Mal auch die korrespondierenden Systemkomponenten mit aufgeführt. Hier zu sehen sind der Workflow und der WorkflowManager, welche als systemseitige Vertreter der Teilenummer und des Sortiments dienen. Sie werden aktiv im System eingesetzt und gesteuert.

Ein Workflow übernimmt alle Aufgaben und Anforderungen, die an eine Teilenummer entstehen. Der WorkflowManager hingegen übernimmt die Steuerung der einzelnen Workflows eines Sortiments. Er überwacht, steuert und koordiniert die einzelnen Workflows entsprechend den Stations- und Prozesslisten.

Die Workflow Engine enthält noch eine weitere Komponente: den GlobalManager. Er stammt, wie auch der WorkflowManager, aus dem Pattern von [Mes97], welches in Abschnitt 6.2.2.1 beschrieben wurde. Die Aufgabe des GlobalManagers ist die Überwachung von globalen Bedingungen. Er sorgt beispielsweise für die Einhaltung der Geschäftsregeln. Durch seine Aktivität wird verhindert, dass falsche Stationslisten für einen Workflow erzeugt werden.

Die Workflow Engine kann allerdings nicht ohne weitere Komponenten arbeiten. Für die Abarbeitung der zu einem Workflow gehörenden Prozess- und Stationslisten müssen weitere, nicht in der Workflow Engine realisierte Komponenten hinzugezogen werden. Diese Komponenten sind im Workflow Enactment Service zu finden.

6.4.3 Entwicklung des Workflow Enactment Services

Die Workflow Engine wird in diesem Unterabschnitt genutzt, um im Weiteren den Workflow Enactment Service zu konzipieren.

Die kommende Abbildung 6-17 zeigt die Workflow Engine und die Erweiterungen für den Workflow Enactment Service auf einen Blick.

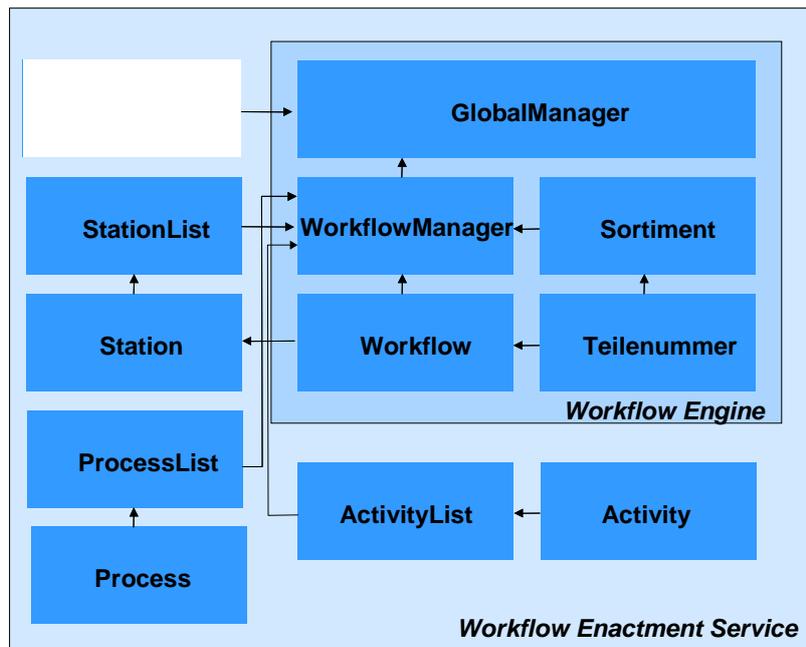


Abbildung 6-17: Workflow Enactment Service und Workflow Engine auf einen Blick

Betrachtet der Leser den Workflow Enactment Service, so kann leicht festgestellt werden, dass fast alle Komponenten bereits in einem der vorherigen Abschnitte erwähnt wurden. Trotzdem soll deren Funktion hier noch einmal kurz umrissen werden:

- Station: Eine Komponente, die inhaltlich zusammengehörige Funktionen und Ereignisse bündelt. Eine Station im System IPOS repräsentiert ein Objekt für die in der theoretischen Station durchzuführenden Aktivitäten.
- StationList: Die Liste von Stationen gibt die Vorgabe für einen Workflow entsprechend der Bevorratungsart.
- Process: Ein Process ist entweder intern einer Station zugeordnet oder beschreibt eine autonome Tätigkeit des WFMS.
- ProcessList. Die ProcessList beinhaltet alle notwendigen Prozesse, die ein Workflow während der Ausführung durchzuführen hat.
- Activity: Eine Aktivität ist eine elementare Tätigkeit innerhalb des Systems IPOS.
- ActivityList: Die Liste von Aktivitäten hängt von den einzelnen Einträgen der ProcessList ab und führt alle notwendigen, elementaren Tätigkeiten für einen Prozess in der Prozessliste auf.

Eine neue Komponente findet sich nur verdeckt in dem äußeren Kern. Es handelt sich um die Komponente „BusinessLogic“. Der Autor hat hier die Verwendung einer Komponente vorgesehen, die alle relevanten impliziten und expliziten Geschäftsregeln enthält. Im Falle einer Realisierung kann der GlobalManager auf diesem Wege alle notwendigen Regeln abrufen und ständig überprüfen. Sollte die Komponente nicht realisiert werden, so müssen die Regeln im GlobalManager hinterlegt werden bzw. implizit in der Programmierung des Systems mit einfließen. Die Verdeckung der Komponente fand aus dem Grund statt, dass

eine Realisierung wünschenswert wäre, aber bislang noch nicht vorgesehen ist. Darüber hinaus muss auch festgehalten werden, dass es sich bei dieser Komponente um eine sehr aufwendige Realisierung handeln würde, die viel Zeit in Anspruch nehmen würde. Eine Verwendung eines kommerziellen Produktes wäre möglich.

Der Workflow Enactment Services ist damit vollständig. Die Komponenten sind nun in der Lage Workflows anhand von Stations-, Process- und Activity-Listen zu bearbeiten. Anhand des Referenzmodells der WFMC ist aber die Notwendigkeit nach weiteren Komponenten zu erkennen. So fehlen bisher im System IPOS beispielsweise die Möglichkeiten externe Applikationen für die Bearbeitung der Workflows zu nutzen.

6.4.4 Workflow Application Programming Interface

Das Workflow Application Programming Interface, kurz WAPI, dient dem System IPOS dazu, eine Schnittstelle vorzugeben, die beispielsweise für die Nutzung externe Applikationen von Nöten ist. Die WAPI dient hierbei als „äußere Schicht“ für den Workflow Enactment Service und ermöglicht als komplexe Schicht durch verschiedene Komponenten einen einheitlichen, kontrollierten und gesteuerten Zugriff auf das System IPOS.

Die folgende Abbildung 6-18 zeigt den Workflow Enactment Service, inklusive Workflow Engine, und die WAPI, die sich in die vier Komponenten, interne API, externe API, Monitor und Repository unterteilt.

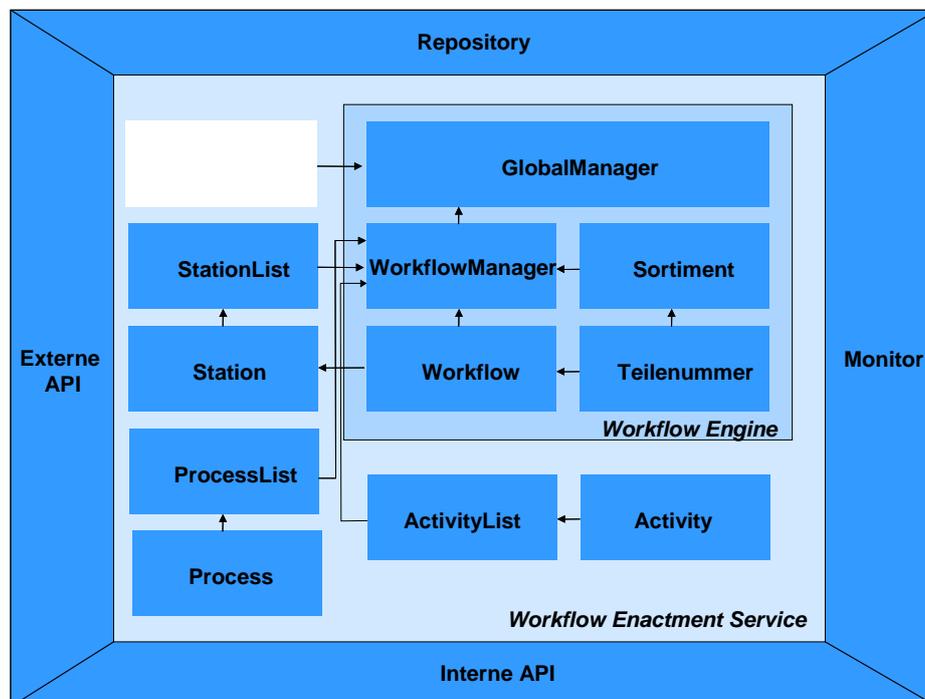


Abbildung 6-18: Umgebene WAPI

Die Komponenten haben folgende Funktionen und Aufgaben:

- Interne API: Das System IPOS wird nicht nur aus dem Workflow Enactment Service und der Workflow Engine bestehen, sondern noch viele andere Komponenten umfassen. Damit diese Komponenten mit dem Workflow Enactment Service kommunizieren können, existiert das interne API, welches nur für Komponenten, welche nicht innerhalb des Workflow Enactment Service, aber innerhalb des Systems IPOS liegen, verfügbar ist.
- Externe API: Das externe API steht den externen Anwendungen zur Verfügung, um beispielsweise aktuelle Werte zu den Workflows abzufragen.
- Monitor: Der Monitor erhält durch die internen Komponenten des Workflow Enactment Service ständig aktuelle Werte zum Zustand der Komponenten und meldet beispielsweise ein Fehlverhalten eines Workflows bzw. ein Fehlverhalten eines Benutzers. Externe Anwendungen zur Überwachung können somit schnell Informationen über die externe API abfragen.
- Repository: Das Repository hält ständig aggregierte Informationen bereit und stellt diese externen Applikationen zur Verfügung. Somit können externe Anwendungen zur Überwachung aggregierte Werte über das externe API abfragen.

Jegliche Interaktionen und Kommunikationen zwischen dem Workflow Enactment Service und den externen Komponenten und Anwendungen, müssen über das WAPI ablaufen. Somit ist der Workflow Enactment Service vor unautorisierten Aufrufen und Anfragen sicher.

6.5 Zusätzliche Komponenten im WFMS

Zusätzlich zu den bisher entwickelten Komponenten, die für den WorkflowEnactment Service notwendig sind, müssen noch weitere Komponenten im System IPOS integriert werden. Dazu gehören die Benutzer, das Portal, der MappingManager und die Stationskomponenten. Alle diese Komponenten werden in den folgenden Abschnitten erläutert.

6.5.1 Der Benutzer

Da die Nutzer im System IPOS verschiedene Rechte besitzen sollen, wird diese unterschiedliche Kompetenz der Benutzer durch die Einführung von Untertypen im System hinterlegt. Das bedeutet, dass es je Nutzergruppe jeweils eine Klasse mit den entsprechenden Methoden, Eigenschaften und Rechten existiert. Die Untertypen sind:

- Disponent
- Administrator
- Management
- Techniker

- ClassicCenter

Sofern eine weitere Nutzergruppe im System hinterlegt werden soll, kann ein neuer Typ im System hinterlegt werden und die Nutzer entsprechend erzeugt werden.

6.5.2 Das Portal

Das Portal besitzt die Aufgabe die Eingaben der User entsprechend ihren Vorgaben zu validieren und abzugleichen, sprich: ist das richtige Passwort eingegeben worden? Sofern ein User sich korrekt im System angemeldet hat, wird eine Klasse entsprechend den Vorgaben des Users instanziiert. Danach erhält der User Zugriff auf das interne System über das interne API.

Das Portal besitzt nur eine Verbindung zum internen API, wodurch ein einheitlicher Zugriff möglich ist.

Weiterhin hat das Portal die Aufgabe, alle Anforderungen des Benutzers weiterzugeben und an das interne API zu leiten. Dieses leitet danach alle Anforderungen an das restliche System weiter. Der so vorgeschriebene Pfad ist vorteilhaft, da somit die Kommunikation zwischen den Komponenten immer klar gegliedert ist.

6.5.3 Der MappingManager

Der MappingManager arbeitet ebenfalls wie das Portal mit einem zentralen Zugriff für alle Komponenten mittels der internen API und hat folgende Aufgaben im System:

- Abbilden der Komponenten auf ein relationales Datenbankschema
- Laden und Speichern aller angeforderten Daten für die Komponenten und die Arbeit der Komponenten
- Zentraler Zugriff auf die Daten für alle Komponenten
- Zugriff und Abgleich mit externen Datenquellen

Der erste Punkt in der Aufgabenbeschreibung der MappingManagers ist hierbei einer der wichtigsten. Die Speicherung aller notwendigen Daten in ein entsprechendes Schema, in dem keine Informationen verloren gehen und die einzelnen Komponenten schnell und unkompliziert wieder abgefragt und aufgebaut werden können, ist einer der Hauptaufgaben des MappingManagers.

Der zweite Punkt des Ladens aller Daten und des Speicherns der Daten in dem Datenbanksystem ist natürlich einer der einfacheren Punkte. Hierzu werden die entsprechenden Prozeduren bereitgestellt und im MappingManager hinterlegt. Durch die partielle Nutzung von so genannten Stored Procedures im Microsoft SQL-Server, der die

Datenbankumgebung darstellt, können die entsprechenden Abfragen und die Anpassung der Ergebnisse schnell und einfach durchgeführt werden.

Der zentrale Zugriff auf die Komponenten wird durch die Verwendung des MappingManagers durch alle restlichen Komponenten gewährleistet. Hierbei lag der Hauptaugenmerk vor allem auf einen einheitlichen Datenzugriff. Nicht jede Komponente soll innerhalb des Systems Zugriff auf Daten haben. Es muss vorher der Zugriff spezifiziert werden und genau geklärt werden, welche Daten tatsächlich gebraucht werden. Dieser zentrale Zugriff eröffnet eine gute Steuerung der Datenbewirtschaftung.

6.5.4 Stationskomponenten

Im Abschnitt 6.3.7 wurden bereits die Stationen genannt, die bei der Extraktion aus dem Soll-Prozess entstehen. Zu jeder dieser Station müssen verschiedene Aktivitäten durchgeführt werden, die in der ActivityList des zugeordneten Prozesses zu finden sind. Hierbei ist wichtig, dass bei der Ausführung der Tätigkeiten verschiedene Daten nötig sind, als auch verschiedene Daten als Ausgangsergebnisse der Aktivitäten entstehen. Diese Daten müssen gespeichert werden und in dem Komponentenmodell durch die Komponenten hinterlegt werden.

Es entstehen also für jede Station eine Komponente:

- Datensammlung
- Prognose
- JDAX
- Einkauf
- ClassicCenter
- WiRe
- Eskalation
- Werksplanung

All diese Komponenten repräsentieren jeweils die Eingangsdaten, als auch die Ausgangsdaten der Prozesse, die den Stationen zugeordnet sind. Weiterhin enthält jede Komponente einen Verweis auf den zugeordneten Prozess, der mit der Liste der Aktivitäten verbunden ist.

Nach der Einführung der restlichen Komponenten ergibt sich daher das folgende der bisherigen Architektur:

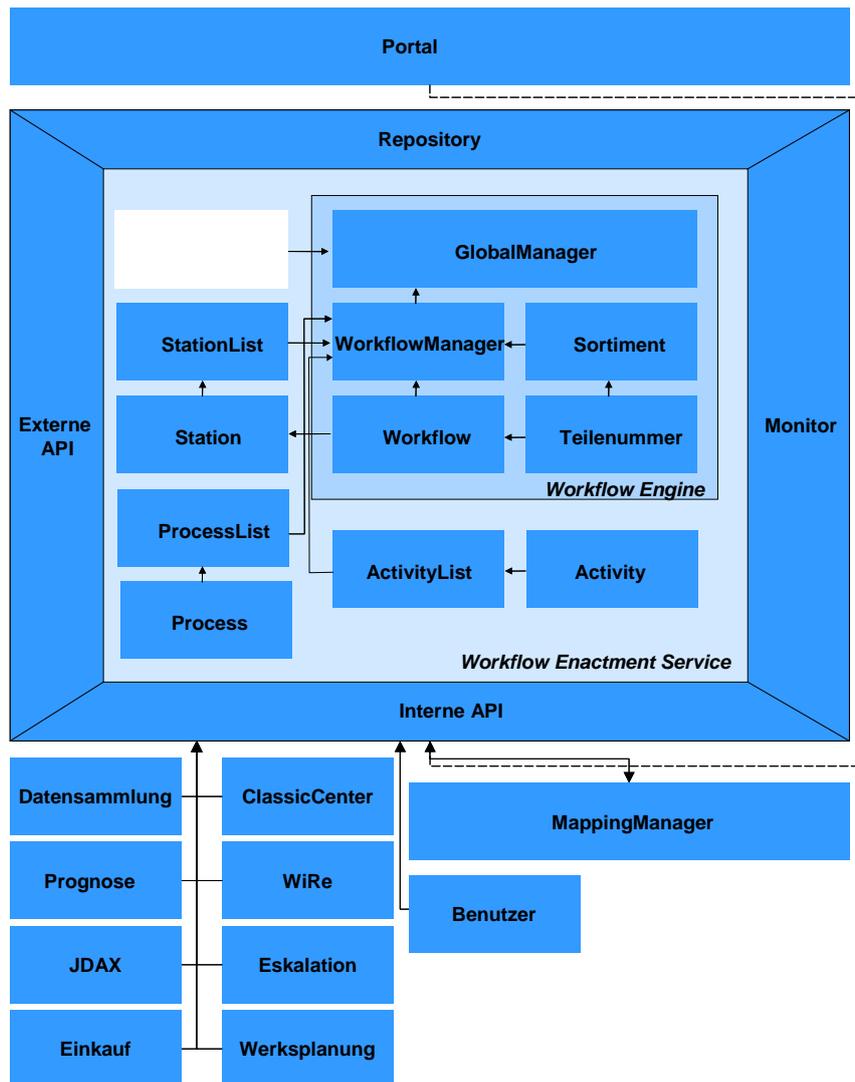


Abbildung 6-19: Workflow Enactment Service mit umgebener WAPI und zusätzlichen Komponenten

Hierbei sind alle zusätzlichen Komponenten in der obigen Abbildung zu sehen. Gut zu sehen ist die Kommunikation der Komponenten mit dem internen API.

Die Abbildung zeigt die komplette verarbeitende Schicht des Systems IPOS auf einen Blick.

6.6 Überführung der Komponenten in Klassen

Um die bisher konzipierten Komponenten weiter zu verwenden, werden in diesem Abschnitt die Komponenten zuerst in Typen überführt, danach die Kardinalitäten und die Beziehungen zwischen den Komponenten erfasst und abschließend in ein Klassendiagramm überführt. Der Abschnitt wird mit einem umfassenden Klassendiagramm abschließen.

6.6.1 Erfassen der Kardinalitäten und Komponentenbeziehungen

Es gibt verschiedene Möglichkeiten die Beziehungen zwischen den Komponenten abzubilden. Innerhalb dieser Arbeit wird die Krähenfußnotation (siehe [Fow97]) genutzt, um Kardinalitäten und Beziehungen zwischen den Komponenten abzubilden.

Hierbei wird die so genannte Krähenfußnotation eingesetzt, da sie einen schnellen Überblick über die Kardinalitäten geben kann. An dieser Stelle sei die grafische Notation kurz erläutert.

Kardinalitäten werden immer durch die Minimum- und Maximum-Werte auf beiden Seiten einer Beziehung dargestellt. Für eine Null in einer Beziehung wird ein 0 eingesetzt. Für eine Eins wird ein | eingesetzt und für ein n bzw. m in einer Kardinalität wird ein ‚Krähenfuß‘ eingesetzt, der wie folgt aussieht: \llcorner

Aus dem Krähenfuß leitet sich auch Betitelung der Notation ab. Eine (0,n):(0,m)-Beziehung würde demnach wie folgt aussehen:

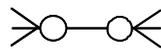


Abbildung 6-20: n:m-Beziehung in Krähenfußnotation

In dieser Beziehung sind 0 bis n Komponenten auf der linken Seite mit 0 bis n Komponenten auf der rechten Seite verbunden.

Zuerst werden die einzelnen Komponenten in Beziehung gesetzt und danach die Kardinalitäten erarbeitet. Aus Gründen der Übersicht wird im Folgenden auf eine Abgrenzung von Workflow Engine, Workflow Enactment Service und den restlichen Komponenten in der Abbildung verzichtet. Die folgende Abbildung 6-21 zeigt die Kardinalitäten und Beziehungen aller Komponenten.

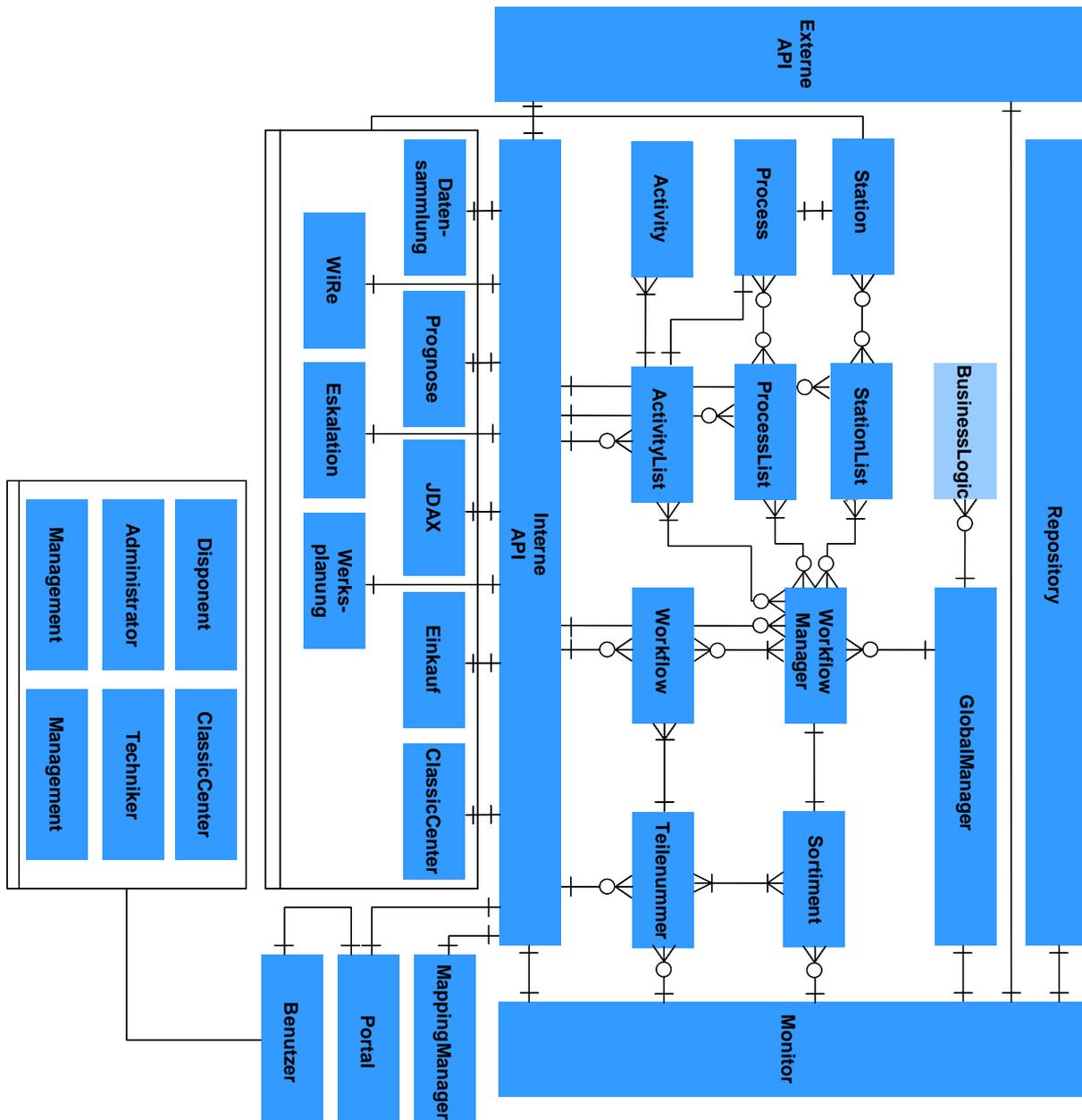


Abbildung 6-21: Beziehungen und Kardinalitäten zwischen Komponenten

An dieser Stelle sollen die Beziehung nur kurz umrissen werden, aber alle komplett durchgegangen werden, um dem Leser einen Überblick zu geben, wie die einzelnen Komponenten interagieren.

Die Betrachtung startet mit dem GlobalManager, als wesentliche Komponente der Workflow Engine. Der GlobalManager steht mit den Komponenten BusinessLogic, WorkflowManager, und Monitor in Beziehung. Alle Beziehungen sind vom Typ 1:n bzw. 1:1. Dies drückt die Einmaligkeit des GlobalManagers aus. Er wird nur ein Mal im System existieren. Von der Komponente BusinessLogic können verschieden viele Instanzen existieren, aber auch gar keine. Der GlobalManager besitzt ebenfalls eine Beziehung zum Monitor, welcher ständig Informationen empfängt und bereit hält. Als letzte Verbindung besitzt der GlobalManager noch eine Beziehung zum WorkflowManager (1:n). Eine Beziehung des GlobalManagers zu

verschieden vielen Instanzen der WorkflowManager zeigt hier die Funktionalität des GlobalManagers als Instanz zum Überwachen, Steuern und Kontrollieren aller Vorgänge, welche die WorkflowManager durchführen sollen. Es muss zu einem GlobalManager nicht unbedingt ein WorkflowManager zugeordnet sein, aber im Standardfall wird von sehr vielen WorkflowManagern ausgegangen.

Interessant wird nun die Verbindung der WorkflowManager zu den Sortimenten. Hier erkennt man leicht eine typische 1:1-Beziehung. Jedem Sortiment wird automatisch ein WorkflowManager zugeordnet. Da jedes Sortiment über n Teilenummern verfügen kann (mindestens jedoch über eine), müssen dementsprechend viele Workflows erstellt werden. Die Kardinalität zwischen den Workflows und dem WorkflowManager ist äquivalent dem Verhältnis zwischen Sortiment und Teilenummer. Workflow und WorkflowManager können darüber hinaus auch auf die interne API, welche als zentrale Instanz die Datenweitergabe zu den nicht im Workflow Enactment Service liegenden Komponenten steuert.

Verfolgt man vom WorkflowManager ausgehend weiter die Beziehungen zu der Komponente StationList, so stellt man hier eine n:m-Beziehung fest. Viele StationLists können vielen WorkflowManagern im System zugeordnet sein, wobei im Endeffekt immer mindestens eine StationList zugeordnet sein muss. Zu jeder StationList gibt es eine Menge von Stationen. Jedoch kann auch eine StationList leer sein und somit keine Station enthalten.

Je Station existiert ein Prozess, der die Tätigkeiten an dieser Station umfasst. Ein Prozess kann in mehreren ProcessLists auftauchen, kann aber auch in keiner ProcessList erscheinen. Sofern beispielsweise eine Station nicht verwendet wird, so wird auch der Prozess nicht verwendet und ist damit in keiner Prozessliste zu finden. Je WorkflowManager existiert aber immer eine ProcessList, welche die Prozesse, die mit den Stationen korrespondieren und mit anderen Tätigkeiten zu vergleichen sind, enthalten. Je Prozess existiert dazu eine ActivityList, die darüber hinaus viele unterschiedliche Tätigkeiten enthalten kann. Jede Aktivität muss aber mindestens einer ActivityList zugeordnet sein.

Die Stationen werden in der Abbildung noch weiter durch die Auflistung von Untertypen verfeinert. So sind die Komponenten Datensammlung, Prognose, JDAX usw. Untertypen von Stationen. Alle diese Komponenten besitzen eine Verbindung zum internen API. MappingManager und Portal besitzen ebenfalls eine Verbindung zu dem internen API; damit ist für diese beiden Komponenten eine Verbindung zu den anderen Komponenten innerhalb des Workflow Enactment Services möglich.

Der Monitor besitzt Beziehungen zu den wichtigsten Komponenten, von denen er Informationen beziehen kann. Wichtig sind besonders die Verbindungen zur Workflow Engine, da hier alle Informationen zu den Workflows abgefragt werden können.

Das externe API besitzt nur zwei Verbindungen: zum Monitor und zum internen API. Durch die letzte Verbindung können jegliche Anfragen von außerhalb des Systems weitergeleitet werden.

Die Komponente der Nutzer wird hier durch die Auflistung von Untertypen verfeinert.

6.6.2 Überführung der Beziehungen

Die folgende Abbildung 6-22 zeigt das aus dem Typdiagramm entwickelte Klassendiagramm in UML-Notation.

An dieser Stelle sollen nicht alle einzelnen Verbindungen explizit erläutert werden, sondern nur die wichtigsten. Dazu gehören vor Allem die Verbindungen innerhalb der Workflow Engine. Hier sind durch den Autor Komposition und Aggregation als Verbindungen eingesetzt wurden. Dies hat am Beispiel des GlobalManagers und des WorkflowManagers folgenden Grund. Bei der späteren Umsetzung der Verbindung zwischen diesen beiden Komponenten muss sichergestellt werden, dass beim Löschen des GlobalManagers keine weiteren Instanzen der WorkflowManager bestehen, denn diese können/sollen ohne einen GlobalManager nicht existieren und arbeiten können. Die Aggregation bietet hier eine gute Möglichkeit diese Abhängigkeit umzusetzen.

Die Verbindung zwischen Sortiment und WorkflowManager ist ebenfalls interessant. Hier handelt es sich um eine Komposition (strenge Aggregation). Da ein WorkflowManager nur existieren soll, wenn er einem Sortiment zugeordnet ist, ist die Komposition der richtige Weg diese Abhängigkeit zu modellieren.

Bei der Entwicklung des Klassendiagramms wurde stets auf die Einhaltung der Kardinalitäten geachtet. Aus speziellen Kardinalitäten und Beziehungen, wie z.B. zwischen Sortiment und WorkflowManager wurden spezielle Formen der Abhängigkeiten eingesetzt.

Die Vorarbeit der Entwicklung eines Typdiagramms muss nicht immer geschehen, hat sich in diesem Falle aber als nützlich erwiesen. Anhand der detaillierten Bestimmung der Kardinalitäten waren verschiedene spezielle Verbindungstypen schnell ersichtlich und konnten im Klassendiagramm adäquat umgesetzt werden.

6.6.3 Funktionen der Klassen in der Übersicht

Damit der Leser einen Überblick zu den Klassen und deren Funktionen im System IPOS erhält, sollen an dieser Stelle die Klassen sukzessive behandelt werden und deren Funktionen nochmals kurz erläutert werden.

Tabelle 6-1: Übersicht zu den Funktionen der Klassen

Klasse	Funktionsbeschreibung
<i>Workflow Engine</i>	
GlobalManager	Die Klasse überwacht alle Aktivitäten der restlichen Komponenten in der WorkflowEngine.
WorkflowManager	Die Klasse steuert die Ausführung der Workflows und kontrolliert die Einhaltung der Stationsreihenfolge.
Workflow	Die Klasse übernimmt alle Daten, die beim Ablauf der einzelnen Stationen auf Ebene der Teilenummer entstehen.
Sortiment	Die Klasse definiert das zugrunde liegende Sortiment.
Teilenummer	Eine Elementare Klasse zur Datenhaltung aller Daten für eine Teilenummer.
<i>Workflow Enactment Service</i>	
Station	Abstrakte Klasse für alle Stationen.
StationList	Die Klasse definiert eine Abfolge von Stationen.
Process	Abstrakte Klasse für alle Prozesse.
ProcessList	Die Klasse definiert eine Abfolge von Prozessen.
Activity	Abstrakte Klasse für alle Aktivitäten.
ActivityList	Die Klasse definiert eine Abfolge von Aktivitäten.
BusinessLogic	Die Klasse definiert Geschäftsregeln.
<i>WAPI</i>	
InterneAPI	Die Klasse leitet alle Anfragen an die internen Komponenten weiter.
ExterneAPI	Die Klasse steht für externe Applikationen zur Verfügung.
Monitor	Die Klasse sammelt Informationen von allen Komponenten im System.
Repository	Die Klasse stellt aggregierte Informationen über das System bereit
<i>Externe Komponenten</i>	
MappingManager	Die Klasse koordiniert den Zugriff auf Daten.
Portal	Die Klasse regelt den Zugang zum System für die Benutzer.
Benutzer	Abstrakte Klasse für alle Benutzer des Systems.

Obige Tabelle zeigt alle Klassen nochmals auf einen kurzen Überblick mit den Funktionen.

Wie die einzelnen Klassen in einem speziellen Anwendungsfall interagieren und wie die Klassenhierarchie aufgebaut ist, soll im Folgenden durch ein Diagramm visualisiert werden.

Hierbei handelt es sich um die Situation des Benutzers, der innerhalb des Systems IPOS einen Workflow für die Abarbeitung einer Station starten will. Das Diagramm zeigt weiterhin die Aktionen, die zwischen den einzelnen Klassen stattfinden.

Bei dem Diagramm handelt es sich um ein gemeinsames Klassen-, Sequenz- und Aktivitätsdiagramm, welches sich an der Darstellung in [Fern00] orientiert. Die folgende Abbildung zeigt die Legende zu dem Diagramm.

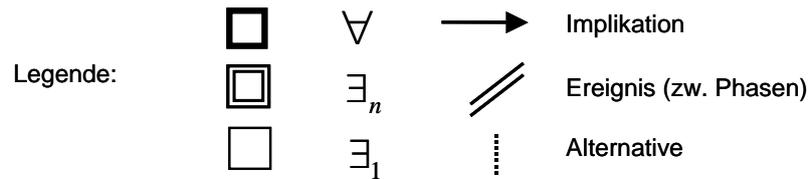


Abbildung 6-23: Legende zum Diagramm

In der Legende sind auf der linken Seite die unterschiedlichen Darstellungsformen der Klassen zu sehen – unterschieden durch die Kardinalitäten. Rechts daneben sind die unterschiedlichen Aktionen und Ereignisse zu sehen.

Bearbeitung einer Teilnummer in Station

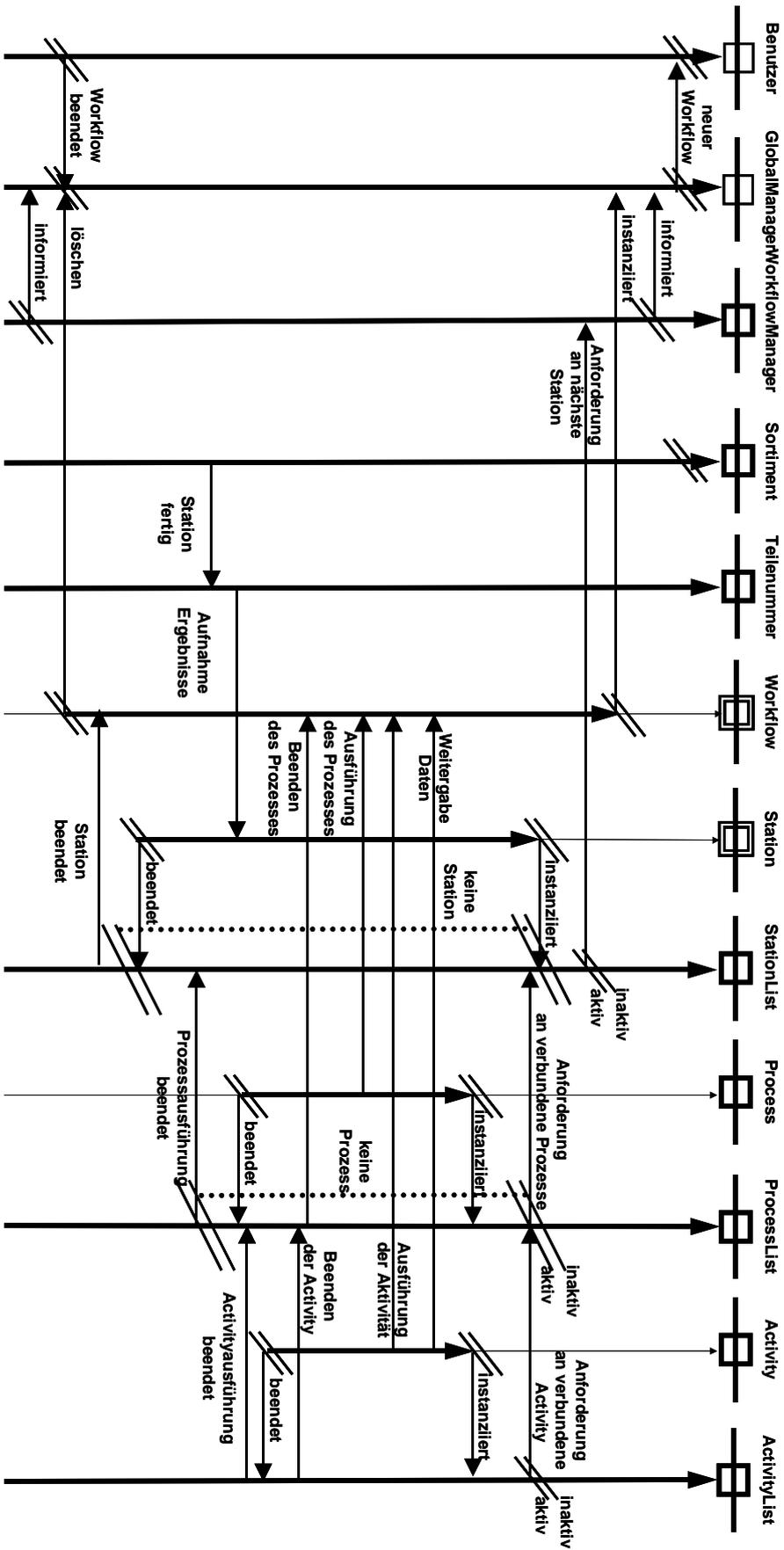


Abbildung 6-24: Gemeinsames Klassen -, Sequenz- und Aktivitätsdiagramm

Das obige Diagramm zeigt die Funktionsweise der einzelnen Komponenten auf einem Blick an. Der hier gewählte Fall ist ein typischer Anwendungsfall im System IPOS. Der Benutzer startet einen neuen Workflow, in dem die Klasse Benutzer dem GlobalManager die Aufgabe überträgt. Dieser informiert den WorkflowManager darüber und instanziiert danach einen neuen Workflow. Durch den WorkflowManager wird dann an die StationList eine Anforderung geschickt, dass die nächste Station, die für den Workflow relevant ist, aufzurufen ist. Die StationList instanziiert dann die nächste Station, sofern noch eine Station für den Workflow in der Liste vorgesehen ist. Andernfalls schickt die StationList sofort die Meldung, dass die Bearbeitung der Station schon vollendet ist.

Nachdem die Station instanziiert wurde, wird an die ProcessList eine Anforderung geschickt, den Process, der für die Station notwendig ist, zu instanziiieren. Sofern ein Process hinterlegt ist, wird dieser durch die ProcessList instanziiert, andernfalls bekommt die StationList die Rückmeldung, dass die Bearbeitung des Process bereits abgeschlossen ist. Die ProcessList sendet zeitgleich zur Instanziiierung des Process noch eine Anforderung an die ActivityList, alle Activity zu instanziiieren, die für die Abarbeitung des Process notwendig sind.

Nachdem die Klassen instanziiert wurden, werden alle Activity ausgeführt und der Process durchgeführt. Hierbei nimmt die Klasse Workflow alle relevanten Daten auf, die während der Ausführung der Activity entstehen. Nachdem alle Activity durchgeführt wurden, ist der Process damit beendet. Mit der Beendigung des Process, wird auch die Arbeit an der Station beendet und alle Ergebnisse der Arbeit werden an die Klasse Teilenummer weitergegeben. Nach der Beendigung der Arbeiten werden alle Instanzen wieder gelöscht und mit der Meldung an den Benutzer schließlich die Abarbeitung vollständig beendet.

Dies ist nur ein sehr spezifischer Anwendungsfall, der während der Arbeit des WFMS auftauchen kann. Im folgenden Abschnitt wird auf die allgemeineren Anwendungsfälle eingegangen.

6.7 Betrachtung der typischen Anwendungsfälle – die Interfaces der WFMC

Da die WFMC fünf Interfaces für ein WFMS definiert hat, soll in diesem Abschnitt untersucht werden, inwieweit das WFMS IPOS entsprechende Schnittstellen für die Anwendungsfälle bietet.

6.7.1 Neue Prozesses

Sollen neue Prozesse, als auch Stationen und Aktivitäten in das System IPOS eingepflegt werden, so muss dies durch die direkte Manipulation des Quellcodes geschehen. Das

System bietet kein Interface, welches das Einpflegen neuer Prozesse, Stationen oder Aktivitäten per formaler Sprache zulässt. In diesem Punkt wird dem Referenzmodell der WFMC keine Rechnung getragen, da der Autor die Entwicklung und Implementierung einer Komponente, welche eine formale Sprache zur Prozessdefinition interpretieren und verarbeiten kann, innerhalb des Projektes IPOS als zu aufwändig betrachtete.

6.7.2 Interaktion mit anderen Workflow Enactment Services

Die Interaktion zwischen dem System IPOS und anderen WFMS kann durch die Komponente der ExterneAPI erfolgen. Externe Workflow Enactment Services können durch dieses API auf bestimmte Dienste des Systems IPOS zugreifen. Dieser Zugriff ist reglementiert und nur für vorher festgelegte WFMS möglich.

Der direkte Zugriff auf interne Komponenten, wie z.B. die Workflow Engine, ist nicht möglich.

6.7.3 Interaktion mit anderen Applikationen und dem Benutzer

Die Interaktion des Systems IPOS mit externen Applikation wird ebenfalls über die ExterneAPI geregelt. Neben dieser normalen Interaktion existiert nur eine weitere: die Interaktion über die Stationen. Da es für jede Station eine Komponente gibt, die jegliche Dienste, die für eine Station nötig sind, bereit hält und alle erforderlichen Daten kennt, kann eine Station auch externe Dienste von anderen Applikationen aufrufen und diese nutzen. So können z.B. externe Berechnungen, die sehr lange dauern, von einer Fremdapplikation durchgeführt werden.

Die Interaktion mit dem Benutzer des WFMS fällt ebenfalls unter diesen Punkt. Schließlich ist die grafische Benutzerschnittstelle im Endeffekt nichts Anderes, als eine andere Applikation. Sie ist berechtigt mittels ExterneAPI auf die Dienste des Systems zuzugreifen.

6.7.4 Administration und Monitoring

Die Administration des Systems IPOS, wie es die WFMC im entsprechenden Interface vorsieht, wurden und sollten in diesem System nicht realisiert werden. Grund sind die hierfür als zu hoch eingeschätzten Entwicklungszeiten.

Allerdings wurde eine hohe zeitliche Kapazität in die Entwicklung des Monitors und des Repository's investiert. Hierbei sind die dort enthaltenen Werte von hoher Wichtigkeit und großer Bedeutung. Gerade verschiedene Funktionen, wie die stetige Überwachung der Vorgänge in einer Bevorratung, sind für eine frühzeitige und schnelle Fehlerentdeckung relevant.

Die im Repository enthaltenen aggregierten Daten stehen ohne zeitliche Verzögerung sofort zur Verfügung und sind daher ebenfalls von großer Bedeutung. Die Daten geben einen sehr schnelle Überblick über das System, die enthaltenen Bevorratungen und die an den BV-Aktionen gebundenen Kapazitäten.

6.8 Zusammenfassung

Das Kapitel hat die Entwicklung einer verarbeitenden Schicht für ein neu zu entwickelndes WFMS beschrieben. Hierbei wurde zuerst auf die Referenzarchitektur der WFMC eingegangen und danach eine Soll-Prozess-Erstellung durchgeführt. Auf der Basis des modellierten Prozesses wurde eine Extraktion der Stationen vorgenommen.

Die Entwicklung der internen Komponenten wurde anhand der Referenzarchitektur der WFMC vorgenommen und durch die Nutzung von zwei Patterns noch weiter verfeinert. Beginnend bei der Konzipierung auf der Ebene von Komponenten wurde durch die Analyse der Beziehungen und Kardinalitäten zwischen den Komponenten ein erstes Typdiagramm erstellt. Dieses Diagramm wurde danach in ein Klassendiagramm überführt werden.

Eine Betrachtung der von der WFMC vorgegebenen Interfaces im Vergleich zu der Umsetzung im System IPOS schließt das Kapitel ab.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Diese Arbeit hat die Konzeption und die Entwicklung eines Workflow Management Systems zur Steuerung von End- und Zwischenbevorratungsaktionen in der Automobilindustrie gezeigt. Hierbei ist ein Vorschlag für eine Konzeption eines schlanken WFMS entstanden, der in einem konkreten Klassendiagramm endete.

Der Autor hat beginnend bei der Projektbeschreibung die einzelnen Problematiken untersucht, auf deren Grundlage das Projekt erst gegründet wurde und hat daraufhin verschiedene Anforderungen für ein späteres Softwareprodukt formuliert. Auf diesen Anforderungen basierend, wurde eine kurze Untersuchung zu den möglichen Lösungswegen für die gegebenen Probleme durchgeführt. Alle Lösungswege wurden hierbei auf Vor- und Nachteile untersucht. Da die freie Entscheidung zur Lösung der Probleme nie gegeben war, wurde die Variante der Entwicklung einer Individualsoftware weiter verfolgt.

Im weiteren Verlauf der Arbeit hat der Autor beginnend bei der Erläuterung der Referenzarchitektur der WFMC eine konzeptionellen Anwendungsschicht entwickelt. Dabei stand für ihn eine Entwicklung möglichst nahe an der Referenzarchitektur der WFMC im Vordergrund. Während der Entwicklung hat der Autor Gebrauch von Patterns gemacht, durch deren Hilfe er eine schnellere Lösung für die Konzeption der internen Komponenten finden konnte. Weiterhin wurden Komponenten in das Konzept mit eingeführt, die aus dem Kontext des Projektes entstanden sind.

Das so entstandene Konzept umfasst die wichtigsten Komponenten zur Steuerung von Workflows in einem Bevorratungsprozess, wie er in der DCAG durchgeführt wird. Das Konzept ist frei von anderen Klassen/Komponenten/Details, die für eine breitere Anwendung des späteren Produktes nötig wären. Es orientiert sich an der obersten Prämisse dieser Arbeit: der Konzeptionierung und Entwicklung eines WFMS für End- und Zwischenbevorratungen in der Automobilindustrie.

7.2 Ausblick

Das vorliegende Konzept für das WFMS ist zwar ein Konzept zur Steuerung von Workflows in einem komplexen System, es ist aber nicht die Grundlage für ein WFMS, wie es definiert wurde. Hierzu fehlt eine Komponente, welche fähig ist eine formale Sprache zur Abbildung von Workflows zu verarbeiten und im System umzusetzen. Da es im Projekt IPOS nie eine solche Anforderung gab und eine Umsetzung einer solchen Komponenten nie angedacht war, hat der Autor entschieden diese Komponente nicht mit in das System aufzunehmen und

zu ignorieren. Gleichwohl sollte eine Komponente dieser Art noch entwickelt werden, wenn man das System IPOS in einem weiteren Schritt zu einem vollständigen und der Definition entsprechenden WFMS weiterentwickeln möchte.

Literaturverzeichnis

- [Alex77] Christopher Alexander, Sara Ishikawa, und Murray Silverstein: *A Pattern Language*, Band 2 in *Center for Environmental Structure Series*“, Oxford University Press, New York, NY, 1977
- [Du05] Reiner Dumke, *Skript zur Vorlesung Software Engineering*, Universität Magdeburg 2005
- [Fer00] Fernandez, Yuan, Brey: „*Analyse Patterns for the Order and Shipment of a Product*“, PloP 2000 Conference
- [Fow97] Fowler, M.: „*Analyse Patterns*“, Reading, MA. Addison-Wesley, 1997
- [Gaschler] Martin Gaschler, DaimlerChrysler AG
- [GIG03] GIGA Information Group: <http://www.gigaweb.com>, Abruf am 12.10.2005
- [Glo00] Glowka, Löckle: „*ARIS-Konventionenhandbuch - Anleitung zur Modellierung und Pflege der PMS PPA-Prozesse mit dem ARIS-Toolset 4.11*“, Untertürkheim 2000
- [HE03] Dissertation Klaudia Hergula: „*Daten- und Funktionsintegration durch Föderierte Datenbanksysteme*“, Technische Universität Karlsruhe, veröffentlicht 2003
- [IPOS05] Kickoff-Meeting, Sammlung von ersten Anforderungen durch Teamleiter für das Projekt IPOS
- [ISO03] International Standard Organisation: „*DIN EN ISO 9000:2000-12 Qualitätsmanagement – Grundlagen und Begriffe (ISO 9000:2000)*“, 2003
- [Kast04] Kastewski, C.: „*Untersuchung und Verbesserung der Disponententätigkeiten im Rahmen des BPR-Projektes ‚Compress*“, Diplomarbeit, Germersheim 2004
- [Ke98] Keller, Teufel: „*SAP R/3 prozeßorientiert anwenden*“, Bonn 1998, S. 156
- [Ku01] Kunze, A.: Powerpointpräsentation: „*Kurzvorstellung GSP LS*“, DaimlerChrysler AG, 2004
- [Korn05] Kornberger, Kastner: „*Operative Überprüfung der Endbevorratungsmenge*“, Dokumentennummer: W002GC95, Germersheim 2005
- [Lip04] Lipinski, J.: „*Ableitung von Lebenszyklusmodellen für ausgewählte Ersatzteilsortimente in der Automobilindustrie*“, Diplomarbeit, Germersheim 2004
- [Mau04] Maunz, A.: Projektantrag, 2004
- [Mau05] Maunz, A.: Präsentation Steuerkreis, 2005
- [Mau06] Maunz, A.: Präsentation Steuerkreis, 2006

- [Mes97] Gerard Meszaros, Kyles Brown: „*A Pattern Language for Workflow Systems*“, PLoP '97 Conference
- [MSKW96] J.A. Miller, A.P. Sheth, K.J. Kochut und X. Wang: „*CORBA-based Run-Time Architectures for Workflow Management Systems*“, 1996
- [Pro01] Winkler, L., *Skript zur Vorlesung Softwarelebenszyklus*, Hochschule Mittweida 2004
- [Roh95] Rohloff, M.: „*Integrierte Informationssysteme durch Modellierung von Geschäftsprozessen*“, Heidelberg 1995
- [Sch91] Scheer, A. W.: „*Architektur Integrierter Informationssysteme*“, Berlin 1991
- [Sch96] Schmidt et al.: „*Software Patterns*“, Commun. ACM, October 1996, 36-39
- [Seu00] Seung Il Lee, Dongsoo Han und Dongman Lee: „*A Pattern for Managing Distributed Workflows*“, PloP 2000 conference
- [WfMC96] Workflow Management Coalition: „*Workflow Management Coalition Terminology & Glossary*“, Document Number WFMC-TC-1011; 1996
- [WfMC98] Workflow Management Coalition: „*Workflow Management Application Programming Interface (Interface 2&3) Specification*“, Document Number WFMC-TC-1009, 1998
- [WfMC00] Workflow Management Coalition: „Standards“, <http://www.wfmc.org/standards/standards.htm>, Abruf am 20.02.2006

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, den 6. April 2006