



**Automatisierbarkeit der Suche von Referenzmodellen zu gegebenen  
Unternehmensmodellen**

**Diplomarbeit**

Arbeitsgruppe Wirtschaftsinformatik  
Institut für technische und betriebliche Informationssysteme, Fakultät für Informatik

Themensteller: Dr.-Ing. Gamal Kassem  
Betreuer: Dr.-Ing. Gamal Kassem

Erstgutachter: Prof. Dr. rer. pol. habil. Hans-Knud Arndt  
Zweitgutachter: Dr.-Ing. Gamal Kassem

vorgelegt von: Marcel Kempka

Abgabetermin: 28. Juli 2009

## **Danksagung**

Für die stetige Unterstützung bei der Erstellung der vorliegenden Arbeit und das Verständnis für die vielen verschobenen Freizeitaktivitäten möchte ich mich bei meiner Freundin Anke Kohnert besonders bedanken.

Weiterhin danke ich meinem Betreuer Gamal Kassem für die anregenden Diskussionen. Auch danke ich den Kollegen der Arbeitsgruppe für Wirtschaftsinformatik I, welche immer ein offenes Ohr für die typischen Fragen eines Diplomanden hatten.

Außerdem danke ich meiner Schwester Franziska, meinen Eltern Marita und Detlef, Großeltern Marianne und Gerhard Kempka und Uta und Karl-Heinz Kohnert für die finanzielle und nicht zu unterschätzende moralische Unterstützung!

Ein Dank gebührt auch meinen Freunden und Studienkollegen, welche mir immer mit wohlwollenden Worten zur Seite standen und dazu beitrugen die Zahl der Errata möglichst klein zu halten.

## Inhaltsverzeichnis

Inhaltsverzeichnis .....	III
Verzeichnis der Abkürzungen und Akronyme .....	V
Symbolverzeichnis .....	VI
Abbildungsverzeichnis .....	VII
Tabellenverzeichnis .....	VIII
1 Themeneinführung und Überblick über die Arbeit .....	1
1.1 Historie und Nutzen von ERP-Systemen.....	1
1.2 Problemstellung und Motivation.....	2
1.3 Zielsetzung der Arbeit .....	4
1.4 Aufbau der Arbeit und Einordnung in die Wirtschaftsinformatik .....	5
2 Grundlagen und Methoden des ERP-Customizing .....	7
2.1 Termini des ERP-Customizing .....	7
2.1.1 ERP-System .....	7
2.1.2 Customizing .....	9
2.1.3 Referenzmodell .....	12
2.1.4 Adaptives System.....	13
2.2 Aktuelle Customizing-Methoden als Bestandteil der ERP-Einführung .....	14
2.2.1 Allgemeines Vorgehensmodell nach Hansmann und Neumann.....	14
2.2.2 Standardvorgehensmodell der SAP AG – ASAP.....	15
2.2.3 Integriertes Vorgehen mit SAP NetWeaver und ARIS .....	17
2.3 Meta-Modelle für die Geschäftsprozessmodellierung .....	21
2.3.1 EPK.....	21
2.3.2 Petri-Netze und Erweiterungen .....	23
2.3.3 Workflow-Netze.....	27
3 Konzept und Voraussetzungen für die Automatisierung des Customizing .....	30
3.1 Konzept des Self Adaptive Customizing .....	30
3.1.1 Ziel und Gegenstand des Self Adaptive Customizing .....	30
3.1.2 Initiales Customizing .....	31
3.1.3 Kontinuierliches Customizing.....	33
3.1.4 Self Adaptive Customizing Middleware System .....	35
3.2 Anforderungen und Auswahl der Modellierungssprache .....	35
3.2.1 Anforderungen an die Modellierungssprache .....	36
3.2.2 Auswahl der Modellierungssprache .....	36
3.3 Anforderungen an das ERP-System.....	38
4 Automatisierte Suche von Referenzmodellen.....	40
4.1 Existierende Methoden für den Vergleich von Workflow-Netzen .....	40
4.2 Allgemeine Beschreibung der Suchmethoden.....	42
4.3 Kriterien für die Suche der Modelle.....	45

4.4	Fallbasierte Referenzmodellsuche .....	46
4.4.1	Metrik Fallgleichheit .....	46
4.4.2	Vergleich der potenziellen Fälle .....	49
4.4.3	Exemplarische Anwendung .....	49
4.4.4	Bewertung der Fallbasierten Referenzmodellsuche .....	51
4.5	Abstandsmethode .....	52
4.5.1	Metriken Funktionsabstand und Funktionsdifferenz.....	53
4.5.2	Metrik Inzidenzabstand .....	53
4.5.3	Referenzmodellsuche mittels Abstandsmethode.....	55
4.5.4	Exemplarische Anwendung .....	57
4.5.5	Bewertung der Abstandsmethode.....	59
5	Fazit und Ausblick.....	61
A	Erreichbarkeitsanalyse der Referenzmodelle.....	64
B	Inzidenzmatrizen .....	67
	Literaturverzeichnis .....	69

## **Verzeichnis der Abkürzungen und Akronyme**

ACID	Atomicity, Consistency, Isolation, Durability
AS	Anwendungssysteme
ASAP	Accelerated SAP
CIO	Chief Information Officer
EPK	Ereignisgesteuerte Prozesskette
ERP	Enterprise Resource Planning
IMG	Implementation Guide
IT	Informationstechnologie
MRP	Material Requirements Planning
MRP II	Material Resource Planning
o. V.	ohne Verfasser
SAC	Self Adaptive Customizing
SACMS	Self Adaptive Customizing Middleware System
XI	Exchange Infrastructure

## Symbolverzeichnis

$A$	Abstandsmatrix
$C$	Inzidenzmatrix
$\delta_F$	Dummyvariable für eine Funktion
$\delta_s$	Dummyvariable für eine Stelle
$F$	Flussrelation
$F^{diff}$	Funktionsdifferenz
$I$	Inzidenzabstand
$i$	Indexvariable
$j$	Indexvariable
$K$	Kapazität der Stellen
$M_0$	Anfangsmarkierung
$n$	Anzahl der Elemente einer Menge
$R^I$	für Implementierung gewähltes Referenzmodell
$R^-$	Funktionsabstand
$R^+$	zusätzliche Funktionen des Referenzmodells
$S$	Menge der Stellen
$s_i$	i-te Stelle
$T$	Menge der Transitionen
$t_i$	i-te Transition
$V^U$	Menge Fälle des Unternehmensmodells
$V^{Ri}$	Menge der Fälle des Referenzmodells i
$v$	Anzahl gleicher Fälle zweier Workflow-Netze
$W$	Kantengewichte
$w_j$	gültige Schaltfolge j eines Modells
$w^v$	vollständige Schaltfolge
$w_{i,j}^v$	vollständige Schaltfolge j des Referenzmodells i
$w_{U,j}^v$	vollständige Schaltfolge j des Unternehmensmodells

## Abbildungsverzeichnis

<b>Abb. 1.1:</b> Von realen Abläufen zur konfigurierten Software.....	3
<b>Abb. 1.2:</b> Ziel der Arbeit.....	5
<b>Abb. 2.1:</b> Morphologischer Kasten des Customizing.....	11
<b>Abb. 2.2:</b> Durchgängige Prozessarchitektur .....	19
<b>Abb. 2.3:</b> Sprachelemente der EPK.....	22
<b>Abb. 2.4:</b> Sprachelemente der Petri-Netze.....	24
<b>Abb. 2.5:</b> Aufgabenarten und Prozesse in Workflow-Netzen.....	28
<b>Abb. 2.6:</b> Operatoren in Workflow-Netzen .....	28
<b>Abb. 3.1:</b> Phasen des Initialen Customizing .....	32
<b>Abb. 3.2:</b> Zyklus Kontinuierliches Customizing.....	34
<b>Abb. 3.3:</b> Systemlandschaft Initiales Customizing .....	35
<b>Abb. 4.1:</b> Unternehmensmodell .....	43
<b>Abb. 4.2:</b> Referenzmodell R1 mit zusätzlicher Funktion .....	44
<b>Abb. 4.3:</b> Referenzmodell R2 mit alternativer Funktion.....	44
<b>Abb. 4.4:</b> Referenzmodell R3 mit zusätzlicher paralleler Funktion.....	44
<b>Abb. 4.5:</b> Referenzmodell R4 mit ersetzter Funktion.....	44
<b>Abb. 4.6:</b> Unternehmensmodell mit Hervorhebung der Differenzen.....	56
<b>Abb. 4.7:</b> Unternehmensmodell mit Differenzen zu Referenzmodell R4 .....	59
<b>Abb. 4.8:</b> Unternehmensmodell mit Differenzen zu Referenzmodell R2 .....	59

**Tabellenverzeichnis**

<b>Tab. 4.1:</b> Erreichbarkeitsanalyse Unternehmensmodell.....	47
<b>Tab. A.1:</b> Erreichbarkeitsanalyse Referenzmodell R1 .....	64
<b>Tab. A.2:</b> Erreichbarkeitsanalyse Referenzmodell R2 .....	65
<b>Tab. A.3:</b> Erreichbarkeitsanalyse Referenzmodell R3 .....	66
<b>Tab. A.4:</b> Erreichbarkeitsanalyse Referenzmodell R4 .....	67

# 1 Themeneinführung und Überblick über die Arbeit

## 1.1 Historie und Nutzen von ERP-Systemen

*„Enterprise Resource Planning Systeme (ERP-Systeme) bilden heutzutage in vielen Unternehmen das Rückgrat der Informationsverarbeitung, unabhängig von Branche oder Größe.“*

*Norbert Gronau<sup>1</sup>*

In Unternehmen werden betriebliche Anwendungssysteme eingesetzt, um die Geschäftsprozesse auf operativer, taktischer und strategischer Ebene zu unterstützen. Gegen Ende der sechziger Jahre diente betriebliche Standardsoftware hauptsächlich dem internen und externen Rechnungswesen. Begünstigende Faktoren wie detaillierte Gesetzesvorgaben haben die Entwicklung von Standards in diesem Bereich gefördert. Auch die Verfahren der produzierenden Industrie wurden dank der Arbeiten auf den Gebieten des Material Requirements Planning (MRP) und des Manufacturing Resource Planning (MRP II) standardisiert und resultierten ebenfalls in betrieblicher Standardsoftware.<sup>2</sup> Im Laufe der Zeit wuchsen die Anwendungsbereiche der standardisierten Softwaresysteme und es wurden immer mehr Aufgabenbereiche abgedeckt. In Folge der Integration der betriebswirtschaftlichen Aufgaben in *einem* Anwendungssystem etablierten sich in den neunziger Jahren die Enterprise Resource Planning Systeme (vgl. Gadatsch (2008), S. 324 f. und Rautenstrauch/Schulze (2003), S. 313 f.).

Die bereichs- und ebenenübergreifenden ERP-Systeme<sup>3</sup> haben den Anspruch, sämtliche Abläufe innerhalb eines Unternehmens abzubilden und dadurch die betriebliche Informationsfunktion effektiv und effizient zu unterstützen. So ermöglichen ERP-Systeme beispielsweise die Abbildung durchgehender Geschäftsprozesse von der Auftragsannahme über die Disposition, Produktion, Kommissionierung und Auslieferung bis hin zur Faktura. Durch die integrierten Anwendungssysteme wird versucht, das wirtschaftliche Gut ‚Information‘ auf möglichst ökonomische Weise bereitzustellen.

Ausgehend von monolithischen Informationssystemen über die Trennung von Daten und Funktionen bis hin zu serviceorientierten Architekturen wurde auf der technischen Seite der ERP-Systeme bereits viel Entwicklungsarbeit geleistet.<sup>4</sup> Die fachliche Seite

---

<sup>1</sup> Vgl. Gronau (2009).

<sup>2</sup> MRP und MRP II sind Sukzessivplanungskonzepte für die Produktionsplanung und –steuerung (vgl. Günther/Tempelmeier (2005), S. 305 ff).

<sup>3</sup> Auf den Begriff ERP-System wird im Abschnitt 2.1.1 näher eingegangen.

<sup>4</sup> Die SAP AG setzt beispielsweise auf eine Enterprise Services Architecture (vgl. Woods (2003), S. 2).

wurde ebenfalls berücksichtigt und es hat sich ein eigenes Fachgebiet entwickelt, welches sich mit dem Geschäftsprozessmanagement befasst. Während in früheren Jahren der Fokus auf der Integration von Daten, Funktionen und Prozessen lag, ist jetzt die Flexibilität der Software mehr und mehr in den Vordergrund gerückt. Ziel dieser Entwicklungen ist es, Anwendungssoftware so zu gestalten, dass diese schnell an neue Umgebungen, Anforderungen und Prozesse angepasst werden kann.

Heutzutage sind ERP-Systeme aus mittleren und großen Unternehmen nicht mehr wegzudenken. Der Chief Information Officer (CIO) der Allianz Deutschland AG<sup>5</sup> bezeichnet die Anwendungssysteme gar als Gedächtnis und Nervenbahnen der Aktiengesellschaft (vgl. Theurer (2007), S. 13). Nicht zuletzt ist es erst durch integrierte ERP-Systeme möglich geworden die immer komplexer und komplizierter werdenden Produkte und Dienstleistungen der Unternehmen in ökonomischer Weise zu verwalten.<sup>6</sup>

## 1.2 Problemstellung und Motivation

### *Problemstellung*

Eine Einführung der für die Unternehmen wichtigen ERP-Software ist mit großem Aufwand verbunden. Im Rahmen des Geschäftsprozessmanagements<sup>7</sup> werden, abgeleitet aus der Unternehmensstrategie und den strategischen Zielen, die Aufbau- und Ablaufstruktur eines Unternehmens definiert. Neben dem organisatorischen Aufbau (Geschäftsbereiche, Sparten, Organisationseinheiten etc.) wird dabei auch die Prozesslandschaft konzipiert. Wie in Abb. 1.1 zu sehen, wird durch Dokumentation der realen Strukturen und Abläufe das Unternehmensmodell erstellt. Um die Prozessabläufe auf operativer Ebene durch ERP-Software unterstützen zu können, muss nun das Modell der Unternehmung möglichst struktur- und verhaltensgetreu in der ERP-Software abgebildet werden. Die Übertragung des Unternehmensmodells in die ERP-Software ist daher ein notwendiger Bestandteil bei der Einführung von ERP-Software (Patig (2003), S. 1).<sup>8</sup>

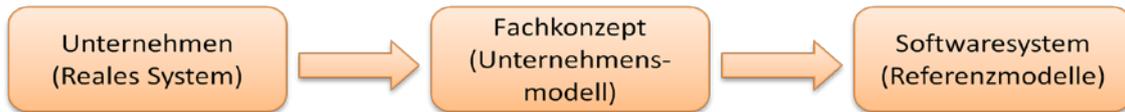
---

<sup>5</sup> Dr. Ralf Schneider ist seit 1. Januar 2006 CIO der Allianz Deutschland AG und somit für die gesamte Informationstechnik der Allianz Deutschland AG verantwortlich.

<sup>6</sup> Es sei hier exemplarisch auf die Vielzahl von Rabatten und Konditionen bei einer KFZ-Versicherung hingewiesen. Allein der Aufwand für eine manuelle Berechnung der Versicherungsprämie würde es unmöglich machen in dieser Geschäftssparte ohne Einsatz von Anwendungssystemen Gewinne zu erzielen. Siehe hierzu ergänzend auch Müller (2008), S. 508, der feststellt, dass über 70 Prozent der Geschäftsprozesse in deutschen Unternehmen elektronisch unterstützt werden.

<sup>7</sup> Auf eine ausführliche Erläuterung des Geschäftsprozessmanagements wird hier verzichtet. Siehe dazu Gadatsch (2008), S. 1 ff. oder auch Weske (2007), S. 5.

<sup>8</sup> Zu den Einführungsmethoden siehe Unterkapitel 2.2.



**Abb. 1.1:** Von realen Abläufen zur konfigurierten Software

Die Übertragung des Unternehmensmodells erfolgt durch Anpassen der ERP-Software an die Anforderungen des Unternehmens. Dieser Anpassungsprozess, in der Fachsprache Customizing<sup>9</sup> genannt, ist jedoch mit großem Aufwand verbunden. Das System SAP R/3 besitzt beispielsweise mehrere hundert Parameter, die eingestellt werden können (vgl. Rautenstrauch/Schulze (2003), S. 315). Probleme bei der Anpassung von ERP-Systemen werden nur selten offen gelegt. Dennoch kommt es vor, dass sowohl der Zeit- als auch der Kostenrahmen für das Customizing um ein erhebliches Maß überschritten werden (vgl. Niedereichholz/Reske (2005), S. 80 ff.; Wenzel (2001), S. 249). Um das Customizing zu erleichtern, werden oftmals Referenzmodelle<sup>10</sup> eingesetzt. Aber auch die Auswahl eines passenden Referenzmodells ist mit Aufwand verbunden.

### *Motivation*

In Anbetracht der oben genannten Situation stellt sich die Frage, ob der zeitliche und finanzielle Aufwand für das Customizing von ERP-Systemen reduziert werden kann. Durch eine verkürzte Einführungszeit würde neben den Kosten für die Inbetriebnahme auch die Zeit von Einführungsbeginn bis zur produktiven Nutzung des Systems erheblich reduziert werden. Auch nach der Einführung muss das ERP-System stetig an geänderte Anforderungen des Unternehmens angepasst werden. Dass in diesem Bereich Handlungsbedarf besteht, beschrieben u. a. SCHEUCH ET. AL. bereits 2004, denn oftmals kann die Informationstechnologie (IT) den geänderten Geschäftsprozessen nicht gerecht werden (vgl. Scheuch et. al. (2004), S. 61 f.). Andererseits ist die Überführung der Prozessveränderungen in die IT des Unternehmens oft schwierig, langwierig und kostspielig (vgl. Volmering/Scholz (2004), S. 38).

Aus Perspektive der Unternehmen ist es daher wünschenswert ERP-Systeme so zu gestalten, dass diese sich selbst den Unternehmensanforderungen anpassen. Genau diese Vision greift KASSEM mit dem Konzept des Self Adaptive Customizing (SAC) auf. Das Initiale Customizing unterteilt KASSEM in die Phasen Konvertierung, Evaluierung und Implementierung (vgl. Kassem (2007), S. 65). In der Konvertierungsphase werden das Unternehmensmodell und die Referenzmodelle der ERP-Software in Modelle mit einer gemeinsamen Modellierungssprache und gemeinsamen Fachtermini umgewandelt.

<sup>9</sup> Für eine ausführlichere Definition des Begriffes Customizing wird auf Kapitel 2.1.2 verwiesen.

<sup>10</sup> Eine nähere Erläuterung zum Begriff Referenzmodell befindet sich im Abschnitt 2.1.3.

Während der Evaluierungsphase wird dann zu dem gegebenen Unternehmensmodell das am besten geeignete Referenzmodell des ERP-Systems ermittelt. Die Automatisierung dieser Phase kann dazu beitragen, die Komplexität der Referenzmodellauswahl zu beherrschen. Denn ERP Systeme besitzen eine große Anzahl von Referenzmodellen. Das System R/3 der SAP AG enthält beispielsweise mehr als 800 Referenzmodelle (vgl. Hansmann/Neumann (2005), S. 331). Liegt ein geeignetes Referenzmodell vor, kann anschließend die ERP-Software gemäß den Referenzvorgaben angepasst werden. Eine ausführlichere Darstellung des SAC folgt im Unterkapitel 3.1.

### **1.3 Zielsetzung der Arbeit**

Aus den oben genannten Gründen wird untersucht, ob das Customizing von ERP-Software im Rahmen des Self Adaptive Customizing beschleunigt werden kann. Da dies jedoch ein umfangreicher Bereich ist, wird hier das Hauptaugenmerk auf die Evaluierungsphase des Initialen Customizing innerhalb des Self Adaptive Customizing gelegt.

In der vorliegenden Arbeit wird untersucht, ob es möglich ist automatisiert ein Referenzprozessmodell zu finden (siehe Abb. 1.2). Dabei ist davon auszugehen, dass ein Unternehmensmodell existiert, welches die abzubildenden Abläufe und Strukturen vorgibt. Ist dieses Referenzmodell gefunden, kann das ERP-System in der Implementierungsphase auf Basis der Referenzprozesse und den Parametern im Unternehmensmodell automatisiert angepasst werden. Darüber hinaus wird noch ein zweites Ziel verfolgt. Es ist anzunehmen, dass in den meisten Fällen für das Unternehmensmodell kein vollständiges Pendant in den Referenzprozessen gefunden werden kann. Für Prozesse des Unternehmensmodells, für welche kein direktes Abbild in der Menge der Referenzprozesse auffindbar ist, soll daher der Referenzprozess mit der geringsten Abweichung zum Unternehmensprozess gewählt und die Differenzen anschließend hervorgehoben werden.

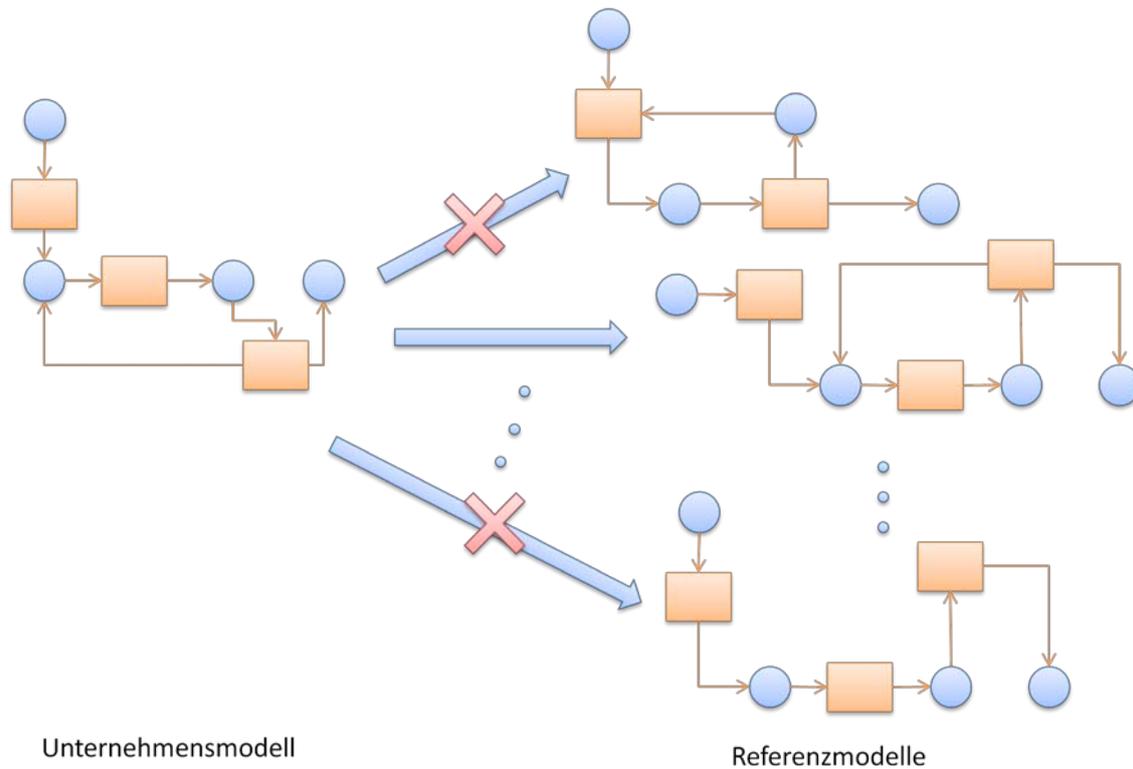


Abb. 1.2: Ziel der Arbeit

Zur Abgrenzung des Arbeitsbereichs wird angemerkt, dass es sich hier um eine theoretische Arbeit handelt, wenngleich die vorgestellten Methoden in der betriebswirtschaftlichen Praxis eingesetzt werden sollen.

#### 1.4 Aufbau der Arbeit und Einordnung in die Wirtschaftsinformatik

Um das aufgezeigte Ziel erreichen zu können, werden im *zweiten* Kapitel zunächst die grundlegenden Begriffe erläutert bzw. benötigte Konzepte definiert. Dort werden ebenfalls Methoden zur Einführung von ERP-Software vorgestellt und der aktuelle Stand bezüglich der Beschleunigung des Customizing kurz skizziert. Für die Modellierung der Prozesse werden in dieser Arbeit Workflow-Netze verwendet und daher am Ende des zweiten Kapitels deren Grundlagen vorgestellt.

Im *dritten* Kapitel wird zunächst das Konzept des Self Adaptive Customizing vorgestellt, bevor die für die Realisierung des Self Adaptive Customizing notwendigen Anforderungen erläutert werden. Unter anderem werden dort die Anforderungen an die Modellierungssprache und die ERP-Systeme begründet.

Im *vierten* Kapitel werden dann zwei Methoden entwickelt, welche eine Automatisierung der Referenzmodellssuche ermöglichen. Neben der detaillierten Be-

schreibung der Methoden wird auch deren Anwendung am Beispiel von einem Unternehmensmodell und vier Referenzmodellen erläutert.

Die Ergebnisse der Arbeit werden im *fünften* Kapitel abschließend zusammengefasst und kritisch hinterfragt. So wird im Resümee geprüft, ob die angestrebte Automatisierung des Customizing mit den hier vorgestellten Methoden realisierbar ist. Darüber hinaus werden potenzielle Ansatzpunkte für eine weitere Verfolgung des Self Adaptive Customizing aufgezeigt.

Die *Einordnung* der Arbeit in den Bereich der Wirtschaftsinformatik lässt sich wie folgt begründen. Nach MERTENS beschäftigen sich Wirtschaftsinformatiker mit „[...] der Konzeption, Entwicklung, Einführung, Wartung und Nutzung von AS [...]“, (Mertens et. al. (2005), S. 1). Anwendungssysteme<sup>11</sup> (AS) i. e. Sinn werden in die Arten Individualsoftware und Standardsoftware unterteilt (vgl. Rautenstrauch/Schulze (2003), S. 281). Wird Standardsoftware eingeführt, so muss diese in der Regel an die Unternehmensanforderungen angepasst werden. Während dieses Anpassungsvorgangs werden die Unternehmensprozesse mit den Referenzprozessen der ERP-Software abgeglichen.<sup>12</sup> Folglich ist die Suche nach geeigneten Referenzprozessen zu gegebenen Unternehmensprozessen für das Customizing von Standardsoftware eine Teilmenge des Gegenstands der Wirtschaftsinformatik.

---

<sup>11</sup> Anwendungssysteme setzen sich aus Hardware, Systemsoftware, Kommunikationseinrichtungen und der Anwendungssoftware zusammen. Im engeren Sinne ist mit dem Begriff Anwendungssystem lediglich die Anwendungssoftware gemeint (vgl. o. V. (2007), S. 319).

<sup>12</sup> Für eine ausführlichere Erläuterung der Einführungsmethoden siehe Unterkapitel 2.2.

## 2 Grundlagen und Methoden des ERP-Customizing

In diesem Kapitel werden zunächst die essenziellen Begriffe und Methoden dieser Arbeit erläutert. Anschließend werden die formalen Workflow-Netze eingeführt. Dabei wird gezeigt, auf welche Art und Weise Workflow-Netze für die Modellierung von Geschäftsprozessen eingesetzt werden können.

### 2.1 Termini des ERP-Customizing

Zu Beginn des Unterkapitels wird der Begriff ERP-System erläutert, da dieser den Hauptgegenstand der Arbeit bezeichnet. Sind Eigenschaften und Zweck eines ERP-Systems bekannt, wird auf dessen Customizing eingegangen, bevor die beim Customizing zum Einsatz kommenden Referenzmodelle beleuchtet werden. Letztendlich wird in diesem Unterkapitel der Terminus adaptives System genauer betrachtet, da er ein zentraler Bestandteil des weiter unten beschriebenen Self Adaptive Customizing ist.

#### 2.1.1 ERP-System

Ein Enterprise Resource Planning System ist ein betriebliches Anwendungssystem, welches auf Standards basiert. Es unterstützt die operative Ausführung von Aufgaben mehrerer betriebswirtschaftlicher Funktionsbereiche. Die im ERP-System abgebildeten Daten und Funktionen sind miteinander integriert, sodass sämtliche innerbetrieblichen Prozesse<sup>13</sup> eines Unternehmens abgedeckt werden können. Um einen branchenübergreifenden Einsatz zu ermöglichen, kann ein ERP-System an Branchen- und Unternehmensanforderungen angepasst werden (vgl. Rautenstrauch/Schulze (2003), S. 314; Heinrich et. al. (2004), S. 237 und 625 f.; Hannsmann/Neumann (2005), S. 329).<sup>14</sup>

Grundlage für *Standards* können gleichartige betriebswirtschaftliche Anforderungen oder gesetzliche Normen sein (vgl. Alpar et al. (2008), S. 185). Technische Standards werden hingegen oftmals von Konsortien<sup>15</sup> erarbeitet oder ergeben sich aus der weiten Verbreitung einer Software (proprietärer Standard). Beispiele aus dem betriebswirtschaftlichen Bereich sind Kontenrahmen für das Rechnungswesen, Anreizsysteme für das Controlling, Portfolioanalysen für das Marketing oder Planungs- und Steuerungssysteme für die Produktion (KANBAN). Gesetzliche Standards ergeben sich beispielsweise aus dem Versicherungsvertragsgesetz oder der Regulierung des

---

<sup>13</sup> Mit betrieblichen Prozessen sind hier sowohl Management-, Support- als auch Kernprozesse gemeint.

<sup>14</sup> Gadatsch benennt als weitere Eigenschaften ein einheitliches Entwicklungskonzept, die Realisierung als Schichtenarchitektur sowie die Transaktionsorientierung (vgl. Gadatsch (2008), S. 308 ff.).

<sup>15</sup> Als Beispiele seien hier das World Wide Web Consortium und die Object Management Group genannt.

Energiemarktes.<sup>16</sup> Durch Berücksichtigen von Standards zum Entwicklungszeitpunkt kann die Standardsoftware in mehreren Unternehmen ohne Modifikationen eingesetzt werden<sup>17</sup> und erlaubt die Zusammenarbeit mit anderen Anwendungssystemen innerhalb und außerhalb eines Unternehmens.

Typische betriebswirtschaftliche *Funktionsbereiche* von ERP-Systemen sind (vgl. Stahlknecht/Hasenkamp (2005), S. 327; Rautenstrauch/Schulze (2003), S. 314 f.):

- Beschaffung,
- Produktionsplanung und -steuerung,
- Vertrieb,
- Materialwirtschaft,
- Servicemanagement,
- Finanzwesen,
- Controlling,
- Personalwesen,
- Projektmanagement,
- Qualitätsmanagement,
- Instandhaltung und Anlagenwirtschaft.

Die *Datenintegration* wird in der Regel mittels Datenbanksystemen umgesetzt. Diese ermöglichen den Zugriff auf einen konsistenten Datenbestand aus verschiedenen Funktionsbereichen innerhalb eines Unternehmens.<sup>18</sup> Die *Funktions-* und *Prozessintegration* verknüpft Abläufe innerhalb und zwischen betriebswirtschaftlichen Funktionen miteinander. Im System SAP R/3 ist die Datenintegration beispielsweise

---

<sup>16</sup> Mit der Novelle vom Jahr 2005 des Energiewirtschaftsgesetzes wurde beschlossen, bei größeren Energieversorgern den Netzbereich von allen anderen Aktivitäten zu trennen (vgl. § 7 EnWG). Für die Standardsoftware der SIV AG wurde daraufhin beispielweise ein 2-Mandanten-Modell entwickelt, welches die getrennte Verwaltung von Netzbetreiber-gesellschaft und Versorgungsgesellschaft auf technologischer Ebene umsetzt.

<sup>17</sup> Voraussetzung ist, dass die Standards in den entsprechenden Unternehmen angewendet werden.

<sup>18</sup> Für weitere Vorteile der Datenintegration sei auf HEUER und SAAKE verwiesen und beispielsweise das ACID-Prinzip (Atomicity, Consistency, Isolation, Durability) sowie die neun Codd'schen Regeln genannt (vgl. Heuer/Saake (2000), S. 4 und 6 ff.).

mittels Stammdaten und die Prozessintegration mittels Belegen realisiert (vgl. Patig (2003), S. 1).

Viele ERP-Systemhersteller bieten *Erweiterungen* für die Integration von überbetrieblichen Prozessen an. Exemplarisch seien hier Module für das Customer Relationship Management, welche eine kundenindividuelle Betreuung ermöglichen, das Supplier Relationship Management, für die Verwaltung von Lieferanten, sowie das Supply Chain Management, zur Verwaltung der gesamten Lieferkette, genannt. Diese Erweiterungen sind jedoch gemäß obiger Definition kein Bestandteil eines ERP-Systems im engeren Sinn.

### 2.1.2 Customizing

Wird betriebliche Standardsoftware in einem Unternehmen eingeführt, muss diese an die Anforderungen des Unternehmens angepasst werden. Diese Anpassungen, werden Customizing genannt. Das Customizing beeinflusst die Stammdaten und Abläufe in den betriebswirtschaftlichen Funktionen (vgl. Patig (2003), S. 1) sowie die Auswahl der zu verwendenden Funktionen (vgl. Rautenstrauch/Schulze (2003) S. 315). Die SAP AG definiert Customizing wie folgt: „Mit Customizing passen Sie die unternehmensneutral und branchenspezifisch ausgelieferte Funktionalität den spezifischen betriebswirtschaftlichen Anforderungen Ihres Unternehmens an.“, (SAP (2009a)).<sup>19</sup> Im engeren Sinn wird unter dem Begriff Customizing die *Konfiguration* und *Parametrisierung* der ERP-Software verstanden. Im weiteren Sinn wird auch die *Modifikation* zum Customizing gezählt. Obwohl es sich hierbei weniger um ein Einstellen der Software als viel mehr um eine Umprogrammierung handelt (vgl. Stahlknecht/Hasenkamp (2005), S. 297 f.).

Die *Konfiguration* einer Software ist Customizing auf der höchsten Abstraktionsebene. Dabei werden lediglich die zu verwendenden Softwaremodule ausgewählt und bei Bedarf deren Schnittstellen aufeinander abgestimmt (vgl. Heinrich et. al. (2004) S. 373). Diese Form des Customizing bereitet den geringsten Aufwand. Die Konfiguration kann jedoch nur für Unternehmensbereiche angewandt werden, welche sich vollständig an dem Standard orientieren. Kleinere Abweichungen von der Standardkonfiguration können durch die Einstellung von *Parametern* abgebildet werden (vgl. Heinrich et. al. (2004), S. 487). Dabei werden die in der ERP-Software hinterlegten Standardwerte durch unternehmensspezifische Werte ersetzt. Genügt diese Form des Customizing ebenfalls nicht, um die ERP-Software an die Bedürfnisse des Unternehmens anzu-

---

<sup>19</sup> So müssen beispielsweise die im Unternehmen genutzte Währung oder die Organisationseinheiten eingestellt werden.

passen, kann die Software auch *modifiziert* werden. Dabei werden individuelle Funktionen programmiert und über Schnittstellen, welche von der ERP-Software bereitgestellt werden, in die Software integriert (vgl. Alpar et. al. (2008), S. 396). Um den Aufwand und die Kosten des Customizing möglichst gering zu halten wird versucht, so wenig Änderungen wie möglich vorzunehmen. Darüber hinaus gefährdet eine Modifikation der ERP-Software unter Umständen deren Releasefähigkeit<sup>20</sup> (vgl. Alpar et. al. (2008), S. 396).

*Voraussetzung* für das Customizing ist eine sorgfältige Analyse und Modellierung der Geschäftsprozesse des Unternehmens. Liegen die Prozessmodelle vor, können die Parameter der betrieblichen Standardsoftware im Rahmen des Customizing systematisch eingestellt werden (vgl. Rautenstrauch/Schulze (2003), S. 315). *Auslöser* für das Customizing eines ERP-Systems können neben einer Einführung eine Erweiterung oder ein Releasewechsel sein (vgl. SAP (2009a)).

Die für das Customizing *relevanten Objekte* einer ERP-Software sind Organisationseinheiten, Stammdaten, Funktionen und Prozesse (Abläufe) (vgl. Wenzel, P. (2001), S. 101 f.). Die Organisationseinheiten stellen die Aufbauorganisation des Unternehmens dar. Stammdaten sind alle Daten, die nur selten verändert werden (z. B. Produktbeschreibungen) (vgl. Mertens (2004), S. 21). Die Funktionen der ERP-Software setzen die operative Ausführung der betrieblichen Funktionen bzw. Aufgaben um. Einzustellende Funktionsparameter sind beispielsweise Grenzwerte, ab denen ein Vorgesetzter einem bestimmten Geschäftsvorfall zustimmen muss.<sup>21</sup> Zusammen mit der Einstellung der sachlogischen Reihenfolge bilden die Funktionen die Prozesse des Unternehmens.

### *Vorgehen*

Im Rahmen der ERP-Einführung wird durch (interne oder externe) Unternehmensberater ein Istmodell des Unternehmens erstellt, welches den aktuellen Zustand der Realität widerspiegelt. Aus diesem wird im Anschluss ein Sollmodell der betriebswirtschaftlichen Strukturen und Abläufe des Unternehmens entwickelt. Dieses Modell (Business Blueprint) dient als Vorlage für das Customizing der einzuführenden Software. Im Beispiel von SAP wird im nächsten Schritt (Realisation des ASAP) das

---

<sup>20</sup> Releasefähigkeit bezeichnet die Möglichkeit, Änderungen an der ERP-Software bei einem Wechsel auf eine neue Version (Release) automatisch zu übernehmen. Siehe hierzu auch Mertens et. al. (2005), S. 167 f.

<sup>21</sup> In der Versicherungsbranche ist es z. B. üblich, bei einem KFZ-Sachschaden ab einer bestimmten Schadenshöhe einen Gutachter zur Bewertung heranzuziehen. Bei kleineren Schäden erfolgt die Regulierung ohne Beteiligung eines Gutachters.

System mit Hilfe des Implementation Guide (IMG) bzw. Einführungsleitfadens angepasst. Dabei wird der Einführungsleitfaden von oben nach unten abgearbeitet. Jedes Blatt des Baumes, den der Einführungsleitfaden aufspannt, ist mit einer Customizing-Transaktion hinterlegt. Mit diesen können jeweils unternehmensspezifische, vom Standardprozess abweichende, Parameter definiert bzw. geändert werden. Die konkreten Werte für diese Anpassungen entnimmt der Berater dem Business Blueprint.

### *Ordnungsrahmen*

Nachdem der Begriff Customizing aus verschiedenen Perspektiven betrachtet wurde, werden an dieser Stelle die Eigenschaften zusammengefasst und in einem *morphologischen Kasten* veranschaulicht.

Merkmals	Ausprägung			
Auslöser	Einführung	Erweiterung	Releasewechsel	
Ebenen	fachliche Ebene		technische Ebene	
Adaptionshäufigkeit	initial (einmalig, statisch)		kontinuierlich (dynamisch)	
Automatisierungsgrad	nicht automatisiert	semi-automatisiert	automatisiert (ideal)	
Abstraktionsgrad	Konfiguration	Parametrisierung	Modifikation	
Customizing-Objekte	Organisations-einheiten	Stammdaten	Funktionen	Prozesse

**Abb. 2.1:** Morphologischer Kasten des Customizing

Die Ausprägungen der Adaptionshäufigkeit werden in Geihs ausführlicher erläutert (vgl. Geihs (2008), S. 133 ff.). Für eine Beschreibung der Merkmalsausprägungen Parametrisierung und Modifikation siehe auch Rautenstrauch/Schulze (2003) S. 298 ff.

Soll beispielsweise eine neue ERP-Software im Unternehmen genutzt werden, so handelt es sich um eine Einführung mit initialer Anpassung. Die anschließende ständige Anpassung der Software an geänderte Anforderungen (z. B. optimierte Prozessabläufe) wird als Kontinuierliches Customizing bezeichnet.

### 2.1.3 Referenzmodell

Beim Customizing von ERP-Systemen kommen häufig Referenzmodelle zum Einsatz. Da diese auch insbesondere für das Konzept des Self Adaptive Customizing relevant sind, werden deren Merkmale kurz erläutert.

#### *Definition Referenzmodell*

Referenzmodelle sind allgemeingültige Modelle und somit auf viele Unternehmen anwendbar. Entwickelt werden diese entweder von Grund auf oder durch Abstraktion von unternehmensspezifischen Modellen. Unternehmensmodelle hingegen sind genau auf ein Unternehmen zugeschnitten und bilden dessen Strukturen und Abläufe ab. Da Referenzmodelle sowohl als Idealmodelle als auch als Sollmodelle entworfen werden, müssen diese nicht zwingend in einer praktischen Anwendung umgesetzt sein (vgl. Rautenstrauch/Schulze (2003), S. 230). Als Idealmodelle werden Modelle bezeichnet, die frei von Umwelteinflüssen oder anderen Restriktionen die beste bzw. ideale Lösung für eine Diskurswelt<sup>22</sup> bereit stellen (vgl. Rosemann (1996), S. 33). In Sollmodellen wird der gewünschte zukünftige Zustand eines Systems erfasst. Sie sind als Gestaltungsmodelle die Grundlage für eine spätere Umsetzung (vgl. Rosemann (1996), S. 32).

Weiterhin enthalten Referenzmodelle Empfehlungen für die Gestaltung von Organisations- und Anwendungssystemen und werden daher hinsichtlich des Adressaten und Modellierungszweckes in Referenz-Organisationsmodelle und Referenz-Anwendungssystemmodelle unterschieden (vgl. Becker/Schütte (2004), S. 77).

Zu Gunsten einer besseren Lesbarkeit wird im Rest der Arbeit anstelle von Unternehmensprozessmodell der verkürzte Terminus Unternehmensmodell verwendet. Ebenso wird für die Bezeichnung Referenzprozessmodell die verkürzte Form Referenzmodell genutzt.

#### *Einsatzgebiete und Vorteile*

Anwendung finden Referenzmodelle bei der Dokumentation der Funktionalität einer ERP-Software, der Softwareauswahl, der Softwarekonfiguration und -anpassung (vgl. Rosemann et. al. (2005), S. 54 f.) sowie der Prozessidentifikation (vgl. Becker/Meise (2005), S. 124).

---

<sup>22</sup> Die Diskurswelt ist der im jeweiligen Modell betrachtete Ausschnitt aus der Realität (vgl. Rosemann (1996), S. 17 ff.).

Ein Vorteil von Referenzmodellen ist deren Robustheit gegenüber Umweltveränderungen, sodass diese über einen längeren Zeitraum ihre Gültigkeit behalten (vgl. Speck/Schnetgöke (2005), S. 217). Weiterhin reduzieren Referenzmodelle den Erstellungsaufwand für unternehmensindividuelle Modelle (vgl. Rautenstrauch/Schulze (2003), S. 230). BECKER und SCHÜTTE subsumieren den potenziellen Nutzen von Referenzmodellen unter den Begriffen Kostenminimierung, Erlösmaximierung und Risikosenkung. Für eine ausführlichere Beschreibung des Nutzens von Referenzmodellen siehe Becker/Schütte (2004), S. 80 ff.

#### 2.1.4 Adaptives System

Wie im einleitenden Kapitel erwähnt, ist die vorliegende Arbeit eng mit dem Konzept des Self Adaptive Customizing verknüpft. Daher erläutert der letzte Abschnitt der Begriffsdefinitionen den Terminus ‚Adaptives System‘.

Systeme, die in der Lage sind, sich an geänderte Umweltbedingungen anzupassen werden adaptive Systeme genannt (vgl. Alpar et. al. (2008), S. 18). Um die Veränderungen der Umwelt, in der Wirtschaftsinformatik Kontext genannt, zu erfassen, benötigen diese Systeme Kontextsensoren. Der Kontext eines Systems beschreibt die Umwelt und Umwelteinflüsse, welche auf das System wirken.

Bezüglich des *Adaptionszeitpunktes* werden Entwurfszeitadaptivität und Laufzeitadaptivität unterschieden. Erstere meint die Anpassung der Software vor Inbetriebnahme, welche auch statische Adaption genannt wird. Laufzeitadaptivität hingegen bezeichnet die Fähigkeit einer Software zur Anpassung während ihrer Ausführung (dynamische Adaption) (vgl. Geihs (2008), S. 133 f.). Die *Adaptivitätstiefe* wird unterteilt in parametrische und kompositionelle Adaption. Bei der parametrischen Adaption werden die Werte vorgegebener Parameter verändert. Dies resultiert in einer Verhaltensänderung der Software, wobei die Softwarestruktur erhalten bleibt. Kommt die kompositionelle Adaption zum Einsatz, werden einzelne Komponenten ausgetauscht oder verändert. Folglich werden Verhalten und Struktur des Anwendungssystems der Umwelt angepasst (vgl. Geihs (2008), S. 134 f.).

Adaptive Middleware, welche das Konzept des adaptiven Systems umsetzt, muss die Grundfunktionen *Kontextmanagement*, *Adaptionsmanagement* und *Konfigurationsmanagement* verwirklichen. Das Kontextmanagement überwacht den Kontext einer Anwendung und sendet signifikante Änderungen an das Adaptionsmanagement. Dieses bewertet alle Anpassungsmöglichkeiten nach vorgegebenen Kriterien und übermittelt

die ausgewählte Konfiguration an das Konfigurationsmanagement. Letzteres setzt dann die gewählte Anwendungskonfiguration um (vgl. Geihs (2008), S. 137).

Zur Ermöglichung des Self Adaptive Customizing dienen die Prozessmodelle des Unternehmens als Kontext. Für die kontinuierliche Anpassung an geänderte Anwenderanforderungen dienen darüber hinaus Analysen der Systemnutzung als Kontexteingaben.

## 2.2 Aktuelle Customizing-Methoden als Bestandteil der ERP-Einführung

Nachdem der Begriff des Customizing vorgestellt wurde, werden in den folgenden Abschnitten exemplarisch aktuelle Customizing-Methoden aufgezeigt. Dabei wird auch auf bereits existierende Ansätze zur (Teil-) Automatisierung des Customizing eingegangen. Die Ausführungen der folgenden Abschnitte beziehen sich teilweise auf das ERP-Anwendungssystem SAP R/3. Analoge Verfahren finden sich auch bei anderen ERP-Systemen, werden hier jedoch nicht aufgeführt.<sup>23</sup>

### 2.2.1 Allgemeines Vorgehensmodell nach Hansmann und Neumann

HANSMANN und NEUMANN haben mehrere Phasenmodelle zur Einführung von ERP-Software untersucht und ein abstrahiertes Gesamtmodell entwickelt. Dieses allgemeine Vorgehensmodell enthält die Phasen *Systemauswahl*, *Vorstudie*, *Istanalyse*, *Sollkonzept*, *Realisierung*, *Einführung* und *Betrieb* (vgl. Hansmann/Neumann (2005), S. 336 f.). Für das Customizing sind insbesondere die Phasen *Istanalyse*, *Sollkonzept* und *Realisierung* interessant.

Um die Umsetzung der Anpassungen der ERP-Software zu realisieren, haben HANSMANN und NEUMANN drei Herangehensweisen identifiziert. Nachdem die Istmodelle in der Istanalyse erfasst wurden, können die Sollmodelle auf *Basis des ERP-Referenzmodells* erstellt werden. Alternativ kann auch eine *Modellierung des Idealzustandes* durchgeführt werden oder das Sollmodell wird auf der *Basis der Istmodelle* entworfen (vgl. Hansmann/Neumann (2005), S. 361).

Vorteile der ersten Variante sind neben einer schnellen Modellierung und eine frühe Einarbeitung in die Termini der ERP-Software, die Sicherheit, dass die adaptierten Modelle durch die Software umgesetzt werden können. Als Nachteil muss jedoch angemerkt werden, dass unter anderem die Ist-Situation des Unternehmens nur bedingt

---

<sup>23</sup> Siehe hierzu beispielsweise die Oracle Unified Method (vgl. Oracle (2009)), Lawson StepWise (vgl. Lawson (2009)) oder die Infor OnePoint Implementation Methodology (vgl. Bürner (2007), S. 4 ff.).

berücksichtigt wird und daher die unternehmensspezifischen Anforderungen zum Teil vernachlässigt werden. Die entstehenden Differenzen könnten durch Änderungen der Unternehmensprozesse verringert werden, wodurch jedoch Wettbewerbsvorteile verloren gehen können. Bei der Erstellung von Idealmodellen, losgelöst von Referenzmodellen, stehen hingegen die unternehmensspezifischen Gegebenheiten im Vordergrund. Der Bezug zur ERP-Software wird anschließend durch den Abgleich mit den ERP-Referenzmodellen hergestellt. Hierfür muss die ERP-Software aber in hohem Maße anpassbar sein. Als dritte Variante wird ausgehend vom Istmodell der Unternehmung ein Abgleich mit den ERP-Referenzmodellen vorgeschlagen. Ein Vorteil dieses Ansatzes ist der geringere Aufwand gegenüber der idealmodellbezogenen Methode, da beispielsweise die Unternehmensorganisation nicht geändert werden muss (vgl. Hansmann/Neumann (2005), S. 361 f.).

## **2.2.2 Standardvorgehensmodell der SAP AG – ASAP**

### *ASAP allgemein*

ASAP ist ein Akronym für Accelerated SAP und bezeichnet die Standardvorgehensmethode der SAP AG. Die ASAP-Methode besteht aus den fünf Phasen *Projektvorbereitung*, *Business Blueprint*, *Realisierung*, *Produktionsvorbereitung* und *Go-Live und Support*. Weiterhin sind der Implementation Guide und der Implementation Assistant, ein Softwarewerkzeug zur Navigation durch die fünf Phasen, Bestandteil der ASAP-Methode (vgl. Hansmann/Neumann (2005), S. 337 f.). Bezüglich des allgemeinen Vorgehensmodells entfällt hierbei die Systemauswahl, da es sich um eine systemspezifische Methode handelt.

Der *Implementation Assistant* ist ein Softwarewerkzeug, welches die Navigation durch die fünf Phasen ermöglicht. Dieses Werkzeug enthält den Projektplan, eine Beschreibung aller Aktivitäten und eine ‚Knowledge Corner‘, in der Dokumentationen und Tipps und Tricks enthalten sind. Weiterhin verfügt der Implementation Assistant über eine ‚Question and Answer Database‘ mit einem Fragenkatalog zum R/3-Referenzmodell sowie eine ‚Issues-Datenbank‘, in der offene Fragen oder noch zu erledigende Aktivitäten dokumentiert werden (vgl. Hansmann/Neumann (2005), S. 338).

### *Projektvorbereitung*

Die Projektvorbereitungsphase enthält die Arbeitspakete ‚Projektplanung‘, ‚Projekt-abläufe‘, ‚Projekt-Kickoff‘, ‚Planung der technischen Anforderungen‘ und ‚Qualitäts-

prüfung und Projektvorbereitung‘ (vgl. Wenzel (2001), S. 250). In dieser Phase werden die Projektziele definiert und Vorgehensweisen zur effizienten Entscheidungsfindung vereinbart. Des Weiteren wird der Projektauftrag unterzeichnet, das Projektteam gebildet, der Projektstart ausgerufen und die Einführungsstrategie festgelegt (vgl. Wenzel (2001), S. 251). Mögliche Strategien sind ‚Big Bang‘, ‚Step by Step‘ oder ‚Pilotierung‘ (vgl. Hansmann/Neumann (2005), S. 333 f.).

#### *Business Blueprint (Fachkonzept)*

Während der Erstellung des Business Blueprint werden die Arbeitspakete ‚Projektmanagement Business Blueprint‘, ‚Schulung des Projektteams Business Blueprint‘, ‚Systemumgebung entwickeln‘, ‚Organisationsstruktur‘, ‚Geschäftsprozessdefinition‘ und ‚Qualitätsprüfung Business Blueprint‘ abgearbeitet. In dieser Phase werden die Geschäftsziele des Unternehmens geklärt und die davon abhängigen betriebswirtschaftlichen Anforderungen und Prozesse des Unternehmens bestimmt. Diese Anforderungen werden in einem detaillierten Konzept – dem Business Blueprint – dokumentiert (vgl. Wenzel (2001), S. 258 f.).

#### *Realisierung*

Nachdem die Prozesse des Unternehmens den begleitenden Beratern bekannt sind, wird das System an die betriebswirtschaftlichen Anforderungen in zwei Arbeitspaketen, der Baseline-Konfiguration und der Detail-Konfiguration angepasst. Für diese Anpassungen wird der vorliegende Business Blueprint genutzt. Neben Projektmanagement, Schulung und Qualitätsprüfung werden in dieser Phase die Arbeitspakete ‚Systemadministration‘, ‚Datenübernahmeprogramme entwickeln‘, ‚Schnittstellenprogramme für Anwendungen entwickeln‘, ‚Erweiterungen entwickeln‘, ‚Reports und Formulare entwickeln‘, ‚Berechtigungskonzept erarbeiten‘, ‚Archivierungskonzept einrichten‘, ‚abschließender Integrationstest‘ und Dokumentation und Schulungsunterlagen für Benutzer‘ abgearbeitet (vgl. Wenzel (2001), S. 266 ff.).

#### *Produktionsvorbereitung*

In dieser Phase werden die Tests abgeschlossen, Endbenutzerschulungen durchgeführt und Systemverwaltungs- und Cutover-Aktivitäten beendet. Außerdem werden letzte noch offene kritische Punkte geklärt. Der letzte Schritt dieser Phase ist die Übernahme der Geschäftsdaten aus den Altsystemen. Anschließend ist das neue System für den produktiven Einsatz bereit (vgl. Wenzel (2001), S. 280 ff.).

### *Go-Live & Support*

In der letzten Phase des ASAP wird von der vorproduktiven Umgebung in den Produktivbetrieb übergegangen. Hierfür werden sowohl ein externer als auch ein interner Support eingerichtet. Denn insbesondere in den ersten Wochen nach Inbetriebnahme treten viele Benutzerfragen auf. Weiterhin wird über das Monitoring das System optimiert und eine langfristige Supportstrategie definiert. Letztendlich wird der Projektabschluss vollzogen, wobei in einem Review noch offene Unstimmigkeiten gelöst werden (vgl. Wenzel (2001), S. 285 ff.).

Für eine ausführlichere Beschreibung der Projektphasen siehe Wenzel (2001), S. 249ff.

### *Vorteile ASAP*

Die Schaffung einer einheitlichen Arbeitsumgebung, die Möglichkeit neue Mitarbeiter schnell in ein bestehendes Projekt einzuarbeiten und der Überblick über den aktuellen Projektstatus für alle Projektteilnehmer sind einige der Vorteile der standardisierten Vorgehensweise von ASAP. Durch die detaillierte Beschreibung der einzelnen Aufgaben werden unnötige Tätigkeiten vermieden und auch Nicht-Experten können Teilaufgaben der SAP-Einführung übernehmen. Weiterhin ist durch Auflistung der Aufgaben eine genauere Personaleinsatz- und Kostenplanung möglich. Folgeprojekte sind aufgrund der gesicherten Standards und Vorgehensweisen schneller und kostengünstiger durchführbar (vgl. Brand (1999), S. 2 ff.).

Abschließend kann festgehalten werden, dass die Phasen Business Blueprint und Realisierung, in welchen das Customizing des Systems vorgenommen wird, über 50 % der Gesamtprojektzeit beanspruchen. Für ein Unternehmen mit 130 Systemnutzern verteilt auf 2 Kontinenten bedeutet das einen Aufwand von 80-120 Tagen (vgl. Brand (1999), S. 5). Da dieser Aufwand sowohl personelle wie auch finanzielle Ressourcen bindet, ist eine (Teil-) Automatisierung des Customizings erstrebenswert. Als Nebeneffekt kann durch ein automatisiertes Vorgehen die Time-to-Production verkürzt werden.

### **2.2.3 Integriertes Vorgehen mit SAP NetWeaver und ARIS**

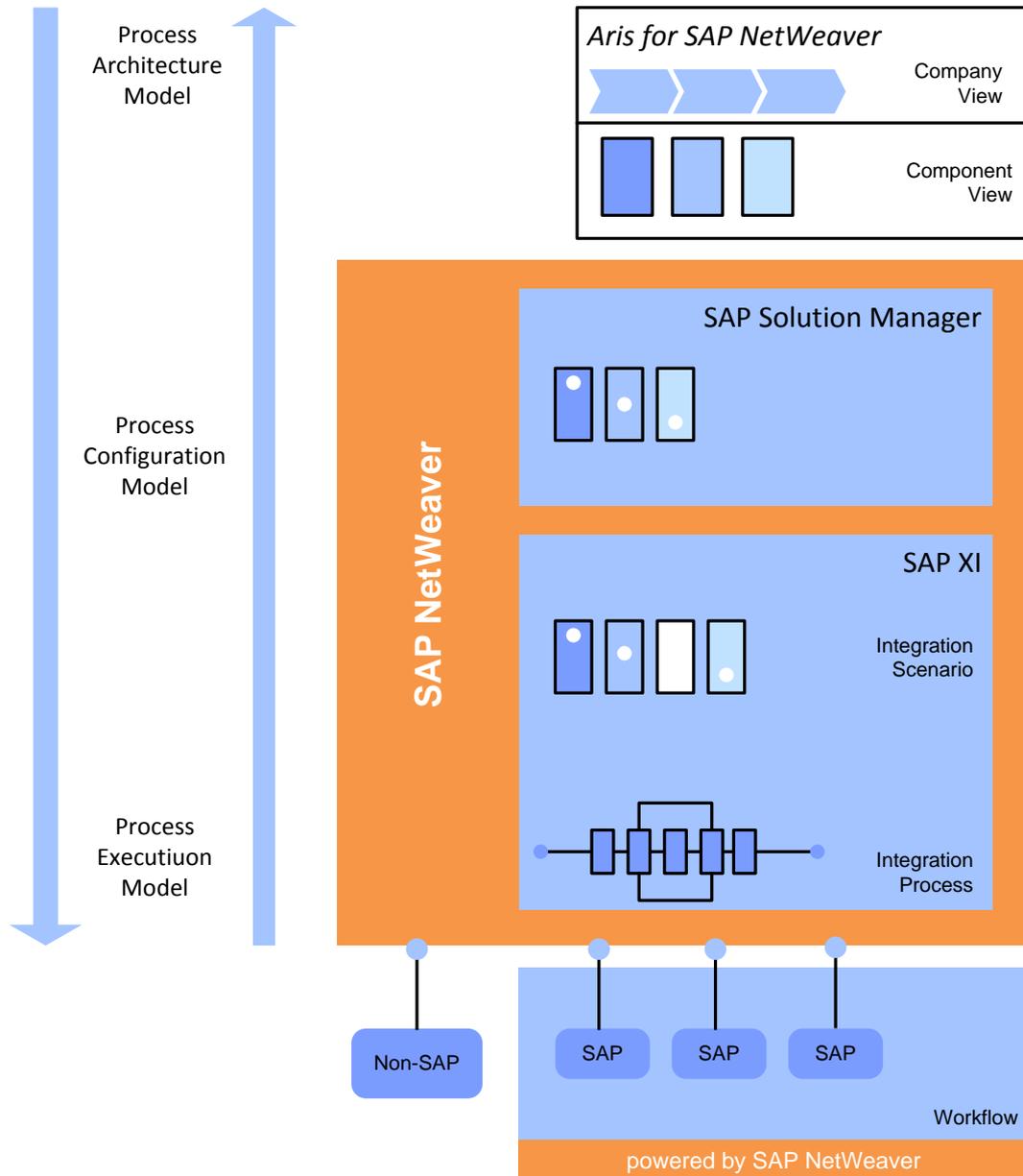
Um das Geschäftsprozessmanagement und das Customizing der ERP-Software effizienter zu gestalten, hat die SAP AG gemeinsam mit der IDS Scheer AG einen Kooperationsvertrag geschlossen. Aus diesem Bestreben ist eine Kombination von Werkzeugen entstanden mit denen versucht wird, Teile des Customizing automatisch ablaufen zu lassen (vgl. SAP (2009b), S. 1).

Ein erster Ansatz wird mit den Werkzeugen ‚ARIS Business Architect for SAP‘,<sup>24</sup> dem ‚Solution Manager‘ und ‚SAP NetWeaver‘ verfolgt. Unterstützt durch diese Werkzeuge werden vom *Prozessarchitekturmodell* über das *Prozesskonfigurationsmodell* bis hin zum *Prozessausführungsmodell* die Prozessmodelle eines Unternehmens erstellt und implementiert. In Abb. 2.2 sind die verschiedenen Modelle den jeweils realisierenden Anwendungssystemen gegenübergestellt.

Das Prozessarchitekturmodell enthält die Wertschöpfungskette sowie die Prozesslandschaft des Unternehmens, welches das ERP-System einführen möchte. Dieses wird mit dem ARIS Business Architect for SAP erstellt. Im nächsten Schritt werden Szenarien ausgewählt und die entsprechenden Referenzmodelle aus dem SAP-System in ARIS Business Architect for SAP geladen. Die Referenzmodelle werden angepasst und mit dem Solution Manager (Teil des SAP-Systems) synchronisiert, sodass im ERP-System der Business Blueprint vorliegt. Durch dieses Vorgehen ermöglichen die Werkzeuge eine durchgängige Beschreibung von der fachlichen Ebene bis zur Implementation der Unternehmensmodelle (vgl. Volmering/Scholz (2004), S. 39 f.).

---

<sup>24</sup> Früher bezeichnet als ‚ARIS for SAP NetWeaver‘.



In Anlehnung an Volmering/Scholz (2004), S. 39.

**Abb. 2.2:** Durchgängige Prozessarchitektur

### *Prozessarchitekturmodell*

Im Prozessarchitekturmodell werden die Unternehmensprozesse aus der betriebswirtschaftlichen Sicht erfasst. Nach Erfassen des Istmodells der Unternehmung wird dieses analysiert. Unter Berücksichtigung der Unternehmensziele wird dann das Sollmodell der Prozesslandschaft erstellt, sodass im Ergebnis ein Architekturmodell vorliegt (vgl. Volmering/Scholz (2004), S. 40).

### *Prozesskonfigurationsmodell*

Im nächsten Schritt werden die Referenzinhalte (z. B. Referenzmodelle) aus dem SAP Solution Manager in ARIS Business Architect for SAP importiert. Der SAP Solution Manager hält hierfür Geschäftsszenarien und Referenzprozesse bereit. Die verantwortlichen Berater wählen die von der ERP-Software zu unterstützenden Geschäftsszenarien aus. Anschließend werden die gewählten Szenarien in die Prozessarchitektur eingeordnet. Im ARIS Business Architect werden die SAP-Prozesse so weit wie möglich an die Unternehmensprozesse angeglichen. Um die Änderungen an das SAP-System zu propagieren, wird die detaillierte Anforderungsspezifikation in den SAP Solution Manager übertragen. In einem Projekt innerhalb des SAP Solution Managers erfolgt anschließend die technische Realisierung des Modells. Auf diese Weise werden die globalen Geschäftsprozesse des Unternehmens mit dem ERP-System verbunden (vgl. Volmering/Scholz (2004), S. 41).

### *Prozessausführungsmodell*

Nach der technischen Realisierung des Prozesskonfigurationsmodells werden die Prozessmodelle von der SAP Exchange Infrastructure (XI)<sup>25</sup> und den SAP-Anwendungen zur Ausführung genutzt. Außerdem kann eine abstrahierte Prozesslogik auf den Integration Server übertragen werden, sodass ERP-systemübergreifende Prozesse mit den Prozessen innerhalb des ERP-Systems verbunden werden können. Darüber hinaus können mittels der standardbasierten Entwicklungsumgebung von SAP XI Systemschnittstellen definiert und Business Workflows für eine menschliche Interaktion orchestriert werden. Abschließend können die individuellen Integrations-szenarien und -prozesse mit dem ARIS Business Architect for SAP synchronisiert und dort für die Prozessarchitektur bereitgestellt werden.

Aus den obigen Angaben kann geschlussfolgert werden, dass ARIS Business Architect for SAP ein effizientes Customizing ermöglicht. Erreicht wird dies durch die Integration der Werkzeuge für die Prozessmodellierung und das Customizing. Anzumerken ist jedoch die Fokussierung auf ein referenzmodellbasiertes Customizing. So werden die Referenzszenarien und Prozesse manuell ausgewählt, obwohl Sollmodelle aus dem Architekturmodell vorhanden sind. Ein automatischer Abgleich von Unternehmensmodellen und Referenzmodellen kann mit den beschriebenen Werkzeugen nicht umgesetzt werden.

---

<sup>25</sup> SAP XI dient als Schnittstelle zwischen Anwendungssystemen verschiedener Hersteller und ist Bestandteil von SAP NetWeaver (vgl. Alpar et. al. (2008), S. 231 f.).

### 2.3 Meta-Modelle für die Geschäftsprozessmodellierung

Die Modellierung von Unternehmensprozessmodellen und Referenzprozessmodellen erfolgt in der Regel unter Nutzung von Meta-Modellen. Zwei in der Praxis eingesetzte Modellierungssprachen für die Modellierung von Geschäftsprozessen sind die semi-formalen Ereignisgesteuerten Prozessketten (EPK) (vgl. Rautenstrauch/Schulze (2003), S. 244)<sup>26</sup> und die formalen Petri-Netze (vgl. Oberweis (1996), S. 27 f.).

Neben den drei Meta-Modellen, die im Folgenden vorgestellt werden, gibt es eine Vielzahl weiterer Meta-Modelle für die Modellierung von Geschäftsprozessen. Zur Modellierung von Abläufen eignen sich beispielsweise auch Aktivitätsdiagramme, welche Bestandteil der Unified Modeling Language sind (vgl. Rautenstrauch/Schulze (2003), S. 251). SEEMANN und VON GUDENBERG erläutern die Verwendung von Aktivitätsdiagrammen zur Prozessmodellierung (vgl. Seemann/von Gudenberg (2006) S. 27 ff.).<sup>27</sup> Ein neueres, ebenfalls von der Object Management Group veröffentlichtes, Meta-Modell ist die Business Process Modelling Notation (BPMN). Die gegenwärtig aktuelle Version 1.2 wurde im Januar 2009 veröffentlicht (vgl. OMG (2009)).<sup>28</sup>

#### 2.3.1 EPK

Die Ereignisgesteuerte Prozesskette wurde von KELLER, NÜTTGENS und SCHEER entwickelt und 1992 erstmals publiziert. Sie wird verwendet um die Steuerungssicht im ARIS-Haus<sup>29</sup> zu modellieren, welche die anderen Sichten Daten, Funktionen, Organisation und Leistung des ARIS-Hauses integriert (vgl. Keller et al. (1992), S. 5).

##### *Modellelemente*

Das Meta-Modell der Ereignisgesteuerten Prozesskette besteht aus vier Grundelementen. Diese sind Ereignisse, Funktionen, Verknüpfungsoperatoren und gerichtete Verbindungen, wie in Abb. 2.3 dargestellt. Bei der Definition der EPK haben sich die Autoren auf eine semantische Darstellung von Funktionen fokussiert, um die Zusammenhänge eines Unternehmens auf abstrakter Ebene darstellen zu können (vgl. Keller et. al. (1992), S. 2 und 11). Die Entwicklung einer formalen Syntax und

<sup>26</sup> Die ursprüngliche Entwicklung von Ereignisgesteuerten Prozessketten geht auf eine Arbeit von KELLER, NÜTTGENS und SCHEER zurück (vgl. Keller et. al. (1992), S. ).

<sup>27</sup> Für weitere Erläuterungen zu Aktivitätsdiagrammen siehe auch Booch et. al. (2006), S. 313 ff. oder Österreich (2006), S. 302 ff.

<sup>28</sup> Auf eine Auflistung weiterer potenzieller Modellierungssprachen für Geschäftsprozessmodelle wird hier aus Gründen der Übersichtlichkeit verzichtet.

<sup>29</sup> Eine vertiefende Beschreibung der einzelnen Sichten des ARIS-Hauses kann Scheer (2002), S. 32 ff. entnommen werden.

Semantik war bei deren Entwicklung von geringerer Relevanz. Wenngleich später einige Werke entstanden sind, in denen versucht wird, die EPK mit einer formalen Syntax und Semantik zu hinterlegen. Vergleiche hierzu Nüttgens/Rump (2002), S. 65, welche auch einige vorherige Arbeiten aufzählen.

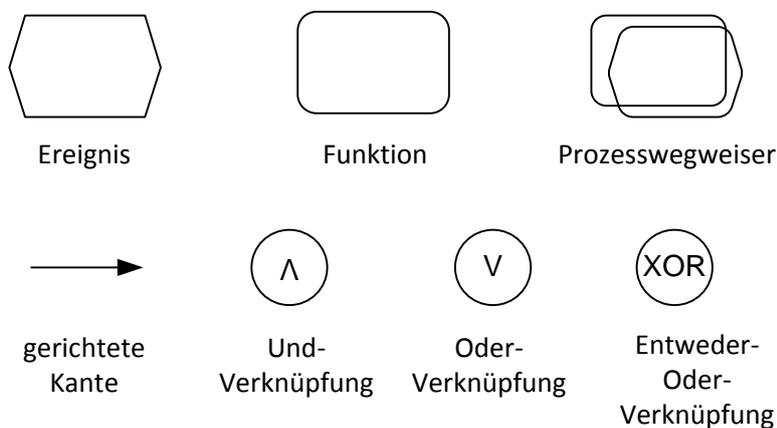


Abb. 2.3: Sprachelemente der EPK

Ein *Ereignis* ist eine passive Komponente des Modellsystems. Es stellt einen eingetretenen Systemzustand dar, welcher die weitere Ablauffolge bestimmt und somit keine Zeit verbraucht. Ereignisse stoßen betriebswirtschaftliche Funktionen an und werden von diesen nach Beendigung ausgelöst. Hieraus ergibt sich für die Modellierung von EPK die Notwendigkeit, dass alle Prozessmodelle mit einem Ereignis beginnen und enden müssen.

Eine *Funktion* hingegen ist ein aktives Systemelement, da sie die Durchführung eines betrieblichen Vorganges beschreibt. Bei der Durchführung transformiert die Funktion Inputdaten in Outputdaten und verändert dadurch den Systemzustand. Aufgrund ihres aktiven Charakters besitzen Funktionen Entscheidungskompetenz.

Um unterschiedliche Kontrollflüsse, z. B. nach Entscheidungen, darzustellen, können drei Arten von Verknüpfungsoperatoren (Routingelementen) verwendet werden. Diese sind die *Konjunktion*, *Disjunktion* und *Adjunktion*. Eine Konjunktion erlaubt eine parallele Verzweigung des Kontrollflusses, sodass die folgenden Funktionen unabhängig voneinander ausgeführt werden können. Die Disjunktion stellt ein exklusives Oder dar und erlaubt eine Verzweigung nur in einen der nachfolgenden Pfade, wohingegen die Adjunktion eine Verzweigung des Kontrollflusses sowohl in einen als auch in mehrere Pfade zulässt. Die gerichteten Verbindungen oder *Kanten*, stellen den Kontrollfluss zwischen den anderen Elementen dar.

### *Verknüpfungsregeln*

Aus den obigen Beschreibungen lässt sich die Verknüpfungsregel ableiten, dass auf ein Ereignis immer eine Funktion folgen muss und auf eine Funktion immer ein Ereignis. Durch Verknüpfen dieser Elemente mittels gerichteter Kanten entsteht so eine Ablaufkette. Zwischen den Ereignissen und Funktionen können sich dabei beliebig viele Verknüpfungsoperatoren befinden. Allerdings sind nicht alle Kombinationen von Ereignissen und Verknüpfungsoperatoren zulässig. Auf ein Ereignis darf lediglich eine Konjunktion folgen. Eine Disjunktion oder Adjunktion würde implizieren, dass zuvor eine Entscheidung über den weiteren Ablauf innerhalb der Prozesskette getroffen wurde. Eine solche Entscheidungskompetenz besitzt jedoch nur eine Funktion. Weiterhin dürfen Funktionen und Ereignisse nur eine eingehende und eine ausgehende Kante besitzen, Verknüpfungsoperatoren hingegen entweder mehrere eingehende und eine ausgehende Kante oder umgekehrt.

Ereignisgesteuerte Prozessketten werden zu den semi-formalen Sprachen gezählt, denn die oben beschriebene Menge ihrer Zeichen sowie die erlaubten Verknüpfungen zwischen diesen können auch formal definiert werden (vgl. Nüttgens/Rump (2002), S. 67 ff.). Hier soll die natürlich-sprachliche Beschreibung genügen. Doch die Semantik ist nicht immer deterministisch. Enthält eine EPK beispielsweise eine Oder-Verknüpfung (Adjunktion), so kann nicht genau bestimmt werden, welcher Pfad im Prozessablauf gewählt wird. Auch NÜTTGENS und RUMP, welche die EPK weitestgehend formalisieren, geben für die Semantik der XOR-Verknüpfung nur eine natürlich-sprachliche Definition an (vgl. Nüttgens/Rump (2002), S. 73).

### **2.3.2 Petri-Netze und Erweiterungen**

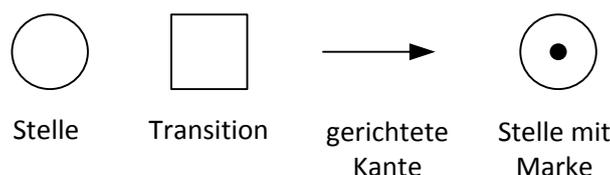
Seit der ersten Veröffentlichung zur Sprache der Petri-Netze im Jahre 1962,<sup>30</sup> hat es eine Vielzahl von Arbeiten gegeben, aus denen Weiterentwicklungen oder Unterarten der Petri-Netze hervorgegangen sind. Dies belegt die Arbeit von PATIG, in der 115 Arten von Petri-Netzen untersucht wurden und gleichzeitig keineswegs ausgeschlossen wird, dass noch weitere Modifikationen der Petri-Netze existieren (vgl. Patig (2006), S. 32). In dieser Arbeit sollen jedoch nur die markierten Stellen/Transitionen-Netze (S/T-Netze) und Workflow-Netze näher betrachtet werden.

---

<sup>30</sup> CARL ADAM PETRI entwickelte in seiner Dissertationsschrift „Kommunikation mit Automaten“ die grundlegenden Modellierungskonzepte zur Sprache der Petri-Netze (vgl. Petri (1962), S. 48 ff.)

### Modellelemente

Petri-Netze werden durch die Sprachelemente *Stellen*, *Transitionen* und *gerichtete Verbindungen* (Kanten) dargestellt. S/T-Netze verfügen zudem zusätzlich über *Marken*, die den aktuellen Zustand des Systems repräsentieren. Die Stellen werden in Form von Kreisen, Transitionen als Rechtecke, die gerichteten Verbindungen als Pfeile und die Marken als kleine ausgefüllte Kreise innerhalb der Stellen visualisiert (vgl. Reisig (1986), S. 16 f. und S. 70 f.). Die Notationen dieser Elemente können Abb. 2.4 entnommen werden.



**Abb. 2.4:** Sprachelemente der Petri-Netze

### Verknüpfungsregeln

Die Regeln zur Verknüpfung der Sprachelemente von S/T-Netzen beruhen auf einer formalen Beschreibung und können daher präzise angegeben werden. So wird ein Netz definiert als ein Tripel  $N = (S, T, F)$ , wenn gilt:

- $S \cap T = \emptyset$  und
- $F \subseteq (S \times T) \cup (T \times S)$  ist eine zweistellige (Fluss-) Relation.

Die Elemente von  $S$  werden dabei als Stelle bezeichnet, die Elemente von  $T$  als Transitionen. Gleichzeitig wird für diese beiden Elementtypen der Oberbegriff Knoten verwendet, während  $F$  als Flussrelation bzw. deren Elemente als Kanten bezeichnet werden (vgl. Baumgarten (1996), S. 50).

Ein S/T Netz ist dann ein 6-Tupel  $N = (S, T, F, K, W, M_0)$  wenn gilt:

- $(S, T, F)$  ist ein Netz,
- $K: S \rightarrow \mathbb{N} \cup \{\infty\}$  (Kapazitäten der Stellen – evtl. unbeschränkt),
- $W: F \rightarrow \mathbb{N}$  (Kantengewichte der Kanten),
- $M_0: S \rightarrow \mathbb{N}_0$  (Anfangsmarkierung) und
- $\forall s \in S: M_0(s) \leq K(s)$ , die Kapazitäten jeder Stelle somit berücksichtigt werden

(vgl. Reisig (1986), S. 71 und Baumgarten (1996), S. 79).

### *Dynamische Eigenschaften*

Für die Analyse von Prozessen sind insbesondere die dynamischen Eigenschaften von S/T-Netzen interessant. Einige dieser Eigenschaften werden hier vorgestellt.

Eine *Markierung* eines S/T-Netzes ist eine Abbildung  $M: S \rightarrow \mathbb{N}_0$  für die gilt:

$$\forall s \in S: M(s) \leq K(s).$$

Die Anfangsmarkierung eines S/T-Netzes wird mit  $M_0$  bezeichnet. Eine Änderung einer Markierung in einem S/T-Netz wird durch das Schalten einer Transition erreicht.

Bevor eine Transition  $t \in T$  schalten kann, muss diese aktiviert sein. Eine Transition ist aktiviert, wenn:

$$\forall s \in \bullet t: M(s) \geq W(s, t),$$

$$\forall s \in t \bullet: M(s) \leq K(s) - W(t, s).$$

Dabei bezeichnet  $\bullet t$  die Menge aller direkten Vorgänger von  $t$  im Netz  $N$ :

$$\bullet t = \{s \in S \mid (s, t) \in F\},$$

und  $t \bullet$  die Menge aller direkt auf  $t$  folgenden Stellen:

$$t \bullet = \{s \in S \mid (s, t) \in F\}.$$
<sup>31</sup>

Wenn eine Transition  $t$  schaltet, ändert sich die Markierung des S/T-Netzes. Die Notation  $M [t \rangle M'$  bezeichnet das Schalten der Transition  $t$  unter Markierung  $M$  sodass die Folgemarkierung  $M'$  erreicht wird. Mehrere Schaltungen hintereinander werden als Schaltfolge  $w$  bezeichnet. Die analoge Notation hierzu ist  $M [w \rangle M'$  (vgl. Baumgarten (1996), S. 80 ff.)<sup>32</sup>.

<sup>31</sup> Eine allgemeine Definition zu Vor- und Nachbereich von Netzknoten befindet sich in Baumgarten (1996), S. 51.

<sup>32</sup> Für eine ausführlichere Beschreibung der dynamischen Eigenschaften siehe Baumgarten (1996), S. 80 ff.

### *Erweiterungen der S/T-Netze*

Für Modellierungsvorhaben, welche Geschäftsprozesse zum Modellierungsgegenstand haben, sind die Erweiterungen der Petri-Netze um Farbe, Zeit und Hierarchie interessant (vgl. van der Aalst/van Hee (2004), S. 41).

Mit der Erweiterung um *Farben* werden die Marken nicht einfach nur farbig dargestellt. Vielmehr sollen die Farben verdeutlichen, dass es sich um unterscheidbare Marken handelt. Die Unterscheidung der Marken wird durch die Zuweisung von Werten möglich. Auf diese Weise können verschiedene Attribute eines Workflowfalles<sup>33</sup> in den Marken verankert werden. Darüber hinaus können Stellen typisiert werden, um festzulegen, welche Marken bzw. Attribute akzeptiert werden. Für Transitionen kann festgelegt werden, welche Bedingungen die Marken in den Input-Stellen erfüllen müssen. Die so eingeschränkte Transition kann nur schalten, wenn die zu konsumierenden Marken der Input-Stellen die Bedingungen erfüllen. In farbigen Petri-Netzen müssen für Transitionen folgende Merkmale definiert werden:

- Angabe, ob es für die Transition Vorbedingungen gibt. Wenn ja, muss eine präzise Definition vorliegen.
- Die Anzahl der erzeugten Marken für jede Output-Stelle.
- Die Werte, welche die erzeugten Marken besitzen. Enthält die Transition zum Beispiel eine Summenfunktion, so werden die Werte der verbrauchten Marken in der Output-Stelle aufsummiert.

Eine formale Beschreibung von farbigen Petri-Netzen kann in JENSEN nachgelesen werden (vgl. Jensen (1991), S. 350 ff.).

In einfachen S/T-Netzen feuern bzw. schalten Transitionen sofort, nachdem sie aktiviert wurden. Um auch Aussagen wie ‚Versende nach 7 Tagen eine Erinnerung‘ zu ermöglichen, gibt es Netzvarianten, die um den Faktor *Zeit* erweitert wurden. Hierfür erhält jede Marke einen Zeitstempel. Dieser besagt, zu welchem Zeitpunkt die Marke frühestens verbraucht werden kann. Der Aktivierungszeitpunkt einer Transition ergibt sich dann aus dem maximalen Zeitstempel der zu verbrauchenden Marken. Sind mehrere Marken in einer Stelle vorhanden, so werden diejenigen mit dem kleinsten Zeitstempel zuerst konsumiert. Da eine Transition schaltet, sobald sie aktiviert ist, schaltet bei konkurrierenden Transitionen diejenige mit dem kleineren Aktivierungszeitpunkt zuerst. Erzeugte Marken können eine Verzögerungszeit besitzen, welche durch die schaltende Transition bestimmt wird. Der Zeitstempel einer erzeugten Marke

---

<sup>33</sup> Ein Workflowfall ist eine Instanz eines Workflows.

ergibt sich aus deren Schaltzeitpunkt plus der Verzögerungszeit. Die Verzögerungszeit kann auch von den Werten einer Marke abhängig sein oder zufällig bestimmt werden. Das Schalten einer Transitionen verbraucht keine Zeit (vgl. van der Aalst/van Hee (2004), S. 45 f.).

Die Erweiterung der S/T-Netze um das Konzept der *Hierarchie* wird mittels Transitionsverfeinerung realisiert. Dabei werden die äußeren Transitionen eines Teilprozesses zu einer einzigen Transition im übergeordneten Prozess zusammengefasst.<sup>34</sup> Durch die hierarchische Strukturierung wird das Modellieren von Prozessen auf unterschiedlichen Abstraktionsebenen möglich. Weiterhin kann auch das Prinzip der Wiederverwendung von Teilprozessen angewendet werden (vgl. van der Aalst/van Hee (2004), S. 46 ff.).

### 2.3.3 Workflow-Netze

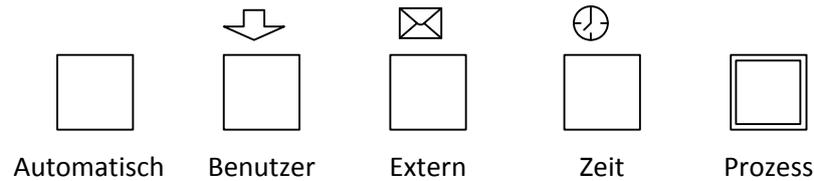
In diesem Abschnitt wird erläutert, wie Geschäftsprozesse mittels Workflow-Netzen modelliert werden können. Da diese von den Petri-Netzen abstammen, werden nur die ergänzenden Konzepte vorgestellt. Zunächst werden die für Workflow-Netze benötigten Sprachelemente vorgestellt.

#### *Modellelemente*

Neben den klassischen Transitionen respektive Aufgaben, welche direkt bei Aktivierung feuern, haben VAN DER AALST und VAN HEE drei weitere Aufgabenarten bzw. *Trigger* eingeführt. Die Trigger ‚Benutzer‘, ‚Extern‘ und ‚Zeit‘ sowie die klassische Transition sind in Abb. 2.5 dargestellt. Der Trigger ‚Benutzer‘ verdeutlicht, dass das System auf Eingaben vom Benutzer wartet bzw. dass diese Aktivität durch einen Benutzer ausgeführt wird. Mit dem Trigger ‚Extern‘ wird modelliert, dass der Prozess für die Fortsetzung auf ein Ereignis aus der Prozessumgebung wartet. Zum Beispiel auf eine Bestätigung für einen Zahlungseingang aus einem anderen Prozess. Eine verzögerte Aktivierung einer Aufgabe kann mit dem Trigger ‚Zeit‘ modelliert werden (vgl. van der Aalst/van Hee (2004), S. 63 ff.).

---

<sup>34</sup> Zur Vergrößerung und Verfeinerung von transitionsberandeten Netzen siehe auch Baumgarten (1996), S. 58 ff.

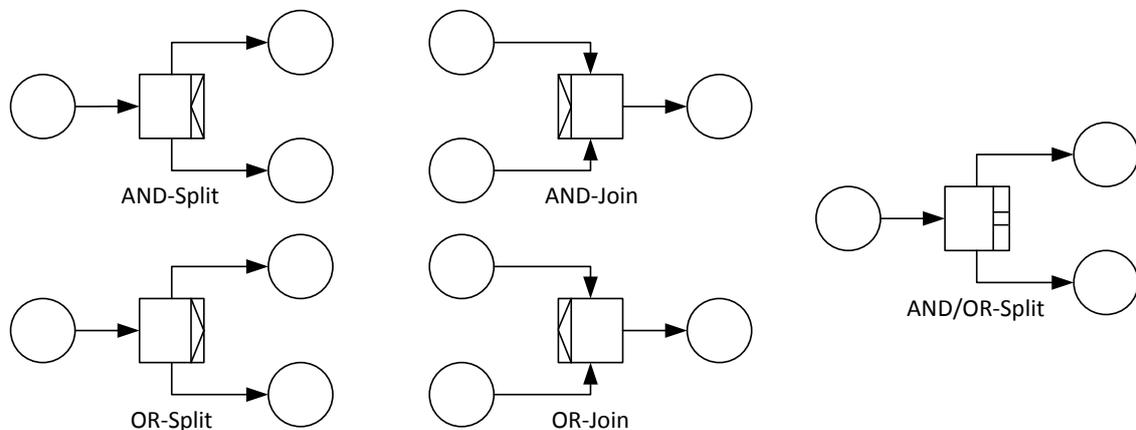


**Abb. 2.5:** Aufgabenarten und Prozesse in Workflow-Netzen

Um die hierarchische Strukturierung kenntlich zu machen, haben VAN DER AALST und VAN HEE ebenfalls ein weiteres Symbol eingeführt. Mit dem Symbol ‚Prozess‘ aus Abb. 2.5 werden Teilprozesse in Workflow-Netzen abgebildet (vgl. van der Aalst/van Hee (2004), S. 46 f). Somit ist eine durchgehende Modellierung auf verschiedenen Abstraktionsebenen von der Fachkonzept- bis zur Detailebene möglich ohne die Modellsyntax wechseln zu müssen. Änderungen auf der Detailebene sind folglich automatisch im Gesamtmodell enthalten.

### *Verknüpfungsregeln*

Basierend auf der Spezifikation der Petri-Netze haben VAN DER AALST und VAN HEE die Operatoren ‚And-Split‘, ‚And-Join‘, ‚OR-Split‘ und ‚OR-Join‘ eingeführt (vgl. van der Aalst/van Hee (2004) S. 53 ff.).



In Anlehnung an van der Aalst/van Hee (2004), S. 59.

**Abb. 2.6:** Operatoren in Workflow-Netzen

Hervorzuheben ist, dass VAN DER AALST und VAN HEE nicht einfach nur ‚neue‘ Symbole eingeführt haben. Die Semantik der ‚neuen‘ Symbole beruht auf der Semantik von Petri-Netzen und den Erweiterungen zu farbigen Petri-Netzen, hierarchischen Petri-Netzen und der Erweiterung um die Komponente Zeit. Hieraus ergibt sich der Vorteil, dass auf Workflow-Netze ebenso wie auf Petri-Netze zahlreiche Formen von (formalen) Analysen angewendet werden können. Die formale Definition von Workflow-Netzen

kann bei VAN DER AALST und VAN HEE nachgelesen werden (vgl. van der Aalst (1998), S. 36 ff.; van der Aalst/van Hee (2004), S. 271 ff.).

### **3 Konzept und Voraussetzungen für die Automatisierung des Customizing**

Nachdem die grundlegenden Begriffe und Konzepte im Gebiet der Einführung und des Customizing von ERP-Systemen bekannt sind, wird nun die von KASSEM entworfene Idee des Self Adaptive Customizing vorgestellt. Weiterhin werden auch die an die Modellierungssprache und das ERP-System gestellten Anforderungen beleuchtet. Dies ist notwendig, da einige der Vorbedingungen des fünften Kapitels aus dem SAC-Konzept abgeleitet werden.

#### **3.1 Konzept des Self Adaptive Customizing**

In diesem Unterkapitel wird der Lösungsansatz vorgestellt, welcher die Automatisierung des Customizing bei der Einführung von ERP-Software ermöglichen soll. Das Konzept des Self Adaptive Customizing unterscheidet zwei Arten des Customizing: Das Initiale Customizing und das Kontinuierliche Customizing. Zur Realisierung des Konzeptes schlägt KASSEM die Nutzung eines Self Adaptive Customizing Middleware Systems (SACMS) vor (vgl. Kassem (2007), S. 73). In den nächsten Abschnitten werden nach Vorstellung der allgemeinen Merkmale des Self Adaptive Customizing diese Begriffe näher erläutert.

##### **3.1.1 Ziel und Gegenstand des Self Adaptive Customizing**

Die im zweiten Kapitel vorgestellten Methoden zur Einführung bzw. Customizing einer ERP-Software bieten bereits die Möglichkeit, Teilschritte des Customizingprozesses effizient umzusetzen. Keine dieser Methoden unterstützt jedoch die Automatisierung des gesamten Prozesses von der Prozessmodellierung bis zur Implementierung. Um diese Lücke zu schließen, wurde von KASSEM das Konzept des Self Adaptive Customizing entwickelt. Mit diesem Konzept wird die Vision einer sich selbstständig (automatisiert) anpassenden ERP-Software an die betrieblichen Anforderungen eines Unternehmens verfolgt (vgl. Kassem (2007), S. 63).

Neben der adaptiven ERP-Software muss auch der *Kontext* des Systems betrachtet werden, da Kontextänderungen die Auslöser für das Anpassen eines Systems sind (siehe Abschnitt 2.1.4). Im Rahmen des Self Adaptive Customizing dienen *Unternehmensprozessmodelle* und *Analysen* der Systemnutzung als Umgebungs- und Benutzerkontext (vgl. Kassem (2007), S. 63).

Während des Customizing sind auch die *Workflow-Bezugsobjekte* von besonderer Bedeutung. Unter dem Begriff Workflow-Bezugsobjekt versteht KASSEM Input-Objekte, Output-Objekte und Workflow-Leistungsobjekte. Input-Objekte werden zum Ausführen einer oder mehrerer Workflowaktivitäten benötigt. Output-Objekte eines Workflows sind alle Objekte, die durch eine oder mehrere Aktivitäten des entsprechenden Workflows verändert werden. Leistungsobjekte<sup>35</sup> sind Workflowobjekte, welche an jeder Aktivität eines Workflows als Input oder Output-Objekte beteiligt sind (vgl. Kassem, (2007), S. 66 f.). Gemäß dieser Definition können Workflowobjekte sowohl Input- als auch Output-Objekte einer Aktivität sein. Orthogonal zu dieser Einteilung unterscheidet KASSEM weiterhin die disjunkten Mengen von Organisationsobjekten und Ressourcen-Objekten. Organisationsobjekte repräsentieren Organisationseinheiten des Unternehmens. Alle anderen Geschäftsobjekte werden den Ressourcen-Objekten zugeordnet (vgl. Kassem (2007), S. 67).

Beim Customizing von ERP-Systemen treten *zwei Hauptprobleme* auf. Zum einen müssen während des Customizing eines ERP-Systems die Terminologien des Unternehmens und des ERP-Systems integriert werden. Zum anderen müssen die zu verwendenden Prozesse aus dem Referenzmodell ausgewählt und angepasst werden. Und zwar in einer Weise, sodass die Unternehmensprozesse möglichst vollständig und realitätsgetreu im ERP-System abgebildet sind (vgl. Kassem (2007), S. 64 f.). In dieser Arbeit wird untersucht, ob das letztere der beiden Probleme durch eine automatisierte Methode gelöst werden kann. Mithilfe dieser Methode sollen zu erfassten Ist- (oder Soll-) Prozessen eines Unternehmens die geeigneten Referenzprozesse gefunden werden.

Zurzeit ist das Self Adaptive Customizing ein Konzept, welches durch Forschungen bezüglich der einzelnen Phasen und Schritte dieses Konzeptes zu einem anwendbaren Verfahren entwickelt werden kann. In den folgenden Abschnitten werden die einzelnen Aufgaben, die durch das Self Adaptive Customizing realisiert werden sollen, erläutert.

### 3.1.2 Initiales Customizing

Das Initiale Customizing<sup>36</sup> wird, wie in Abbildung Abb. 3.1 zu sehen, in die drei Phasen *Konvertierung*, *Evaluierung* und *Implementierung* unterteilt. Es wird bei der Einführung von ERP-Software eingesetzt und wird daher nicht oder nur in großen zeitlichen Abständen wiederholt (vgl. Kassem (2007), S. 65).

---

<sup>35</sup> BECKER und KAHN verwenden anstelle des Begriffes Leistungsobjekt die Bezeichnung prozessprägendes Objekt (vgl. Becker/Kahn (2005), S. 6).

<sup>36</sup> KASSEM verwendet den Begriff ‚erstes Customizing‘, (vgl. Kassem (2007), S. 65).



**Abb. 3.1:** Phasen des Initialen Customizing

### *Konvertierungsphase*

Zu Beginn des Customizing müssen die unterschiedlichen Termini des Unternehmensprozessmodells und der Referenzmodelle integriert werden. Für diesen Zweck schlägt KASSEM die Nutzung von Text-Mining-Methoden vor (vgl. Kassem (2007), S. 66). In einem zweiten Schritt innerhalb der Konvertierungsphase werden die Unternehmensprozessmodelle und die Referenzprozessmodelle in eine einheitliche formale Sprache transformiert. Dieser Schritt ist notwendig, um einen automatisierten Vergleich der Modelle zu ermöglichen. Das Ergebnis dieser Phase sind Prozessmodelle, welche in der gleichen Meta-Sprache und der gleichen Terminologie vorliegen.

Ansätze zur Realisierung der Terminologieintegration könnten aus Konzepten wie dem Resource Description Framework, Ontologien, Topic Maps, Klassifikationen oder Thesauri abgeleitet werden. Alternativ können auch Standards entwickelt werden, welche dann von Softwareherstellern und Anwenderunternehmen angewendet werden. Für den Bereich des Personalwesens hat das HR-XML Consortium bereits einen zertifizierungsfähigen Standard definiert (vgl. Tolksdorf et. al. (2006), S. 21). Zur automatischen Transformation von Modellierungssprachen existieren bereits einige Ansätze. So haben LOHMANN, VERBEEK und DIJKMAN die Business Process Execution Language, Ereignisgesteuerte Prozessketten, Yet Another Workflow Language und die Business Process Modelling Notation betrachtet und deren Transformierbarkeit in Petri-Netze untersucht (vgl. Lohmann et. al. (2009), S. 46 ff.). Ebenso haben CHEN und SCHEER die Darstellung von EPK durch farbige Petri-Netze untersucht (vgl. Chen/Scheer (1994), S. 8 ff.)

### *Evaluierungsphase*

In der anschließenden Evaluierungsphase werden dann die vom ERP-System bereitgestellten Referenzprozessmodelle hinsichtlich ihrer Eignung zur Umsetzung der Unternehmensprozesse bewertet. Ziel dieser Phase ist es, die Referenzprozesse auszuwählen, welche am besten das Unternehmensmodell abbilden. Am besten bedeutet hier Referenzprozesse, welche die Unternehmensprozesse möglichst vollständig und unverändert abbilden. Denn jede Abweichung zwischen Referenzmodell und Unternehmensmodell muss aufwendig behoben werden. Die Auswahl des am besten geeigneten

Referenzmodells zu dem gegebenen Unternehmensmodell soll vollständig automatisch realisiert werden. Am Ende dieser Phase sind die zu verwendenden Referenzprozesse aus dem Referenzprozessmodell des ERP-Systems bestimmt (vgl. Kassem, (2007), S. 68).

Eventuell identifizierte Differenzen zwischen den Prozessmodellen können auf zwei Wegen eliminiert werden. Einerseits kann das ERP-System durch Programmierung verändert werden. Andererseits können die Unternehmensprozesse an die Prozesse des Referenzmodells angepasst werden. Ebenso wie die Modifikation des ERP-Systems ist auch die zweite Variante mit Aufwand verbunden, denn hierbei muss ein konsequentes Changemanagement betrieben werden. Bei den Mitarbeitern muss Akzeptanz für die veränderten Prozesse geschaffen werden und neben der Kommunikation der neuen Prozesse sind eventuell auch Schulungen der involvierten Mitarbeiter notwendig.

### *Implementierungsphase*

In der Implementierungsphase wird das ERP-System gemäß den Vorgaben der gewählten Referenzprozesse konfiguriert. Hierfür werden historische ‚best-practice‘ Informationen genutzt, welche in den Referenzmodellen hinterlegt sind (vgl. Kassem (2007). S. 68 f.). Weiterhin können hier auch Parametereinstellungen aus dem Unternehmensmodell, die während der Evaluierung in die gewählten Referenzprozesse übertragen wurden, genutzt werden.

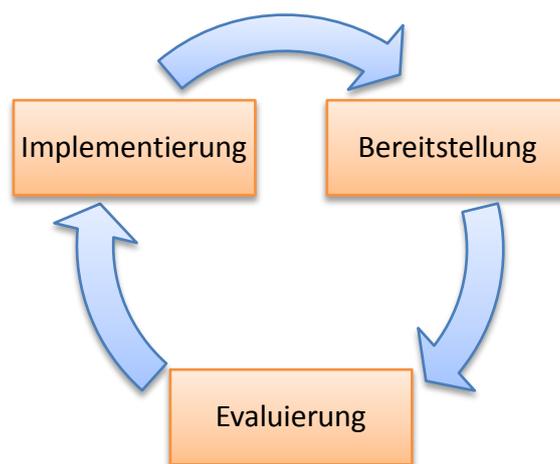
### **3.1.3 Kontinuierliches Customizing**

Da die Prozesse eines Unternehmens ständig den veränderten Markt- und Kundenanforderungen angepasst werden, müssen auch die Anwendungssysteme, welche die operative Ausführung dieser Prozesse unterstützen oder gar erst ermöglichen, adaptiert werden.<sup>37</sup> Dies ist die Aufgabe des Kontinuierlichen Customizing.<sup>38</sup> Dabei werden die Anpassungen der ERP-Software ständig verfeinert.

Ebenso wie das Initiale Customizing ist auch das Kontinuierliche Customizing in drei Phasen untergliedert. Auf die *Bereitstellung* der aktuellen Prozessmodelle folgt die *Evaluierungsphase* des Kontinuierlichen Customizing, bevor in der *Implementierungsphase* die veränderten Prozesse in die ERP-Software übertragen werden. Nach beenden der Implementierungsphase ist das Kontinuierliche Customizing aber nicht abgeschlossen, sondern wird wie in Abb. 3.2 dargestellt, in einem ständigen Zyklus wiederholt.

<sup>37</sup> Siehe hierzu auch Scheer et. al. (2005), S. 2 f. oder Becker/Kahn (2005), S. 9 ff.

<sup>38</sup> KASSEM verwendet den Begriff ‚Nach-Customizing‘ (vgl. Kassem (2007), S. 65).



**Abb. 3.2:** Zyklus Kontinuierliches Customizing

Zu Beginn des Kontinuierlichen Customizing werden in der *Bereitstellungsphase* die aktuell real ausgeführten Geschäftsprozesse ermittelt. Gewonnen werden diese Istmodelle der Prozesse durch die Methode des Application Usage Mining. Das Kontinuierliche Customizing wird unterteilt in die Vorbereitungsphase, die Muster-Entdeckungsphase und die Muster-Analysephase. In der Vorbereitungsphase werden Daten aus mehreren Quellen integriert. Hierbei werden ERP-Trace-Daten ebenso berücksichtigt wie Meta-Interaktionsdaten, welche die ERP-Trace-Daten beschreiben.<sup>39</sup> Anschließend werden in der Muster-Entdeckungsphase Instanzen der ausgeführten Workflows identifiziert. Während der Muster-Analysephase werden Verfahren des Workflow Mining auf die Workflow-Instanzen angewendet und somit die Workflow-Modelle erstellt (vgl. Kassem (2006), S. 30 ff.).

In der darauf folgenden *Evaluierungsphase* werden die Prozesse analysiert. Ziel ist es, die Prozessstrukturen besser an die Bedarfe des Unternehmens anzupassen und mit Blick auf das Unternehmensziel zu optimieren. Schwerpunkte der Analysen sind zum Einen der Zusammenhang zwischen Workflowstruktur und Workflowattributen (z. B. Workflow-Objekte). Hierbei sollen Schwachstellen oder unzulässige Vorgänge entdeckt werden. Zum Anderen werden die Interaktionen der Nutzer mit dem ERP-System analysiert. Eingesetzt werden hierfür Mining-Methoden wie die Pfadanalyse, Assoziationsanalyse, Sequenzanalyse und Nutzerverhaltensanalyse. Ergebnis dieser Phase ist ein angepasstes Workflow-Modell (vgl. Kassem (2007), S. 71 f.).

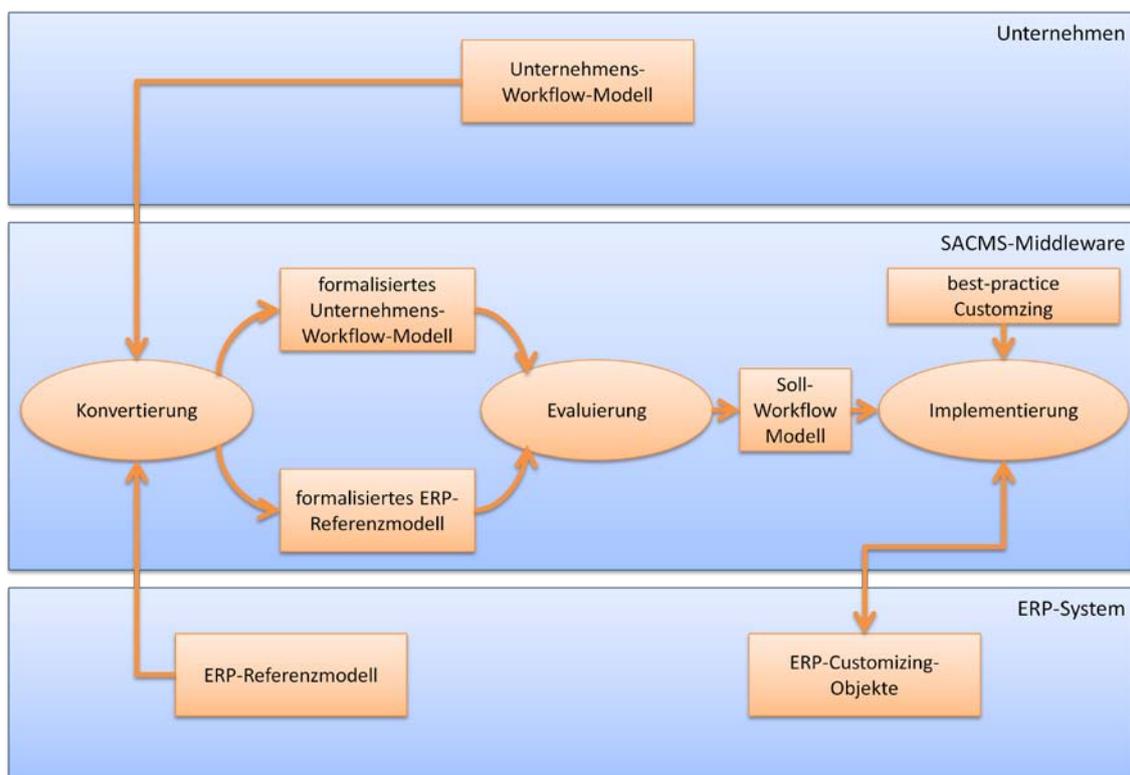
Das modifizierte Workflow-Modell wird dann in der *Implementierungsphase* in der ERP-Software umgesetzt. Zu diesem Zweck werden, analog zum Initialen Customizing,

<sup>39</sup> Die Methode des Application Usage Mining kann in Kassem (2006) nachgelesen werden.

die Customizing-Objekte des ERP-Systems automatisch und gemäß den Vorgaben des modifizierten Workflow-Modells eingestellt.

### 3.1.4 Self Adaptive Customizing Middleware System

Die technische Plattform für die Umsetzung des Self Adaptive Customizing soll das Self Adaptive Customizing Middleware System bilden. Diese Architekturschicht dient als Verbindungsebene zwischen den Unternehmensmodellen und der ERP-Software und ihren Referenzmodellen (vgl. Kassem (2007), S. 73). In Abb. 3.3 ist die SAC-Landschaft des Initialen Customizing abgebildet. Dort ist zu erkennen, dass das SACMS für die Konvertierung, Evaluierung und die Implementierung zuständig ist. Hauptaufgabe des SACMS ist die Vermittlung zwischen den verschiedenen Prozessmodellierungssprachen und den unterschiedlichen ERP-Systemen (vgl. Kassem (2007), S. 65).



In Anlehnung an Kassem (2007), S. 66.

**Abb. 3.3:** Systemlandschaft Initiales Customizing

## 3.2 Anforderungen und Auswahl der Modellierungssprache

Im nächsten Abschnitt werden die Anforderungen vorgestellt, die von der Meta-Sprache erfüllt werden müssen, welche für die Evaluierungsphase des Initialen Customizing

verwendet wird. Im darauffolgenden Abschnitt wird die Auswahl der Workflow-Netze begründet.

### 3.2.1 Anforderungen an die Modellierungssprache

Zweck der Modellierung von Geschäftsprozessen ist die Vermittlung von Informationen zwischen zwei Lebenswelten. Im konkreten Fall zwischen den Fachexperten aus dem betriebswirtschaftlichen Bereich und den Fachexperten aus der IT-Welt. Modellierungssprachen, welche hierfür zum Einsatz kommen, müssen folglich die Elemente von Geschäftsprozessen darstellen können. Für den Vergleich von Prozessmodellen sind insbesondere die betriebswirtschaftlichen Funktionen und deren Reihenfolgebeziehungen sowie deren Einbettung in die Aufbauorganisation und die an dem Prozess beteiligten Ressourcen von Interesse (vgl. Kassem (2007), S. 64).

Neben der Modellierbarkeit der Elemente eines Geschäftsprozesses ist eine mathematische Fundierung der Modellierungssprache für das Self Adaptive Customizing von Vorteil. Somit können Vergleiche zwischen Modellen durch Algorithmen definiert werden. Diese Algorithmen können dann leicht automatisiert ausgeführt werden. Ein Vergleich von semi-formalen Modellen ist ebenfalls möglich. Es bleibt aber zu bedenken, dass zwischen Modellen dieser Sprachen nicht entscheidbare Situationen auftreten können. Ist dies der Fall, kann nicht *automatisiert* geschlossen werden, ob die vorliegenden Modelle äquivalent sind.

In der Konvertierungsphase des Initialen SAC werden die Prozessmodelle sowohl des Unternehmens als auch die Referenzmodelle der ERP-Software in eine gemeinsame Modellierungssprache überführt (siehe Abschnitt 3.1.2). KASSEM fordert hierfür die Nutzung einer formalen Modellierungssprache als Zielsprache (vgl. Kassem (2007), S. 66).

### 3.2.2 Auswahl der Modellierungssprache

Die Elemente von Geschäftsprozessen können mit Workflow-Netzen wie folgt abgebildet werden:

- Die betriebswirtschaftlichen Funktionen werden in Workflow-Netzen als Transitionen dargestellt.
- Die Reihenfolgebeziehung kann mittels der gerichteten Kanten abgebildet werden.

- Die Workflow-Bezugsobjekte wie Organisationsobjekte und Ressourcen-Objekte können durch Marken dargestellt werden.

Folglich sind Workflow-Netze mächtig genug, um sämtliche relevanten Eigenschaften von Geschäftsprozessen zu modellieren.

Ein weiterer Vorteil der Workflow-Netze ist die hierarchische Modellierung. Durch die Verfeinerung von Transitionen kann ein Prozess von der höchsten Fachebene bis hinunter zur Detailebene konkretisiert werden, ohne die Modellierungssprache wechseln zu müssen. Auf der untersten Ebene können die Modelle dann für die Steuerung von Abläufen durch Workflow Engines<sup>40</sup> genutzt werden. Dadurch wird ein Informationsverlust, welcher bei einem Wechsel der Modellierungssprache vorkommen kann, vermieden.

Auch Ereignisgesteuerte Prozessketten können die Elemente eines Geschäftsprozesses abbilden und Teilprozesse hierarchisch strukturieren. Die modellierten Prozesse können jedoch nicht von einer Workflow Engine ausgeführt werden.

Ebenso empfiehlt WESKE die Nutzung von Workflow-Netzen für die Modellierung von Geschäftsprozessen (vgl. Weske (2007), S. 169). Er begründet dies unter anderem mit den *Analysemöglichkeiten* für die Eigenschaften von Geschäftsprozessen und die *Werkzeugunabhängigkeit*.

Aufgrund der formalen Fundierung von Workflow-Netzen können diese für Beweise von bestimmten Eigenschaften genutzt werden und beispielsweise die Korrektheit von Prozessmodellen verifizieren (vgl. van der Aalst (1998), S. 21). Weiterhin werden auch die Analysearten Validieren und Performance-Analyse unterstützt. Als Analysetechniken kommen dafür Erreichbarkeitsgraphen, Invarianten (für Strukturanalyse) und Simulationen (für dynamische Aspekte) zur Anwendung (vgl. van der Aalst/van Hee (2004), S. 99 ff.).<sup>41</sup>

Für die Modellierung von Workflow-Netzen existieren mehrere Werkzeuge, von denen einige als Open-Source-Software angeboten werden. Darüber hinaus ist der Formalismus der Workflow-Netze herstellerunabhängig. Ein Austausch von Netzen zwischen den verschiedenen Werkzeugen wird durch den Standard ‚Petri Net Markup Language‘ unterstützt. Dieser garantiert die Transportierbarkeit der Modelle zwischen ver-

---

<sup>40</sup> Eine Workflow Engine ist gemäß der Spezifikation der Workflow Management Coalition ein Softwareartefakt, der für die Ausführung eines Workflows verantwortlich ist (vgl. Hollingsworth (1995), S. 13 f.).

<sup>41</sup> Die Analyse von Workflow-Netzen wird unter anderem durch die Werkzeuge ProM und Woflan unterstützt.

schiedenen Anwendungssystemen wie dem Self Adaptive Customizing Middleware System und einem ERP-System.

Workflow-Netze stoßen aufgrund ihrer formalen Basis für die Modellierung von Geschäftsprozessen nicht nur auf Zustimmung. Ein mögliches Argument gegen die Verwendung von Workflow-Netzen könnte dabei die komplizierte Handhabung beim (kreativen) Erstellen von Prozessmodellen sein. Auch könnte ein erhöhter Lernaufwand bemängelt werden, bevor ein Laie (aber kompetenter Fachspezialist) mit dem Modellieren beginnen kann.

Dem ist jedoch entgegenzusetzen, dass auch mit Workflow-Netzen genauso schnell modelliert werden kann wie mit anderen (semi-formalen) Sprachen. Denn der Fachspezialist (und Modellierungslaie) muss nicht alle formalen Eigenschaften von Workflow-Netzen kennen. Vielmehr kann er einfach in einem kreativen Vorgang Istmodelle erfassen oder Sollmodelle entwerfen. Im zweiten Schritt können dann, durch ein Modellierungswerkzeug unterstützt, diverse Korrektheits- und Plausibilitätsprüfungen durchgeführt werden. Hierbei kommt wiederum der formale Charakter der Workflow-Netze positiv zu tragen, denn erst durch deren formale Spezifikation sind viele der Verifikationsmethoden überhaupt möglich.

Um diesen zweistufigen Ansatz zu verfolgen, muss natürlich das Werkzeug, mit welchem modelliert wird, auch inkorrekte Workflow-Netze zulassen (in einem kreativen Modus). In einer erweiterten Version dieses Werkzeugs könnten dann Online-Hinweise gegeben werden, mit welchen Schritten das Modell in ein korrektes Modell überführt werden kann.

In Bezug auf das Self Adaptive Customizing ist hier die fehlende formale Basis der EPK der Grund für die Entscheidung zur Nutzung von Workflow-Netzen in dieser Arbeit.

### **3.3 Anforderungen an das ERP-System**

Wie bereits oben erwähnt, wird für das Self Adaptive Customizing vorausgesetzt, dass das Unternehmensmodell sowie eine Menge von Referenzmodellen vorliegen. Daher muss eine ERP-Software, die für das SAC eingesetzt werden soll, Referenzmodelle enthalten. Die Meta-Sprache der Referenzmodelle ist von untergeordneter Relevanz. Denn wenn eine Transformationsmethode für die Meta-Sprache zur Verfügung steht, können die Referenzmodelle in die Zielsprache überführt werden (siehe Abschnitt 3.1.2). Es genügt jedoch nicht, die Referenzmodelle schlicht bereitzuhalten. Vielmehr müssen diese eng mit der Software verknüpft sein, sodass ein gewähltes, konfiguriertes und

parametrisiertes Modell automatisch von der ERP-Software umgesetzt werden kann. Darüber hinaus muss die ERP-Software eine Schnittstelle besitzen, durch welche die Referenzmodelle dem Self Adaptive Customizing Middleware System verfügbar gemacht werden (siehe Abb. 3.3).

Die mögliche Reichweite der Anpassungen wird von der Flexibilität der ERP-Software bestimmt. Sollen beim Customizing die Reihenfolgebeziehungen des Referenzmodells geändert werden, so muss das ERP-System die Änderung dieser Reihenfolge unterstützen. Sind die Reihenfolgebeziehungen zwischen den verschiedenen Funktionen im Quellcode fest verankert, so können lediglich die zu verwendenden Funktionen ausgewählt und Entscheidungsparameter angegeben werden. Eine neue Anordnung der Funktionen gemäß dem Unternehmensmodell ist dann nur durch (manuelle) Modifikation der ERP-Software möglich. Um die ERP-Software auf effiziente Weise zu modifizieren, muss diese über eine integrierte Entwicklungsumgebung verfügen.

Der Kontext für das Kontinuierliche Customizing, die Informationen über die Nutzung der ERP-Software, muss ebenfalls zur Verfügung stehen. Folglich muss die ERP-Software in der Lage sein die für die Bereitstellungsphase notwendigen Monitoring- (Trace-) Daten zu verwalten.

## 4 Automatisierte Suche von Referenzmodellen

Nachdem das abstrakte Konzept zur Automatisierung des Customizing im vorangehenden Kapitel dargestellt wurde, werden nun die entwickelten Verfahren zur Suche des am besten geeigneten Referenzmodells detailliert erläutert und gegenübergestellt. Hierfür wurde recherchiert, welche Methoden für den Vergleich von Workflow-Netzen bereits existieren. Nach einer allgemeinen Beschreibung werden im dritten Unterkapitel die angewendeten Kriterien vorgestellt. Daran anschließend werden die zwei in dieser Arbeit entwickelten Verfahren zur Suche von Referenzmodellen zu gegebenen Unternehmensmodellen erläutert. Die Aufgabe dieser Verfahren kann folgendermaßen beschrieben werden: Finde zu einem gegebenen Unternehmensmodell das am besten passende Referenzmodell aus der Menge der verfügbaren Referenzmodelle.

### 4.1 Existierende Methoden für den Vergleich von Workflow-Netzen

Nachdem bekannt ist, welche Aufgabe in der Evaluationsphase des Initialen Customizing bearbeitet wird, stellt sich die Frage, mit welchen Methoden bzw. Verfahren diese realisiert werden kann. Aus diesem Grund werden bereits existierende Verfahren vorgestellt und hinsichtlich ihrer Eignung für die Evaluationsphase geprüft.

#### *Methoden von van Dongen, Dijkman und Mendling*

DIJKMAN hat eine Methode entwickelt mit welcher Unterschiede zwischen Ereignisgesteuerten Prozessketten entdeckt werden können. Neben der Identifizierung von Unterschieden ermittelt DIJKMAN auch die Art der Unterschiede (vgl. Dijkman (2008), S. 261). Für diesen Zweck hinterlegt er die EPK mit der formalen Semantik von Transitionssystemen (vgl. Dijkman (2008), S. 264). Zur Entdeckung von Unterschieden nutzt er die Metrik<sup>42</sup> der ‚Gleichheit vollständiger Pfade‘.<sup>43</sup> Seine Methode liefert jedoch keine Maßzahlen, welche Aussagen zulassen, ob ein Modell ähnlicher ist als ein anderes. Ziel seiner Methode ist vielmehr das Entdecken und Eliminieren von Unterschieden zwischen Prozessmodellen, um beispielsweise eine Firmenfusion zu unterstützen (vgl. Dijkman (2008), S. 250 f.). Das Eliminieren von Unterschieden ist in der vorliegenden Arbeit nicht möglich, da die Referenzprozesse nicht immer leicht verändert werden können. Teilweise sind die Abläufe fest im Quellcode einer ERP-Software verankert, sodass diese nur mit großem Aufwand geändert werden können. Daher wird im Unterkapitel 4.4 die Idee der Betrachtung von Fällen auf die Suche von geeigneten Workflow-Netzen übertragen.

<sup>42</sup> Eine Metrik ist eine Abstandsfunktion (vgl. Dumke (2003), S. 176 f.).

<sup>43</sup> In seiner Arbeit verwendet DIJKMAN den Begriff ‚Completed Trace Equivalence‘.

In einer weiteren Arbeit haben VAN DONGEN, DIJKMAN und MENDLING auf Basis von kausalen Fußabdrücken<sup>44</sup> von Prozessen eine Methode entwickelt, welche eine Maßzahl ermittelt. Diese gibt die Ähnlichkeit zweier Prozessmodelle in einem Wertebereich von *null* bis *eins* wieder. Als Erstes wird dort der Fußabdruck für ein Referenzmodell und das Unternehmensmodell bestimmt. Anschließend wird mit dem Vektorraum-Modell, welches aus dem Bereich des Information Retrieval für den Dokumenten-Vektor bekannt ist, die Ähnlichkeit der Modelle berechnet<sup>45</sup>. Als Modellierungssprache wurden in der Arbeit der drei Autoren Ereignisgesteuerte Prozessketten verwendet (vgl. van Dongen et. al. (2008), S. 454, ff.).

### *Methoden von Priemer*

Um die Auswahl von Software effizienter zu gestalten, hat PRIEMER zwei Methoden entwickelt, mit welchen Prozessmodelle verglichen werden können. Der *Analysierende Vergleich* setzt voraus, dass das Referenzmodell eine Verfeinerung des Unternehmensmodells darstellt (vgl. Priemer (1995), S. 275 ff.). Diese Einschränkung ist hier jedoch zu restriktiv. Jedes Referenzmodell der ERP-Software, welches auch nur eine Funktion des Unternehmensmodells nicht enthält, würde bei der Referenzmodellsuche sofort eliminiert werden.

Das *Syntheseverfahren* geht von einer hochgradig flexiblen Software aus. Ziel des Verfahrens ist das Nachbauen des Unternehmensmodells mit den Funktionen des Referenzmodells bzw. den von der Software unterstützten Funktionen. Dabei wird versucht, die Funktionen des Unternehmensmodells mit Funktionen aus dem Referenzmodell abzubilden. Liegt keine direkte Entsprechung im Referenzmodell vor, wird eine Kombination von Funktionen des Referenzmodells gesucht, sodass diese Kombination äquivalent zur Funktion des Unternehmensmodells ist (vgl. Priemer (1995), S 284). Auch bei dieser Variante wird davon ausgegangen, dass die Funktionen des Unternehmensmodells als Verfeinerung im Referenzmodell vorliegen. Da ERP-Software selten eine völlig freie Kombination der enthaltenen Funktionen zu Prozessen erlaubt, ist diese Methode nur teilweise geeignet. Zudem benötigen beide Methoden häufige Nutzerinteraktionen, sodass diese nicht für eine automatisierte Realisierung der Evaluierungsphase des Initialen SAC genutzt werden können (vgl. Priemer (1995), S. 296).

---

<sup>44</sup> VAN DONGEN, DIJKMAN und MENDLING verwenden den Begriff ‚causal footprint‘.

<sup>45</sup> Eine grundlegende Beschreibung des Vektorraum-Modells kann in Bodendorf (2006), S. 115 ff. nachgeschlagen werden.

### *Methode von Baumgarten*

Eine Methode, welche sich auf Petri-Netze bezieht, wird in BAUMGARTEN vorgestellt (vgl. Baumgarten (1996), S. 54 ff.). Die dort beschriebene Methode ermöglicht Strukturvergleiche zwischen zwei Petri-Netzen. Dabei werden jedoch die Bezeichner der Stellen und Transitionen außer Acht gelassen. Als Ergebnis liefert diese Methode eine Aussage, ob die betrachteten Modelle strukturgleich sind oder nicht. Eine Aussage über den Grad der Ähnlichkeit der verglichenen Modelle ist nicht möglich.

### *Resümee*

Die meisten existierenden Methoden für den Vergleich von Prozessmodellen erlauben Aussagen über die Äquivalenz zweier Prozesse, nicht aber über deren Ähnlichkeit (vgl. van Dongen et. al. (2008), S. 453). Für die Suche eines geeigneten Referenzmodells ist eine Aussage über die Äquivalenz (gleich oder nicht gleich) unzureichend. Wird ein äquivalentes Referenzmodell zu einem Unternehmensmodell gefunden, so kann dieses für die Umsetzung in der ERP-Software genutzt werden. In der Mehrzahl der Fälle wird jedoch kein Referenzmodell existieren, welches das Unternehmensmodell exakt abbildet. Aus diesem Grund werden Methoden benötigt, welche es erlauben den Abstand zwischen zwei Workflow-Modellen zu ermitteln. Diese sind dann Grundlage für die Bestimmung des geeigneten Referenzmodells.

## **4.2 Allgemeine Beschreibung der Suchmethoden**

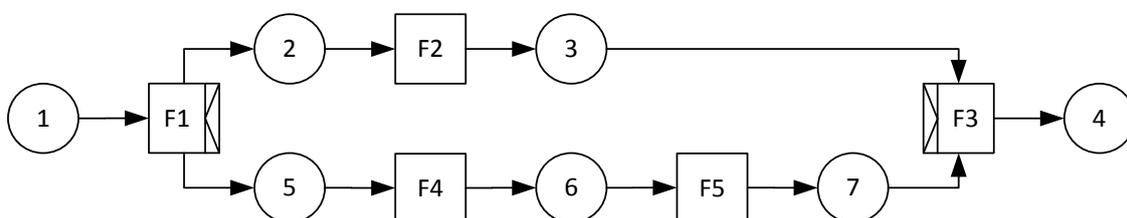
Auch wenn Methoden wie das Accelerated SAP bereits zu einer Verkürzung der Einführungszeit von ERP-Software geführt haben, ist ein automatisches Customizing bisher nicht möglich. So werden auch beim ASAP die konkreten Werte für die unternehmensspezifisch anzupassenden Parameter manuell eingegeben. Dies kann automatisiert werden, da die Werte im Rahmen des Business Blueprint bereits erfasst wurden (siehe Abschnitt 2.2.2). Ziel ist folglich die Integration der Phasen Business Blueprint und Realisierung. Ebenso muss für die Auswahl eines projektspezifischen Einführungsleitfadens ein Referenzmodell ausgewählt werden. In den nächsten Abschnitten wird gezeigt, wie dies automatisiert erfolgen kann.

Bei der *Fallbasierten Referenzmodellssuche* werden zunächst sämtliche vom Referenzmodell abgedeckten Fälle ermittelt. Dies sind alle möglichen Abläufe, welche mit dem Modell abgebildet werden. Anschließend werden diese mit den potenziellen Fällen des Unternehmensmodells verglichen. Letztendlich wird das Referenzmodell mit der größten Anzahl an abgedeckten Fällen ausgewählt.

Die *Abstandsmethode* ermittelt auf Basis von Abstandsmaßen Kennzahlen. Anhand dieser Kennzahlen wird dann das Referenzmodell mit dem geringsten Abstand zum Unternehmensmodell als Lösungsvorschlag ermittelt. Alternativ kann mit dieser Methode auch eine Rangfolge angegeben werden, um dem Entscheider die Möglichkeit zu geben das zweit- oder drittbeste Referenzmodell zu wählen. Die Ränge und Kennzahlenwerte können auch in einer Portfolio-Methode angewendet werden. So können neben dem Abstand bis zu zwei weitere Auswahlkriterien in den Entscheidungsprozess einfließen.<sup>46</sup>

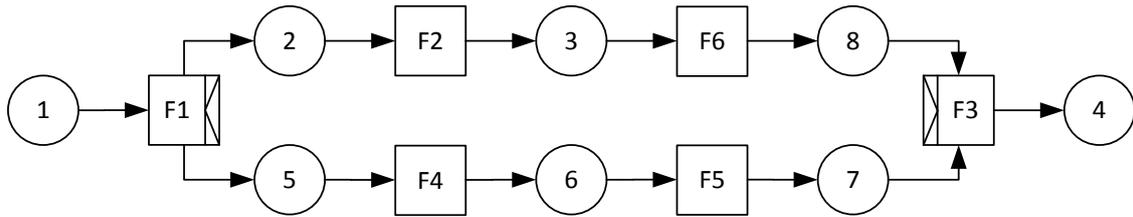
Die im Folgenden vorgestellten Verfahren können aber nicht nur automatisiert das beste Referenzmodell finden. Ein weiterer Vorteil dieser Methoden ist die Möglichkeit die Anpassungspunkte zu markieren. Somit ist es leichter die notwendigen Änderungen an der ERP-Software zu identifizieren. Dies kann das Self Adaptive Customizing Middleware System nutzen, um die notwendigen Anpassungen in der Implementierungsphase zu realisieren. Sollte eine automatisierte Anpassung aufgrund mangelnder Flexibilität der ERP-Software nicht möglich sein, kann den Entwicklern aufgezeigt werden, an welchen Stellen die ERP-Software modifiziert werden muss.

Zur Illustration der zwei entwickelten Verfahren werden die Modelle aus den folgenden Abbildungen verwendet. Das Unternehmensmodell stellt einen Prozess dar, der durch die ERP-Software abgebildet werden soll. Referenzmodell R1 enthält gegenüber dem Unternehmensmodell die zusätzliche Funktion F6, welche auf einem sequenziellen Pfad liegt und folglich nicht ausgelassen werden kann. Im Referenzmodell R2 liegt die Funktion F6 als Alternative zu Funktion F5 vor. Im dritten Referenzmodell hingegen wird die Funktion F6 zusätzlich parallel zu Funktion F5 ausgeführt. Das Referenzmodell R4 enthält ebenso viele Funktionen wie das Unternehmensmodell. Dort wurde jedoch die Funktion F2 durch die Funktion F6 ersetzt.

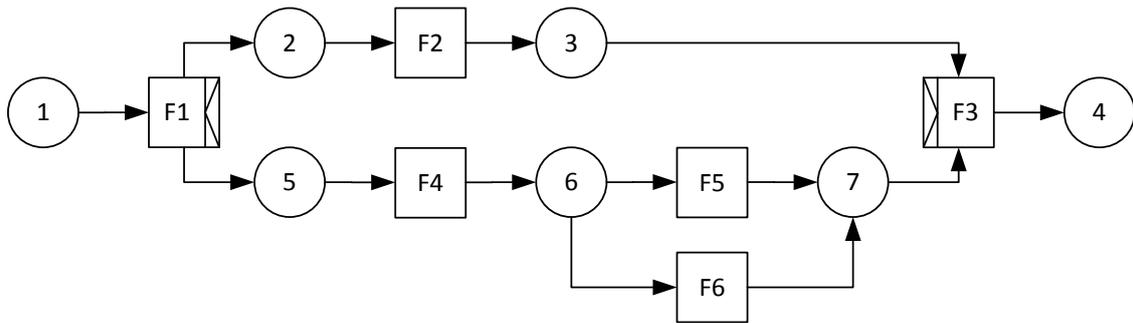


**Abb. 4.1:** Unternehmensmodell

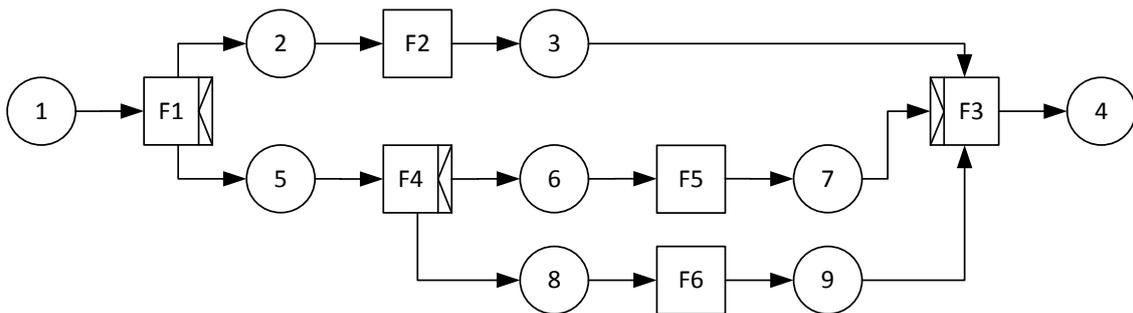
<sup>46</sup> In Analogie zur Boston Consulting Group Matrix können zwei Dimensionen auf den Achsen abgetragen werden und die dritte Dimension kann durch die Fläche des Kreises oder Quadrates im Portfolio visualisiert werden. Das Prinzip der Portfolio-Analyse kann in Olbrich (2006), S. 81 ff. nachgelesen werden.



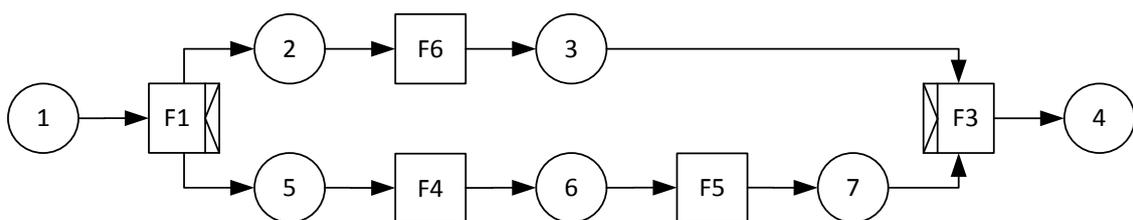
**Abb. 4.2:** Referenzmodell R1 mit zusätzlicher Funktion



**Abb. 4.3:** Referenzmodell R2 mit alternativer Funktion



**Abb. 4.4:** Referenzmodell R3 mit zusätzlicher paralleler Funktion



**Abb. 4.5:** Referenzmodell R4 mit ersetzter Funktion

Je nach Flexibilität der eingesetzten ERP-Software können auch die Organisationseinheiten und Geschäftsprozessobjekte im Unternehmensmodell spezifiziert werden. Bei der Umsetzung des Referenzmodells in die ERP-Software werden dann die entsprechenden Objekte in der Datenbank angelegt. Alternativ kann der verantwortlichen Person angezeigt werden, an welchen Stellen welche Änderungen vorgenommen werden müssen. Auf diese Weise können keine wichtigen Änderungen vergessen werden.

Voraussetzung für die folgenden Suchmethoden ist das Vorliegen von konvertierten Modellen gemäß der ersten Phase des Initialen SAC. Das heißt, die Modelle liegen als Workflow-Netze vor und die Bezeichnungen der Funktionen und Bedingungen wurden semantisch integriert (siehe Abschnitt 3.1.2). Gleiche Funktionsbezeichnungen signalisieren die gleiche Funktionalität<sup>47</sup> (Abwesenheit von Homonymen), ebenso sind gleichartige Funktionalitäten mit denselben Funktionsbezeichnungen versehen (Abwesenheit von Synonymen). Daher genügt es, in den unten vorgestellten Verfahren die Bezeichner von Funktionen zu vergleichen.

Die identifizierten Differenzen werden anschließend im Rahmen des Kontinuierlichen Customizing abgebaut. Das heißt, zunächst werden vermeintlich irrelevante Funktionen des Referenzmodells beibehalten. Aus der Analyse der tatsächlichen Systemnutzung wird dann die Abstimmung der ERP-Software verfeinert. Dies ist sinnvoll, da die Referenzmodelle oftmals auf ‚Best Practice‘-Daten beruhen und eine Funktion bei der Modellierung des Unternehmensmodells möglicherweise vergessen wurde. Dank dieser Vorgehensweise kann das Know-How aus den ‚Best Practice‘-Szenarien genutzt werden.

### 4.3 Kriterien für die Suche der Modelle

Sollen Objekte miteinander verglichen werden, müssen zunächst die für den Vergleich relevanten Kriterien festgelegt werden.

PRIEMER schlägt für die Auswahl von Software anhand von Modellen die Betrachtung von Daten, Funktionen und Prozessen (Reihenfolgebeziehung der Funktionen) vor. Als wichtigstes Kriterium nennt er die Abbildbarkeit der gewünschten Abläufe (vgl. Priemer (1995), S. 125 f.). DIJKMAN verwendet für die Suche nach Unterschieden zwischen Prozessen die Completed Trace Equivalence<sup>48</sup> und konzentriert sich somit auf die Funktionen und Reihenfolgebeziehungen, um Unterschiede zu entdecken. Andere Prozesselemente wie Organisationseinheiten oder das prozessprägende Objekt werden nicht berücksichtigt (vgl. Dijkman (2008), S. 265 f.). Für das Self Adaptive Customizing nennt KASSEM Funktionen,<sup>49</sup> deren Reihenfolge und die abhängigen Workflow-Bezugsobjekte als relevant. Letztere werden bereits im Rahmen der Konvertierungsphase berücksichtigt (vgl. Kassem (2007), S. 64 ff.).

---

<sup>47</sup> Gleiche Funktionalität meint hier, dass zwei Funktionen bei Vorliegen gleicher Inputdaten die gleichen Outputdaten erzeugen.

<sup>48</sup> Completed Trace Equivalence liegt vor, wenn die Mengen aller vollständigen Pfade (Fälle) zweier Prozesse gleich sind.

<sup>49</sup> Bei KASSEM werden diese als auch als Aufgaben oder Aktivitäten bezeichnet.

Alle oben genannten Arbeiten identifizieren die Funktionen und die Reihenfolgebeziehung der Funktionen als wichtiges Merkmal eines Prozesses.<sup>50</sup> Daher bilden diese in der vorliegenden Arbeit die Kriterien für den Vergleich von Prozessen. Die darüber hinaus mit den Funktionen in Beziehung stehenden Workflow-Bezugsobjekte werden nicht explizit betrachtet. Denn die Workflow-Bezugsobjekte werden bereits in der Konvertierungsphase des Initialen SAC einbezogen.<sup>51</sup>

#### **4.4 Fallbasierte Referenzmodellsuche**

Um die Fallbasierte Referenzmodellsuche zu realisieren, kann die Idee von DIJKMAN auf Workflow-Netze angepasst werden. Hierfür werden die vollständigen Pfade von Unternehmensmodellen und Referenzmodellen verglichen.

Im nächsten Abschnitt wird die verwendete Metrik vorgestellt. Anschließend wird gezeigt, wie die Anzahl der gleichen Fälle bestimmt werden kann. Auf Basis dieses Fallabstandes wird dann das am besten geeignete Referenzmodell ausgewählt. Dies wird an einem Beispiel demonstriert. Am Ende dieses Unterkapitels werden die Vor- und Nachteile der Fallbasierten Referenzmodellsuche diskutiert.

##### **4.4.1 Metrik Fallgleichheit**

Grundlage der Fallbasierten Referenzmodellsuche ist die ‚Gleichheit vollständiger Pfade‘. Dabei ist ein Pfad eine mögliche Reihenfolge von Funktionsausführungen in einem Workflow-Netz. Ein vollständiger Pfad bezeichnet einen Pfad, welcher von einem möglichen Anfangszustand eines Workflow-Netzes zu einem Endzustand führt. Im Folgenden wird für einen vollständigen, serialisierten Pfad der Begriff ‚Fall‘ verwendet. Für die Entwicklung der Metrik Fallgleichheit wurde auf die Erreichbarkeitsanalyse zurückgegriffen. Ausgehend von der Erreichbarkeitsanalyse wurde ein Verfahren zur Generierung der Fälle eines Workflownetzes entwickelt. Die generierten Fälle werden dann für die Bestimmung eines Abstandes genutzt.

Um alle Fälle eines Prozessmodells zu erhalten, müssen die potenziellen Funktionsfolgen aus dem Modell extrahiert werden. In Bezug auf Workflow-Netze sind dies die Reihenfolgen der Transitionen. Wie in Abschnitt 2.3.2 erläutert, wird die Ausführung einer Transition auch als Schalten bezeichnet. Um alle Fälle eines Prozessmodells zu

---

<sup>50</sup> VAN DER AALST und VAN HEE sind ebenfalls der Meinung, dass der Kontrollfluss die wichtigste Perspektive beim Workflow-Management ist (vgl. van der Aalst/van Hee (2004), S. 271).

<sup>51</sup> Für die Beschreibung der Konvertierungsphase siehe Abschnitt 3.1.2.

identifizieren, können daher die Schaltfolgen  $w$  eines Workflow-Netzes genutzt werden. Ein Fall ist somit eine vollständige Schaltfolge  $w^v$ , wenn gilt:

$$M_0[w^v] M',$$

$M_0$  die Anfangsmarkierung eines Workflow-Netzes und  $M'$  die Endmarkierung eines Workflow-Netzes ist. Also:

$$M_0(s) = \begin{cases} 1, & \text{wenn } s \text{ die Input - Stelle ist} \\ 0, & \text{sonst} \end{cases}$$

$$M'(s) = \begin{cases} 1, & \text{wenn } s \text{ die Output - Stelle ist} \\ 0, & \text{sonst.} \end{cases}$$

Die Fälle bzw. vollständigen Schaltfolgen  $w^v$  können mit der Erreichbarkeitsanalyse<sup>52</sup> ermittelt werden. Hierfür werden alle Schaltmöglichkeiten systematisch geprüft. Anschließend werden die Fälle gebildet. In Tab. 4.1 ist die Erreichbarkeitsanalyse für das Unternehmensmodell aus Abb. 4.1 dargestellt.

**Tab. 4.1:** Erreichbarkeitsanalyse Unternehmensmodell

Markierung	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	Schaltungen
$M_0$	1	0	0	0	0	0	0	$F1 \rightarrow M_1$
$M_1$	0	1	0	0	1	0	0	$F2 \rightarrow M_2$ $F4 \rightarrow M_3$
$M_2$	0	0	1	0	1	0	0	$F4 \rightarrow M_4$
$M_3$	0	1	0	0	0	1	0	$F2 \rightarrow M_4$ $F5 \rightarrow M_5$
$M_4$	0	0	1	0	0	1	0	$F5 \rightarrow M_6$
$M_5$	0	1	0	0	0	0	1	$F2 \rightarrow M_6$
$M_6$	0	0	1	0	0	0	1	$F3 \rightarrow M_7$
$M_7$	0	0	0	1	0	0	0	

Für die Ermittlung der Fälle eines Workflow-Netzes werden die Schaltungen der Erreichbarkeitsanalyse genutzt. Ausgehend von der Anfangsmarkierung  $M_0$  werden die geschalteten Transitionen der Schaltfolge  $w_j$  hinzugefügt, bis die Endmarkierung erreicht ist. Sind in einer Markierung mehr als eine Transition aktiviert, so wird für jede aktivierte Transition eine Schaltfolge erstellt. Dabei enthält jede erstellte Schaltfolge die bisherigen Schaltungen sowie eine der aktivierten Transitionen. Die Vorgehensweise wird am Beispiel des Unternehmensmodells erläutert:

<sup>52</sup> Zur Erreichbarkeitsanalyse siehe auch Baumgarten (1996), S. 84 f.

Schritt 1:

Zu Beginn existiert nur die leere Schaltfolge  $w_1$ . In der Anfangsmarkierung ist nur eine Transition aktiviert, daher wird diese der Schaltfolge hinzugefügt.

$$w_1 = F1$$

Schritt 2:

In  $M_1$  sind sowohl Transition  $F2$  als auch Transition  $F4$  aktiviert. Daher wird eine neue Schaltfolge  $w_2$  erstellt, welche die bisherigen Schaltungen ( $F1$ ) enthält.

$$w_1 = F1, F2$$

$$w_2 = F1, F4$$

Schritt 3:

Für die Schaltfolge  $w_2$  sind wiederum zwei Transitionen aktiviert. Somit werden aus dieser die zwei Schaltfolgen  $w_2$  und  $w_3$  erstellt.

$$w_1 = F1, F2, F4$$

$$w_2 = F1, F4, F2$$

$$w_3 = F1, F4, F5$$

Schritt 4:

$$w_1 = F1, F2, F4, F5$$

$$w_2 = F1, F4, F2, F5$$

$$w_3 = F1, F4, F5, F2$$

Schritt 5:

$$w_1 = F1, F2, F4, F5, F3 = w_{U,1}^v$$

$$w_2 = F1, F4, F2, F5, F3 = w_{U,2}^v$$

$$w_3 = F1, F4, F5, F2, F3 = w_{U,3}^v$$

Im fünften Schritt erreichen alle Schaltfolgen die Endmarkierung des Unternehmensmodells und entsprechen nun gemäß obiger Definition vollständigen Schaltfolgen bzw. Fällen.

Aus dem obigen Beispiel lässt sich erkennen, dass das Unternehmensmodell die drei Fälle  $w_{U,1}^v$ ,  $w_{U,2}^v$  und  $w_{U,3}^v$  enthält. Als Maßzahl für den Abstand der Referenzmodelle zum Unternehmensmodell wird die *Anzahl der gleichen Fälle (v)* von Unternehmens- und Referenzmodell genutzt.

#### 4.4.2 Vergleich der potenziellen Fälle

Um das beste Referenzmodell in Bezug auf das Unternehmensmodell zu finden, werden zunächst sämtliche Fälle des Unternehmensmodells erstellt. Anschließend werden sequenziell alle Mengen der Fälle der Referenzmodelle ermittelt.

Die Menge der Fälle des Unternehmensmodells wird mit  $V^U$  bezeichnet:

$$V^U = \{w_j^v \mid j \in 0..n, n \in \mathbb{N}, n \text{ ist Anzahl der Fälle des Unternehmensmodells}\}$$

Die Menge der Fälle eines Referenzmodells  $i$  wird mit  $V_i^R$  bezeichnet:

$$V_i^R = \{w_{i,j}^v \mid j \in 0..n, n \in \mathbb{N}, n \text{ ist Anzahl der Fälle des Referenzmodells } i\}$$

Liegen alle Fälle der Unternehmens- und Referenzmodelle vor, wird die Menge  $V^U$  paarweise mit den Mengen  $V_i^R$  verglichen. Dabei wird die Anzahl der durch das jeweilige Referenzmodell abgedeckten Fälle ( $v_i$ ) ermittelt:

$$v_i = |V^U \cap V_i^R|$$

Je größer  $v_i$  ist, desto mehr Fälle können vom entsprechenden Referenzmodell abgedeckt werden. Folglich wird anschließend das Referenzmodell  $R^l$  mit dem größten Wert für  $v_i$  ermittelt und als Basis für die Implementierungsphase vorgeschlagen:

$$R^l = \{Ri \mid v_i = \max(v_1, \dots, v_n), n = \text{Anzahl der Referenzmodelle}\}$$

#### 4.4.3 Exemplarische Anwendung

In diesem Abschnitt wird die Methode der Fallbasierten Referenzmodellsuche auf das Unternehmensmodell aus Abb. 4.1 angewendet. Als Referenzmodelle werden die Workflow-Netze aus den Abb. 4.2 bis Abb. 4.5 verwendet. Die Fälle des Unternehmensmodells wurden in Abschnitt 4.4.1 ermittelt. Im Folgenden werden die Fälle der

Referenzmodelle aufgelistet. Die Bestimmung der Fälle mittels Erreichbarkeitsanalyse kann im Anhang A nachverfolgt werden.

Für Referenzmodell R1 ergeben sich die Fälle:

- $w_{1,1}^v = F1, F2, F4, F5, F6, F3$
- $w_{1,2}^v = F1, F4, F2, F5, F6, F3$
- $w_{1,3}^v = F1, F2, F6, F4, F5, F3$
- $w_{1,4}^v = F1, F4, F5, F2, F6, F3$
- $w_{1,5}^v = F1, F2, F4, F6, F5, F3$
- $w_{1,6}^v = F1, F4, F2, F6, F5, F3$ .

Wie mit einem paarweisen Vergleich der Fälle von Referenzmodell R1 und dem Unternehmensmodell festzustellen ist, besitzen diese keine gemeinsamen vollständigen Pfade. Somit ist  $v_1 = |V^U \cap V_1^R| = 0$ .

Für Referenzmodell R2 ergeben sich die Fälle:

- $w_{2,1}^v = F1, F2, F4, F5, F3$
- $w_{2,2}^v = F1, F4, F2, F5, F3$
- $w_{2,3}^v = F1, F2, F4, F6, F3$
- $w_{2,4}^v = F1, F4, F2, F6, F3$ .

Beim Vergleich von Referenzmodell R2 und dem Unternehmensmodell wird festgestellt, dass die Fälle  $w_{2,1}^v = w_{U,1}^v$  und  $w_{2,2}^v = w_{U,2}^v$  gleich sind. Das Referenzmodell R2 besitzt somit zwei gemeinsame Fälle mit dem Unternehmensmodell ( $v_2 = |V^U \cap V_2^R| = 2$ ).

Für Referenzmodell R3 ergeben sich die Fälle:

- $w_{3,1}^v = F1, F2, F4, F5, F6, F3$
- $w_{3,2}^v = F1, F4, F2, F5, F6, F3$
- $w_{3,3}^v = F1, F4, F5, F2, F6, F3$
- $w_{3,4}^v = F1, F4, F6, F2, F5, F3$

- $w_{3,5}^v = F1, F2, F4, F6, F5, F3$
- $w_{3,6}^v = F1, F4, F2, F6, F5, F3$
- $w_{3,7}^v = F1, F4, F5, F6, F2, F3$
- $w_{3,8}^v = F1, F4, F6, F5, F2, F3$ .

Da sämtliche Fälle die Länge sechs besitzen, alle Fälle des Unternehmensmodells hingegen die Länge fünf, kann schnell festgestellt werden, dass keine gemeinsamen Fälle existieren ( $v_3 = |V^U \cap V_3^R| = 0$ ).

Für Referenzmodell R4 ergeben sich die Fälle:

- $w_{4,1}^v = F1, F4, F5, F6, F3$
- $w_{4,2}^v = F1, F6, F4, F5, F3$
- $w_{4,3}^v = F1, F4, F6, F5, F3$ .

Auch beim Vergleich der Fälle von Referenzmodell R4 mit den Fällen des Unternehmensmodells werden keine gemeinsamen Fälle gefunden. Die Anzahl  $v_4$  beträgt folglich ebenfalls  $v_4 = |V^U \cap V_4^R| = 0$ .

Wird nun das Maximum aus den ermittelten Werten für  $v_i$  gebildet, ergibt sich Referenzmodell R2 als Lösung für die Suche des geeigneten Referenzmodells  $R^l$ :

$$\max(v_1, v_2, v_3, v_4) = \max(0, 2, 0, 0) = 2 \Rightarrow R^l = R2, \text{ denn } v_2 = 2.$$

#### 4.4.4 Bewertung der Fallbasierten Referenzmodellsuche

Anhand des Beispiels im vorherigen Abschnitt wurde gezeigt, dass es möglich ist, unter Nutzung der Metrik ‚gleiche gemeinsame Fälle‘ automatisiert ein geeignetes Referenzmodell zu finden. Hierbei werden als Kriterien sowohl die in den Modellen abgebildeten Funktionen, als auch die Reihenfolgebeziehung der Funktionen berücksichtigt.

Auffällig ist, dass viele Fälle der Referenzmodelle herausfallen, welche nicht die gleiche Länge wie die Fälle des Unternehmensmodells besitzen. Sobald eine Funktion auf einem Pfad hinzukommt oder entfernt wird, welcher nicht alternativ vorliegt (sequenzieller Pfad, oder UND-Verknüpfung) verlängert oder verkürzt sich die Schalt-

folge  $w_{i,j}^v$  für diesen Pfad. Da dieser Pfad Bestandteil aller Schaltfolgen des Referenzmodells  $i$  ist, existiert in diesem Fall keine Schaltfolge, welche die gleiche Länge besitzt wie die Schaltfolgen des Unternehmensmodells  $w_{i,U}^v$ . Folglich beträgt die Anzahl gleicher Fälle mit diesem Referenzmodell  $v^i = 0$ . Die vorgestellte Methode kann daher beispielsweise keine Aussagen treffen, ob ein Referenzmodell mit zwei zusätzlichen Funktionen besser geeignet ist als ein Referenzmodell mit fünf zusätzlichen Funktionen. Denn für beide Referenzmodelle würde die Metrik den Wert *null* liefern. Zudem ist das systematische Prüfen und Erstellen der Fälle aufwendig.

Aus diesen Gründen wurde eine zweite Methode entwickelt, mit welcher ein geeignetes Referenzmodell gesucht werden kann. Mit der Abstandsmethode können die Unterschiede zwischen den Modellen detaillierter betrachtet werden.

#### 4.5 Abstandsmethode

In den nächsten Abschnitten wird die Abstandsmethode näher erläutert. Sie besteht aus den drei Schritten:

1. Maßzahl ermitteln
2. Referenzmodell auswählen
3. Unterschiede hervorheben

Zuerst werden gemäß den unten beschriebenen Metriken die Maßzahlen ermittelt. Die Metriken beziehen sich auch hier immer auf einen paarweisen Vergleich von Unternehmensmodell und einem Referenzmodell. Im darauffolgenden Schritt wird dann auf Basis der vorliegenden Maßzahlen ein Referenzprozess für die Implementierung ausgewählt. Im dritten Schritt werden die Unterschiede zwischen dem gewählten Referenzmodell und dem Unternehmensmodell grafisch sichtbar gemacht.

Entsprechend dieser Reihenfolge werden zunächst die verwendeten Metriken vorgestellt. Dann wird beschrieben, wie das Referenzmodell ausgewählt wird und das Vorgehen an einem Beispiel verdeutlicht. Abschließend wird die Methode kritisch betrachtet.

#### 4.5.1 Metriken Funktionsabstand und Funktionsdifferenz

Da Funktionen ein wesentliches Merkmal eines betrieblichen Prozesses sind, wurden zunächst Metriken bezüglich der Funktionen entwickelt. Der *Funktionsabstand*  $R^-$  gibt Auskunft darüber, wie viele Funktionen des Unternehmensmodells nicht im Referenzmodell abgebildet sind. Fehlen Funktionen im Referenzmodell, so können Teile des Prozesses nicht in der ERP-Software abgebildet werden. Formal kann  $R^-$  als Betrag der Differenzmenge

$$R^- = |T^U \setminus T^R|$$

bestimmt werden.  $T^U$  ist hierbei die Menge der Transitionen des Unternehmensmodells und  $T^R$  die Menge der Transitionen des betrachteten Referenzmodells.

Mit der *Funktionsdifferenz* wird ermittelt, wie viele abweichende Funktionen in den zwei jeweils betrachteten Modellen existieren. Dafür wird zuerst der Funktionsabstand  $R^-$  ermittelt. Anschließend wird die Anzahl der Referenzfunktionen  $R^+$  bestimmt, welche nicht Bestandteil des Unternehmensmodells sind. Enthält das Referenzmodell mehr Funktionen als benötigt, kann dies zu überflüssigen Aufgaben führen. Aus diesem Grund wird die Funktionsdifferenz  $F^{diff}$  ermittelt. Diese ergibt sich als Summe aus  $R^-$  und  $R^+$ :

$$F^{diff} = R^- + R^+.$$

Alternativ kann die Funktionsdifferenz  $F^{diff}$  auch als Betrag der Differenzmenge berechnet werden:

$$F^{diff} = |(T^U \cup T^R) \setminus (T^U \cap T^R)|.$$

#### 4.5.2 Metrik Inzidenzabstand

Sowohl Funktionsabstand  $R^-$  als auch Funktionsdifferenz  $F^{diff}$  berücksichtigen nicht die Reihenfolgebeziehung der Funktionen. Aus diesem Grunde wurde eine weitere Metrik entwickelt. Der *Inzidenzabstand*  $I$  gibt die Anzahl der zwischen zwei Modellen nicht identischen gerichteten Kanten an. Außerdem werden implizit auch die Differenzen zwischen den Funktionen berücksichtigt. Für die Entwicklung der Metrik Inzidenzabstand wurde die Inzidenzmatrix eines Workflow-Netzes genutzt.

Die Inzidenzmatrix  $C$  eines Workflow-Netzes enthält in den Zeilen die Stellen  $s_i$  und in den Spalten die Transitionen  $t_j$  (vgl. Baumgarten (1996), S. 90):

$$C_{(i,j)} := \begin{cases} W(t_j, s_i) & \text{wenn } (t_j, s_i) \in F \setminus F^{-1} \\ -W(s_i, t_j) & \text{wenn } (s_i, t_j) \in F \setminus F^{-1} \\ W(t_j, s_i) - W(s_i, t_j) & \text{wenn } (t_j, s_i) \in F \cap F^{-1} \\ 0 & \text{sonst} \end{cases}$$

Für die Definition der Flussrelation  $F$  siehe Abschnitt 2.3.2. Das Gewicht  $W$  einer Kante ist per Definition bei Workflow-Netzen immer eins ist. Die Inzidenzmatrizen des Unternehmensmodells und des Referenzmodells R1 sind unten dargestellt. Die Inzidenzmatrix des Unternehmensmodells  $C^U$  wurde hier bereits um eine Dummy-Spalte  $\delta_{F6}$  und eine Dummy-Zeile  $\delta_{s8}$  ergänzt. Diese sind Platzhalter für die Funktion<sup>53</sup> F6 und die Stelle  $s_8$  des Referenzmodells und werden im nächsten Schritt benötigt.

$$C^U = \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ \delta_{s8} \end{array} \begin{array}{c} F1 \\ F2 \\ F3 \\ F4 \\ F5 \\ \delta_{F6} \end{array} \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C^{R1} = \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{array} \begin{array}{c} F1 \\ F2 \\ F3 \\ F4 \\ F5 \\ F6 \end{array} \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}$$

Der Spaltenvektor  $\overrightarrow{F1}$  der erweiterten Inzidenzmatrix  $C^U$  gibt an, dass bei Ausführung von Funktion F1 aus Stelle  $s_1$  eine Marke verbraucht und in den Stellen  $s_2$  und  $s_5$  Marken erzeugt werden. Die übrigen Stellen des Unternehmensmodells stehen mit der Funktion F1 nicht in Verbindung.

Sind die Inzidenzmatrizen vorhanden, kann aus der Unternehmensmatrix und der Referenzmatrix die Abstandsmatrix A bestimmt werden. Da die Differenz nur von Matrizen mit gleicher Zeilen- und Spaltenanzahl berechnet werden kann, wurden die Dummyvariablen  $\delta$  eingeführt. In die Dummystellen und -transitionen gehen weder Kanten ein noch gehen welche heraus, sodass die korrespondierenden Plätze in der

<sup>53</sup> Wie in Abschnitt 2.3.3 erläutert, stellen die Transitionen eines Workflow-Netzes die betrieblichen Funktionen dar. Um den Bezug zur Betriebswirtschaft deutlich zu machen, wird hier vorrangig die Bezeichnung Funktion verwendet.

Matrix den Wert 0 erhalten haben. Für das Unternehmensmodell und Referenzmodell R1 ergibt sich folgende Abstandsmatrix:

$$A^1 = C^{R1} - C^U = \begin{matrix} & \text{F1} & \text{F2} & \text{F3} & \text{F4} & \text{F5} & \text{F6} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{matrix} & \left( \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{array} \right) \end{matrix}$$

Nun werden die Beträge der einzelnen Zellen zu Zeilensummen addiert und anschließend die Zeilensummen zu einem Gesamtabstand summiert, dem Inzidenzabstand  $I$ . Dieser gibt die Anzahl der unterschiedlichen Kanten zwischen den zwei Modellen an:

$$I = \sum_{i=1}^n \sum_{j=1}^m |A_{i,j}|, n - \text{Anzahl der Zeilen}, m - \text{Anzahl der Spalten}$$

Weiterhin gibt die Anzahl der Dummy-Spalten in den Inzidenzmatrizen die Funktionsdifferenz  $F^{diff}$  wieder.

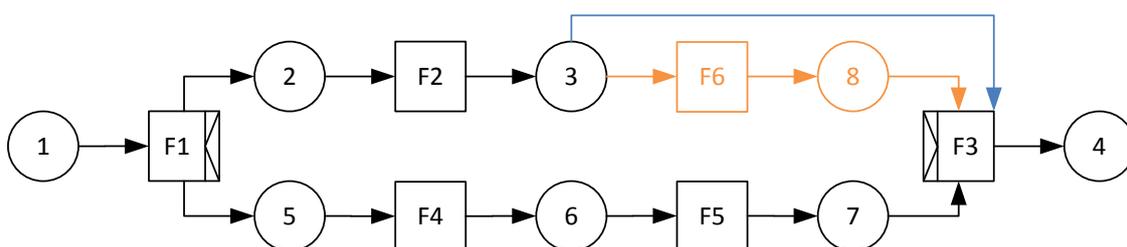
### 4.5.3 Referenzmodellssuche mittels Abstandsmethode

Erlaubt die ERP-Software eine flexible Anpassung der Abläufe, genügt es den Funktionsabstand für die Auswahl eines Referenzmodells heranzuziehen. Ist jedoch das Entfernen von nicht benötigten Funktionen des Referenzmodells mit Aufwand verbunden, so sind nicht nur die im Referenzmodell fehlenden Funktionen von Interesse, sondern auch jene, welche zu viel sind. In diesem Fall sollte die Funktionsdifferenz genutzt werden. Ist darüber hinaus die Art der Unterschiede in den Abläufen relevant, sollte der Inzidenzabstand in die Referenzmodellssuche einbezogen werden. Eine zusätzliche ODER-Verknüpfung könnte je nach ERP-Software beispielsweise leichter angepasst werden als eine UND-Verknüpfung.

Je nach Flexibilität der ERP-Software werden zunächst die entsprechenden Maßzahlen bestimmt. Dabei wird die Bestimmung der Maßzahlen paarweise für das Unternehmensmodell und ein Referenzmodell wiederholt, bis die Maßzahlen für alle Referenzmodelle ermittelt wurden.

Im Anschluss wird dann das Referenzmodell mit dem geringsten Funktionsabstand, der geringsten Funktionsdifferenz oder dem kleinsten Inzidenzabstand gewählt. Dies sollte dann aufgrund der Ähnlichkeiten zum Unternehmensmodell Grundlage für die Implementierungsphase sein.

Letztendlich können bei Nutzung des Inzidenzabstandes die Unterschiede zwischen dem gewählten Referenzmodell und dem Unternehmensmodell grafisch visualisiert werden. In Abb. 4.6 ist das Unternehmensmodell mit den Differenzen zum Referenzmodell R1 dargestellt. Zusätzliche Elemente sind orange, fehlende Elemente blau hervorgehoben.



**Abb. 4.6:** Unternehmensmodell mit Hervorhebung der Differenzen

Diese Darstellung kann im Rahmen des Kontinuierlichen Customizing Entscheider bei der Wahl unterstützen, ob die Prozesse des Unternehmens an die ERP-Software angepasst oder die ERP-Software verändert werden soll.

Um die Differenzen zwischen Unternehmensmodell und Referenzmodell hervorheben zu können, werden die Inzidenzmatrizen und die Abstandsmatrix interpretiert. Die Dummyvariablen in der Inzidenzmatrix des Unternehmensmodells entsprechen zusätzlichen Funktionen oder Stellen, die Dummyvariablen in der Inzidenzmatrix des Referenzmodells hingegen fehlenden Funktionen oder Stellen. Aus der Abstandsmatrix lassen sich die Unterschiede der Flussrelationen ablesen. Anhand des Spaltenvektors  $\vec{F3}$  der Abstandsmatrix  $A^1$  lässt sich erkennen, dass beim Ausführen von Funktion F3 im Referenzmodell R1 eine zusätzliche Marke in Stelle  $s_3$  vorhanden ist. Somit fehlt die gerichtete Kante von Stelle  $s_3$  zu Funktion F3. Weiterhin wird in Stelle  $s_8$  eine zusätzliche Marke verbraucht, was eine ergänzende Kante von Stelle  $s_8$  nach Funktion F3 impliziert. Wird der Spaltenvektor  $\vec{F6}$  betrachtet, muss berücksichtigt werden, dass dieser in der Inzidenzmatrix des Unternehmensmodells eine Dummy-Spalte darstellt. Daher wird der Wert  $-1$  in Zeile 3 als ein zusätzlicher Verbrauch einer Marke in Stelle  $s_3$  interpretiert und entspricht somit einer ergänzenden Kante von Stelle  $s_3$  zu Funktion F6. Weiterhin wird beim Ausführen von Funktion F6 eine zusätzliche Marke in Stelle  $s_8$  erzeugt, was einer ergänzenden Kante von Funktion F6 zu Stelle  $s_8$  entspricht.

#### 4.5.4 Exemplarische Anwendung

Die Referenzmodelle R1 bis R3 enthalten alle Funktionen des Unternehmensmodells sowie zusätzlich die Funktion F6. Daher ergeben sich folgende Werte für Funktionsabstand und Funktionsdifferenz:

- Funktionsabstand  $R^- = 0$
- Funktionsdifferenz  $F^{diff} = 1, (R^+ = 1)$ .

Referenzmodell R4 enthält fast alle Funktionen des Unternehmensmodells, lediglich die Funktion F2 fehlt. Diese wurde jedoch durch die zusätzliche Funktion F6 ersetzt, sodass für Funktionsabstand und Funktionsdifferenz die Werte

- $R^- = 1$
- $F^{diff} = 2, (R^+ = 1)$

ermittelt werden.

In diesem Fall wäre ein Entscheider oder das Self Adaptive Customizing Middleware System zwischen den Referenzmodellen R1, R2 und R3 indifferent. Referenzmodell R4 würde hingegen nicht weiter berücksichtigt, da der Abstand hier größer ist. Dies ist auch aus semantischer Perspektive sinnvoll, denn bei Nutzung von Referenzmodell R4 müsste sowohl Funktion F6 in der ERP-Software ergänzt als auch Funktion F2 möglichst aus dem Prozessablauf herausgelöst werden. Der Aufwand zur Anpassung dieses Modells wäre folglich größer als bei den Referenzmodellen R1, R2 und R3, bei welchen nur eine Funktion angepasst werden muss. Sind die Abläufe des Referenzmodells flexibel in der ERP-Software anpassbar, so kann nun eines der drei Referenzmodelle für die Implementierung genutzt werden. Andernfalls müssen die Inzidenzabstände hinzugezogen werden.

Für die Modelle R1-R4 können die Abstandsmatrizen

$$\bullet A^1 = C^{R1} - C^U = \begin{matrix} & \begin{matrix} F1 & F2 & F3 & F4 & F5 & F6 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{matrix} & \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{matrix} \right\} \end{matrix}$$

$$\bullet A^2 = C^{R2} - C^U = \begin{matrix} & \begin{matrix} \text{F1} & \text{F2} & \text{F3} & \text{F4} & \text{F5} & \text{F6} \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{matrix} & \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right\} \end{matrix}$$

$$\bullet A^3 = C^{R3} - C^U = \begin{matrix} & \begin{matrix} \text{F1} & \text{F2} & \text{F3} & \text{F4} & \text{F5} & \text{F6} \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \end{matrix} & \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{matrix} \right\} \end{matrix}$$

$$\bullet \text{ und } A^4 = C^{R4} - C^U = \begin{matrix} & \begin{matrix} \text{F1} & \delta_{\text{F2}} & \text{F3} & \text{F4} & \text{F5} & \text{F6} \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{matrix} & \left\{ \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right\} \end{matrix}$$

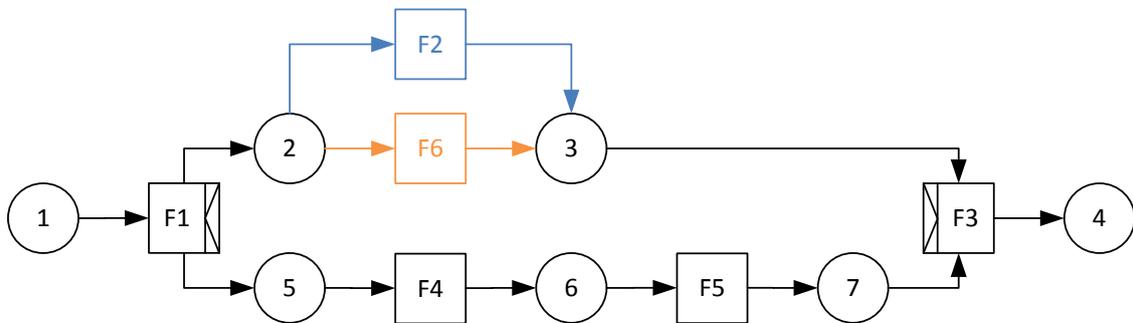
ermittelt werden. Die Inzidenzmatrix des Unternehmensmodells und des Referenzmodells R1 sind in Abschnitt 4.5.2 abgebildet. Die jeweiligen Inzidenzmatrizen der Referenzmodelle R2, R3 und R4 sind im Anhang B dargestellt.

Die Inzidenzabstände besitzen somit die Werte:

- $I_{R1} = 4$
- $I_{R2} = 2$
- $I_{R3} = 4$
- und  $I_{R4} = 4$ .

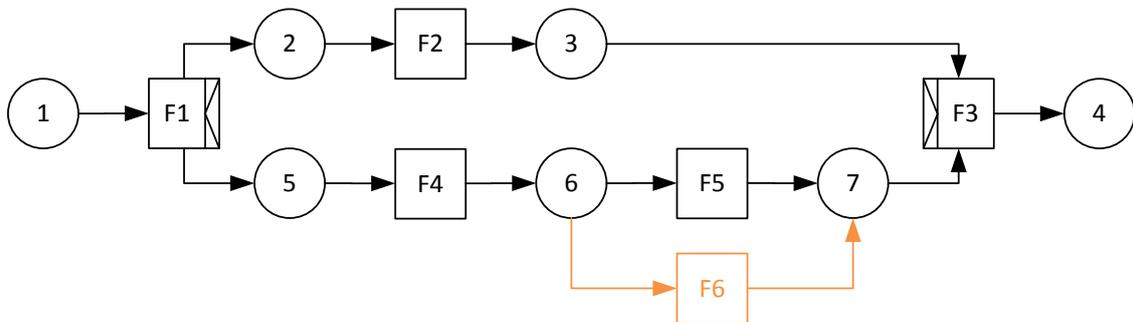
Das Ergebnis der Inzidenzabstände ist interessant, da im Referenzmodell R4 lediglich eine Funktion ausgetauscht, aber keine Kanten geändert wurden und der Inzidenzabstand folglich *null* sein müsste. Bei genauerer Betrachtung fällt jedoch auf, dass wie

in Abb. 4.7 zu sehen, auch die Kanten geändert wurden. Hier wurde beispielsweise die Kante von Stelle  $s_2$  zu Funktion F2 entfernt und die Kante von Stelle  $s_2$  zu Funktion F6 hinzugefügt. Daher ist der Inzidenzabstand für Referenzmodell R4 nicht *null* sondern *vier*.



**Abb. 4.7:** Unternehmensmodell mit Differenzen zu Referenzmodell R4

Da für Referenzmodell R2 der geringste Inzidenzabstand ermittelt wurde, sollte dieses Modell Grundlage für die Implementierung der ERP-Software sein. Aus der Abstandsmatrix lässt sich ablesen, dass im Referenzmodell R2 die Ausführung der Funktion F6 eine zusätzliche Marke aus  $s_6$  verbraucht (gerichtete Kante von  $s_6$  zu F6) und eine zusätzliche Marke in  $s_7$  erzeugt (gerichtete Kante von F6 zu  $s_7$ ). Abb. 4.8 zeigt die Abweichung grafisch, hierbei sind die gegenüber dem Unternehmensmodell zusätzlichen Elemente orange hervorgehoben.



**Abb. 4.8:** Unternehmensmodell mit Differenzen zu Referenzmodell R2

#### 4.5.5 Bewertung der Abstandsmethode

Enthalten zwei Referenzmodelle alle Funktionen des Unternehmensmodells, so besitzen die Referenzmodelle den gleichen Funktionsabstand  $R^- = 0$ . Dabei ist beispielsweise nicht relevant, ob das erste Referenzmodell eine zusätzliche Funktion und das zweite Referenzmodell vier zusätzliche Funktionen zum Unternehmensmodell enthält. Können die zusätzlichen Funktionen im Referenzmodell leicht entfernt werden, entspricht diese Indifferenz zwischen beiden Referenzmodellen auch dem Aufwand für deren

Implementierung. Ist dies nicht der Fall, muss die Funktionsdifferenz hinzugezogen werden.

Liegt eine zusätzliche Funktion in zwei Referenzmodellen als Alternative oder parallel zu einer existierenden Funktion des Unternehmensmodells vor, sind Funktionsabstand und Funktionsdifferenz beider Referenzmodelle identisch (jeweils Funktionsabstand  $R^- = 0$  und Funktionsdifferenz  $F^{diff} = 1$ ). Auch ein objektiver Entscheider würde zu dem Ergebnis kommen, dass beide Referenzmodelle hinsichtlich ihrer Eignung äquivalent sind. Vorausgesetzt dieser geht davon aus, dass die Abläufe im Referenzmodell leicht geändert werden können.

Ist jedoch neben dem Hinzufügen oder Entfernen von neuer Funktionalität auch das Ändern der Abläufe des Referenzmodells mit größerem Aufwand verbunden, bietet der Inzidenzabstand eine genauere Bewertung. Dies ist beispielsweise der Fall, wenn die Abläufe fest im Quellcode verdrahtet sind. Der Inzidenzabstand liefert für Referenzmodell R3 einen Abstand von *vier*, für Referenzmodell R2 einen Abstand von *zwei*. Die automatische Suche würde somit Referenzmodell R2 für die Implementierung vorschlagen. Dies ist auch aus objektiver Sicht sinnvoll, da die zusätzliche Funktion F6 dort nicht genutzt werden muss. Stattdessen kann im operativen Ablauf der alternative Pfad genutzt werden. Die Abstandsmethode ermöglicht somit die Auswahl einer auf die Flexibilität der ERP-Software angepasste Metrik.

Ein weiterer Vorteil der Abstandsmethode ist die Möglichkeit, neben der Suche eines geeigneten Referenzmodells auch die Unterschiede zwischen Unternehmensmodell und Referenzmodell hervorheben zu können. So stehen beim anschließenden Kontinuierlichen Customizing mögliche Anpassungspunkte sofort zur Verfügung.

## 5 Fazit und Ausblick

### *Reflexion*

In der vorliegenden Arbeit wurde untersucht, ob die Suche nach einem geeigneten Referenzmodell aus einer Menge von Referenzmodellen automatisiert werden kann. Dies ist notwendig, um die Evaluierungsphase des Initialen Self Adaptive Customizing zu realisieren. Nach der Einleitung wurden die grundlegenden Begriffe erläutert und das Self Adaptive Customizing Konzept vorgestellt. Anschließend wurde untersucht, ob bereits Methoden für die Referenzmodellsuche existieren. Es konnten zwar einige Methoden für den Vergleich von Prozessmodellen gefunden werden, aber keine war für die Referenzmodellsuche geeignet. Daher wurden die vier Metriken Fallgleichheit, Funktionsabstand, Funktionsdifferenz und Inzidenzabstand entwickelt. Bei der Anwendung der Metrik Fallgleichheit wurde festgestellt, dass diese sehr aufwendig ist und unter bestimmten Voraussetzungen die zur Verfügung stehenden Modelle nicht differenziert genug betrachtet. Aus diesem Grund wurden weitere Metriken entwickelt, welche in der Abstandsmethode Anwendung finden. Vorteil von Funktionsdifferenz und Funktionsabstand ist der geringe Aufwand für die Berechnung der Metriken. Allerdings berücksichtigen diese nicht die sachlogischen Abhängigkeiten zwischen den Funktionen. Mit dem Inzidenzabstand hingegen wurde eine Metrik entwickelt, welche differenziert und unter Berücksichtigung der sachlogischen Abhängigkeiten geeignete Referenzmodelle identifizieren kann.

Aus den Darstellungen in Kapitel 4 kann geschlussfolgert werden, dass die im ersten Kapitel gestellte Frage bezüglich der Automatisierbarkeit der Suche von Referenzmodellen zu gegebenen Unternehmensmodellen positiv zu beantworten ist. Mit der Fallbasierten Referenzmodellsuche wie auch mit der Abstandsmethode kann automatisiert ein adäquates Referenzmodell gefunden werden. Vorausgesetzt wurde dabei, dass die Modelle als Workflow-Netze vorliegen. Die Erfüllung dieser Voraussetzung wird durch die Konvertierungsphase des Initialen Self Adaptive Customizing sichergestellt. Organisationsobjekte, das prozessprägende Objekt sowie Input- und Outputobjekte werden bei der Referenzmodellsuche implizit berücksichtigt, da diese für die Bestimmung der semantisch äquivalenten Funktionen in der Konvertierungsphase verwendet werden.

### *Weitere Anwendungsmöglichkeiten*

Anstatt für die vollständige Automatisierung des SAC können die Suchmethoden auch für die Implementierung eines Expertensystems (Wizard) genutzt werden. Nach Einlesen des Unternehmensmodells könnte das Expertensystem anschließend eine (nach

Ähnlichkeit zum Unternehmensmodell) geordnete Liste mit geeigneten Referenzmodellen bereitstellen. Ein Entscheider kann dann weitere Kriterien in die Wahl des zu verwendenden Referenzmodells einfließen lassen.

Weiterhin könnte die Referenzmodellsuche auch in der komponentenorientierten Softwareentwicklung Anwendung finden. So kann die fachliche Spezifikation von einzelnen Komponenten in Form von Referenzmodellen angegeben werden. Die Komponentenauswahl und -komposition kann dann durch Abgleich mit dem Unternehmensmodell automatisiert erfolgen.

### *Ausblick*

Um das Gesamtkonzept des Self Adaptive Customizing umsetzen zu können, müssen auch die anderen Phasen untersucht und ein Umsetzungskonzept für diese entwickelt werden. Für die Konvertierungsphase des Initialen SAC wurden in Abschnitt 3.1.2 ein paar Lösungsansätze genannt. Bei der Realisierung des Kontinuierlichen Customizing können Verfahren des Applikation Usage Mining zum Einsatz kommen. Zur Ermittlung von geeigneten Parametern für das Customizing der ERP-Software könnten dabei Genetische Algorithmen verwendet werden.<sup>54</sup>

Bezüglich der hier betrachteten Evaluierungsphase des Initialen SAC kann weiterhin untersucht werden, ob eine Gewichtung der Funktionen sinnvoll ist um deren Wichtigkeit differenzieren zu können. Wenn ja, können die Fallbasierte Referenzmodellsuche und die Abstandsmethode um diesen Faktor erweitert werden. Durch entsprechende Verteilung der Gewichte könnten dann K. O.-Kriterien definiert werden, sodass Referenzmodelle, welche diese Funktionen nicht enthalten, aus der Lösungsmenge entfernt werden.

Eine andere Möglichkeit die Fallbasierte Referenzmodellsuche zu verfeinern ist die Einführung von optionalen Funktionen. Hierbei werden Funktionen, welche im Referenzmodell ausgelassen werden können,<sup>55</sup> entsprechend markiert. Beim Vergleich mit den Fällen des Unternehmensmodells werden dann die Referenzprozessfälle mit und ohne optionale Funktionen berücksichtigt. Dies kann zu aussagekräftigeren Abständen führen. Insbesondere wenn davon ausgegangen wird, dass Referenzmodelle meistens mehr Funktionen anbieten als vom Unternehmen benötigt, diese aber auch ausgelassen werden können.

---

<sup>54</sup> Für eine Einführung in Genetische Algorithmen siehe Bodendorf (2006), S. 193 ff.

<sup>55</sup> Beispielsweise weil ein Schalter für die Nutzung oder Nichtnutzung bestimmter Funktionen bereits in der ERP-Software enthalten ist.

Weiterhin könnten die vorgestellten Verfahren erweitert werden, sodass neben den Funktionen und deren Reihenfolgebeziehungen auch Organisationseinheiten und Prozessobjekte explizit berücksichtigt werden. Ist dies möglich, kann auch die Aufbaustruktur innerhalb der ERP-Software während der Implementierungsphase automatisiert angepasst werden.

In einer fortführenden Bearbeitung des vorliegenden Themas könnte zudem die Anwendbarkeit der Verfahren in der Praxis überprüft werden. Mit entsprechenden empirischen Untersuchungen könnte festgestellt werden, ob das Referenzmodell mit dem geringsten Abstand zum Unternehmensmodell unter realen Bedingungen das optimale Modell für die informationstechnische Realisierung der Unternehmensprozesse ist. Dafür sollten mehrere Partner aus der Wirtschaft in die Untersuchungen einbezogen werden.

## Anhang

## A Erreichbarkeitsanalyse der Referenzmodelle

Tab. A.1: Erreichbarkeitsanalyse Referenzmodell R1

Markierung	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	Schaltungen	Schaltfolgen
$M_0$	1	0	0	0	0	0	0	0	$F1 \rightarrow M_1$	$w_{1,1} = F1$
$M_1$	0	1	0	0	1	0	0	0	$F2 \rightarrow M_2$ $F4 \rightarrow M_3$	$w_{1,1} = F1, F2$ $w_{1,2} = F1, F4$
$M_2$	0	0	1	0	1	0	0	0	$F4 \rightarrow M_4$ $F6 \rightarrow M_5$	$w_{1,1} = F1, F2, F4$ $w_{1,3} = F1, F2, F6$
$M_3$	0	1	0	0	0	1	0	0	$F2 \rightarrow M_4$ $F5 \rightarrow M_6$	$w_{1,2} = F1, F4, F2$ $w_{1,4} = F1, F4, F5$
$M_4$	0	0	1	0	0	1	0	0	$F5 \rightarrow M_7$  $F6 \rightarrow M_8$	$w_{1,1} = F1, F2, F4, F5$ $w_{1,2} = F1, F4, F2, F5$ $w_{1,5} = F1, F2, F4, F6$ $w_{1,6} = F1, F4, F2, F6$
$M_5$	0	0	0	0	1	0	0	1	$F4 \rightarrow M_8$	$w_{1,3} = F1, F2, F6, F4$
$M_6$	0	1	0	0	0	0	1	0	$F2 \rightarrow M_7$	$w_{1,4} = F1, F4, F5, F2$
$M_7$	0	0	1	0	0	0	1	0	$F6 \rightarrow M_9$	$w_{1,1} = F1, F2, F4, F5, F6$ $w_{1,2} = F1, F4, F2, F5, F6$ $w_{1,4} = F1, F4, F5, F2, F6$
$M_8$	0	0	0	0	0	1	0	1	$F5 \rightarrow M_9$	$w_{1,3} = F1, F2, F6, F4, F5$ $w_{1,5} = F1, F2, F4, F6, F5$ $w_{1,6} = F1, F4, F2, F6, F5$
$M_9$	0	0	0	0	0	0	1	1	$F3 \rightarrow M_{10}$	$w_{1,1}^v = F1, F2, F4, F5, F6, F3$ $w_{1,2}^v = F1, F4, F2, F5, F6, F3$ $w_{1,3}^v = F1, F2, F6, F4, F5, F3$ $w_{1,4}^v = F1, F4, F5, F2, F6, F3$ $w_{1,5}^v = F1, F2, F4, F6, F5, F3$ $w_{1,6}^v = F1, F4, F2, F6, F5, F3$
$M_{10}$	0	0	0	1	0	0	0	0		

Tab. A.2: Erreichbarkeitsanalyse Referenzmodell R2

Markierung	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	Schaltungen	Schaltfolgen
$M_0$	1	0	0	0	0	0	0	$F1 \rightarrow M_1$	$w_{2,1} = F1$
$M_1$		1			1			$F2 \rightarrow M_2$ $F4 \rightarrow M_3$	$w_{2,1} = F1, F2$ $w_{2,2} = F1, F4$
$M_2$			1		1			$F4 \rightarrow M_4$	$w_{2,1} = F1, F2, F4$
$M_3$		1				1		$F2 \rightarrow M_4$	$w_{2,2} = F1, F4, F2$
$M_4$			1		1			$F5 \rightarrow M_5$ $F6 \rightarrow M_5$	$w_{2,1} = F1, F2, F4, F5$ $w_{2,2} = F1, F4, F2, F5$ $w_{2,3} = F1, F2, F4, F6$ $w_{2,4} = F1, F4, F2, F6$
$M_5$			1				1	$F3 \rightarrow M_6$	$w_{2,1}^v = F1, F2, F4, F5, F3$ $w_{2,2}^v = F1, F4, F2, F5, F3$ $w_{2,3}^v = F1, F2, F4, F6, F3$ $w_{2,4}^v = F1, F4, F2, F6, F3$
$M_6$				1					

Tab. A.3: Erreichbarkeitsanalyse Referenzmodell R3

Markierung	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	Schaltungen	Schaltfolgen
$M_0$	1	0	0	0	0	0	0	0	0	$F1 \rightarrow M_1$	$w_{3,1} = F1$
$M_1$	0	1	0	0	1	0	0	0	0	$F2 \rightarrow M_2$ $F4 \rightarrow M_3$	$w_{3,1} = F1, F2$ $w_{3,2} = F1, F4$
$M_2$	0	0	1	0	1	0	0	0	0	$F4 \rightarrow M_4$	$w_{3,1} = F1, F2, F4$
$M_3$	0	1	0	0	0	1	0	1	0	$F2 \rightarrow M_4$ $F5 \rightarrow M_5$ $F6 \rightarrow M_6$	$w_{3,2} = F1, F4, F2$ $w_{3,3} = F1, F4, F5$ $w_{3,4} = F1, F4, F6$
$M_4$	0	0	1	0	0	1	0	1	0	$F5 \rightarrow M_7$  $F6 \rightarrow M_8$	$w_{3,1} = F1, F2, F4, F5$ $w_{3,2} = F1, F4, F2, F5$ $w_{3,5} = F1, F2, F4, F6$ $w_{3,6} = F1, F4, F2, F6$
$M_5$	0	1	0	0	0	0	1	1	0	$F2 \rightarrow M_7$ $F6 \rightarrow M_9$	$w_{3,3} = F1, F4, F5, F2$ $w_{3,7} = F1, F4, F5, F6$
$M_6$	0	1	0	0	0	1	0	0	1	$F2 \rightarrow M_8$ $F5 \rightarrow M_9$	$w_{3,4} = F1, F4, F6, F2$ $w_{3,8} = F1, F4, F6, F5$
$M_7$	0	0	1	0	0	0	1	1	0	$F6 \rightarrow M_{10}$	$w_{3,1} = F1, F2, F4, F5, F6$ $w_{3,2} = F1, F4, F2, F5, F6$ $w_{3,3} = F1, F4, F5, F2, F6$
$M_8$	0	0	1	0	0	1	0	0	1	$F5 \rightarrow M_{10}$	$w_{3,4} = F1, F4, F6, F2, F5$ $w_{3,5} = F1, F2, F4, F6, F5$ $w_{3,6} = F1, F4, F2, F6, F5$
$M_9$	0	1	0	0	0	0	1	0	1	$F2 \rightarrow M_{10}$	$w_{3,7} = F1, F4, F5, F6, F2$ $w_{3,8} = F1, F4, F6, F5, F2$
$M_{10}$	0	0	1	0	0	0	1	0	1	$F3 \rightarrow M_{11}$	$w_{3,1}^v = F1, F2, F4, F5, F6, F3$ $w_{3,2}^v = F1, F4, F2, F5, F6, F3$ $w_{3,3}^v = F1, F4, F5, F2, F6, F3$ $w_{3,4}^v = F1, F4, F6, F2, F5, F3$ $w_{3,5}^v = F1, F2, F4, F6, F5, F3$ $w_{3,6}^v = F1, F4, F2, F6, F5, F3$ $w_{3,7}^v = F1, F4, F5, F6, F2, F3$ $w_{3,8}^v = F1, F4, F6, F5, F2, F3$
$M_{11}$	0	0	0	1	0	0	0	0	0		

Tab. A.4: Erreichbarkeitsanalyse Referenzmodell R4

Markierung	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	Schaltungen	Schaltfolgen
$M_0$	1	0	0	0	0	0	0	$F1 \rightarrow M_1$	$w_{4,1} = F1$
$M_1$	0	1	0	0	1	0	0	$F4 \rightarrow M_2$ $F6 \rightarrow M_3$	$w_{4,1} = F1, F4$ $w_{4,2} = F1, F6$
$M_2$	0	1	0	0	0	1	0	$F5 \rightarrow M_4$ $F6 \rightarrow M_5$	$w_{4,1} = F1, F4, F5$ $w_{4,3} = F1, F4, F6$
$M_3$	0	0	1	0	1	0	0	$F4 \rightarrow M_5$	$w_{4,2} = F1, F6, F4$
$M_4$	0	1	0	0	0	0	1	$F6 \rightarrow M_6$	$w_{4,1} = F1, F4, F5, F6$
$M_5$	0	0	1	0	0	1	0	$F5 \rightarrow M_6$	$w_{4,2} = F1, F6, F4, F5$ $w_{4,3} = F1, F4, F6, F5$
$M_6$	0	0	1	0	0	0	1	$F3 \rightarrow M_7$	$w_{4,1}^v = F1, F4, F5, F6, F3$ $w_{4,2}^v = F1, F6, F4, F5, F3$ $w_{4,3}^v = F1, F4, F6, F5, F3$
$M_7$	0	0	0	1	0	0	0		

## B Inzidenzmatrizen

$$C^{R2} = \begin{matrix} & \begin{matrix} F1 & F2 & F3 & F4 & F5 & F6 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{matrix} & \left\{ \begin{array}{cccccc} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right\} \end{matrix}$$

$$C^{R3} = \begin{matrix} & \begin{matrix} F1 & F2 & F3 & F4 & F5 & F6 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \end{matrix} & \left\{ \begin{array}{cccccc} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{array} \right\} \end{matrix}$$

$$C^{R4} = \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{array} \begin{array}{cccccc} \text{F1} & \delta_{F2} & \text{F3} & \text{F4} & \text{F5} & \text{F6} \\ \left\{ \begin{array}{cccccc} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \end{array} \right\} \end{array}$$

## Literaturverzeichnis

- Alpar, P.; Grob, H. L.; Weimann, P.; Winter, R. (2008): Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen. 5. Aufl., Wiesbaden.
- Baumgarten, B. (1996): Petri-Netze: Grundlagen und Anwendungen. 2. Aufl., Heidelberg – Berlin – Oxford.
- Becker, J.; Kahn, D. (2005): Der Prozess im Fokus. In: Becker, J.; Kugeler, M.; Rosemann, M. (2005), S. 3-16.
- Becker, J.; Kugeler, M.; Rosemann, M. (Hrsg.) (2005): Prozessmanagement: Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 5. Aufl., Berlin u. a.
- Becker, J.; Meise, V. (2005): Strategie und Ordnungsrahmen. In: Becker, J.; Kugeler, M.; Rosemann, M. (2005), S. 105-154.
- Becker, J.; Schütte, R. (2004): Handelsinformationssysteme: Domänenorientierte Einführung in die Wirtschaftsinformatik. 2. Aufl., Frankfurt am Main.
- Bellahsène, Z.; Léonard, M. (Hrsg.) (2008): Advanced Information Systems Engineering: 20th International Conference, CAiSE 2008. Berlin – Heidelberg – New York.
- Bodendorf, F. (2006): Daten- und Wissensmanagement. 2. Aufl., Berlin – Heidelberg – New York.
- Booch, G.; Rumbaugh, J.; Jacobson, I. (2006): Das UML Benutzerhandbuch: Aktuell zur Version 2.0. München u. a.
- Brand, H. (1999): SAP® R/3® Implementation with ASAP: The Official SAP Guide. San Francisco et. al.
- Bürner, P. (2007): Status ISO Zertifizierung Infor Implementierung.  
<http://www.infor.de/content/downloads/1635529/>. 10. Mai 2009.
- Chen, R.; Scheer, A.-W. (1994): Modellierung von Prozessketten mittels Petri-Netz-Theorie. Arbeitsbericht des Instituts für Wirtschaftsinformatik Nr. 107, Saarbrücken.
- Desel, J.; Wekse, M. (Hrsg.) (2002): Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen – Promise 2002. Bonn.
- Dijkman, R. (2008): Diagnosing Differences between Business Process Models. In: Dumas, M.; Reichert, M.; Shan, M.-C. (2008), S. 261-277.
- Dumas, M.; Reichert, M.; Shan, M.-C. (Hrsg.) (2008): Business Process Management: 6th International Conference, BPM 2008. Berlin – Heidelberg – New York.
- Dumke, R. (2003): Software Engineering: Eine Einführung für Informatiker und Ingenieure: Systeme, Erfahrungen, Methoden, Tools. 4. Aufl., Wiesbaden.
- Energiewirtschaftsgesetz (EnWG) vom 7. Juli 2005. BGBl. I.
- Gadatsch, A. (2008): Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 5. Aufl., Wiesbaden.
- Geihs, Kurt (2008): Selbst-adaptive Software. Informatik Spektrum, 31. Jg, Heft 2, S. 133-145.

- Gronau, N. (2009): Enterprise Resource Planning-System. In: Kurbel, K.; Becker, J.; Gronau, N.; Sinz, E.; Suhl, L. (2009).
- Günther, H.; Tempelmeier, H. (2005): Produktion und Logistik. 6. Aufl., Berlin – Heidelberg – New York.
- Hansmann, H.; Neumann, S. (2005): Prozessorientierte Einführung von ERP-Systemen. In: Becker, J.; Kugeler, M.; Rosemann, M. (2005), S. 329-372.
- Heinrich, L.; Heinzl, A.; Roithmayr, F. (2004): Wirtschaftsinformatik-Lexikon. 7. Aufl., München.
- Heuer, A.; Saake, G. (2000): Datenbanken: Konzepte und Sprachen. 2. Aufl., Bonn.
- Hollingsworth, D. (1995): The Workflow Reference Model. <http://wfmc.org/reference-model.html>. 26. März 2009.
- Jensen, K. (1991): Coloured Petri nets: A high level language for system design and analysis. In: Rozenberg, G. (1991), S. 342-416.
- Jensen, K.; van der Aalst, W. M. P. (Hrsg.) (2009): Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems. Berlin – Heidelberg – New York.
- Kassem, G. (2006): Application Usage Mining: Grundlagen und Verfahren. Dissertation, Otto-von-Guericke Universität Magdeburg.
- Kassem, G. (2007): Self Adaptive Customizing: Ein Konzept zum automatischen Customizing eines ERP-Systems. In: Rautenstrauch, C.: (2007), S. 63-74.
- Keller, G.; Nüttgens, M.; Scheer, A.-W. (1992): Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. Arbeitsbericht des Instituts für Wirtschaftsinformatik Nr. 89, Saarbrücken.
- Kohnke, O.; Bungard, W. (Hrsg.) (2005): SAP-Einführung mit Change Management: Konzepte, Erfahrungen und Gestaltungsempfehlungen. Wiesbaden.
- Kurbel, K.; Becker, J.; Gronau, N.; Sinz, E.; Suhl, L. (Hrsg.) (2009): Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon. 2. Aufl., München. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de>. 29. April 2009.
- Lawson (2009): Lawson StepWise. [http://lawson.xldoc.net/get\\_lores.php?file=LAWDTP\\_SP020\\_GEEN\\_A4](http://lawson.xldoc.net/get_lores.php?file=LAWDTP_SP020_GEEN_A4). 10. Mai 2009.
- Lohmann, N.; Verbeek, E.; Dijkman, R. (2009): Petri Net Transformations for Business Processes: A Survey. In: Jensen, K.; van der Aalst, W. M. P. (2009), S. 46-63.
- Mertens, P. (2004): Integrierte Informationsverarbeitung, Band 1: Operative Systeme in der Industrie. 14. Aufl., Wiesbaden.
- Mertens, P.; Bodendorf, F.; König, W.; Picot, A.; Schumann, M.; Hess, T. (2005): Grundzüge der Wirtschaftsinformatik. 9. Aufl., Berlin – Heidelberg – New York.
- Müller, G. (2008): WI - Für Sie gelesen: The Deep Structure of Business Processes. Wirtschaftsinformatik, 50. Jg., Heft 6, S. 508-509.
- Nüttgens, M.; Rump, F. J. (2002): Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Desel, J.; Weske, M. (2002). S. 64-77.

- Niedereichholz, J; Reske, J. (2005): Probleme bei der Einführung von Standardsoftware. In: Kohnke, O.; Bungard, W. (2005), S. 75-85.
- o. V. (2007): Rahmenempfehlung für die Universitätsausbildung in Wirtschaftsinformatik: Mitteilungen der Wissenschaftlichen Kommission Wirtschaftsinformatik und des GI-Fachbereichs Wirtschaftsinformatik. Wirtschaftsinformatik, 49. Jg., Heft 4, S. 318-325.
- Oberweis, A. (1996): Modellierung und Ausführung von Workflows mit Petri-Netzen. Stuttgart – Leipzig.
- Olbrich, R. (2006): Marketing: Eine Einführung in die marktorientierte Unternehmensführung. 2. Aufl., Berlin – Heidelberg – New York.
- OMG (2009): Business Process Modelling Notation.  
<http://www.omg.org/spec/BPMN/1.2/>. 21. Juni 2009.
- Oracle (2009): Oracle Unified Method.  
<http://www.oracle.com/consulting/library/briefs/oracle-unified-method.pdf>. 10. Mai 2009.
- Österreich, B. (2006): Analyse und Design mit UML 2.1: Objektorientierte Softwareentwicklung. 8. Aufl., München.
- Patig, S. (2003): SAP® R/3® am Beispiel erklärt: Eine Einführung in die Anwendungs-komponenten MM, PP, SD und ihre Integration mit Hinweisen zur Durchführung von Lehrveranstaltungen. Frankfurt am Main u. a.
- Patig, S. (2006): Die Evolution von Modellierungssprachen. Habilitation, Otto-von-Guericke-Universität Magdeburg.
- Petri, C. A. (1962): Kommunikation mit Automaten. Dissertation, Technische Hochschule Darmstadt.
- Priemer, J. (1995): Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle. Dissertation, Universität Münster.
- Rautenstrauch, C. (Hrsg.) (2007): Die Zukunft der Anwendungssoftware – die Anwendungssoftware der Zukunft. Aachen.
- Rautenstrauch, C.; Schulze, T. (2003): Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker. Berlin u. a.
- Reisig, W. (1986): Petrinetze: Eine Einführung. 2. Aufl., Berlin u. a.
- Rosemann, M. (1996): Komplexitätsmanagement in Prozessmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Dissertation, Universität Münster.
- Rosemann, M.; Schwegmann, A.; Delfmann, P. (2005): Vorbereitung der Prozessmodellierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (2005), S. 45-103.
- Rozenberg, G. (Hrsg.) (1991): Advances in Petri nets 1990. Berlin – Heidelberg – New York.
- SAP (2009a): Customizing.  
[http://help.sap.com/erp2005\\_ehp\\_03/helpdata/DE/55/103da2be01c241a106c3cbedbab389/frameset.htm](http://help.sap.com/erp2005_ehp_03/helpdata/DE/55/103da2be01c241a106c3cbedbab389/frameset.htm). 03. April 2009.
- SAP (2009b): ARIS Business Architect for SAP: Geschäftsprozessbasierte SAP-Prozesse umsetzen und optimieren. <http://www.ids->

- scheer.com/set/6473/ARIS%20Business%20Architect%20for%20SAP%20FS%20de.pdf. 21. Juni 2009.
- Scheer, A.-W. (2002): ARIS – Vom Geschäftsprozess zum Anwendungssystem. 4. Aufl., Berlin – Heidelberg – New York.
- Scheer, A.-W.; Adam, O.; Erbach, F. (2005): Next Generation Business Process Management. In: Scheer, A.-W.; Jost, W.; Wagner, K. (2005a), S. 1-15.
- Scheer, A.-W.; Jost, W.; Wagner, K. (Hrsg.) (2005a): Von Prozessmodellen zu lauffähigen Anwendungen. Berlin – Heidelberg – New York.
- Scheuch, R.; Scheidt, S.; Stähler, D. (2004):“Adaptive Enterprise Computing“ – Prozessorientierte Softwareentwicklung als Voraussetzung flexibler IT Strukturen. Information Management und Consulting, 19. Jg., Sonderausgabe Oktober, S. 61-65.
- Seemann, J.; von Gudenberg, J. W. (2006): Software-Entwurf mit UML 2. Objektorientierte Modellierung mit Beispielen in Java. 2. Aufl., Berlin – Heidelberg – New York.
- Speck, M., Schnetgöke, N. (2005): Sollmodellierung und Prozessoptimierung. In: Becker, J.; Kugeler, M.; Rosemann, M. (2005), S. 185-220.
- Stahlknecht, P.; Hasenkamp, U. (2005): Einführung in die Wirtschaftsinformatik. 11. Aufl., Berlin – Heidelberg – New York.
- Theurer, M. (2007): Herr Schneider zieht um. In: Frankfurter Allgemeine Zeitung vom 22. August 2007, S. 13.
- Tolkdorf, R.; Mochol, M.; Heese, R.; Eckstein, R.; Oldakowski, R.; Bizer, C. (2006): Semantic-Web-Technologien im Arbeitsvermittlungsprozess. Wirtschaftsinformatik, 48. Jg., Heft 1, S. 17-26.
- van der Aalst, W. M. P. (1998): The Application of Petri Nets to Workflow Management. In: The Journal of Circuits, Systems and Computers. Nr. 8-1, S. 21-66.
- van der Aalst, W. M. P.; van Hee, K. M. (2004): Workflow Management: Models, Methods and Systems. Cambridge – London.
- van Dongen, B.; Dijkman, R.; Mendling, J. (2008): Measuring Similarity between Business Process Models. In: Bellahsène, Z.; Léonard, M. (2008), S. 450-464.
- Volmering, T; Scholz, T. (2004): Business Process Management mit SAP NetWeaver und ARIS. Information Management und Consulting, 19. Jg., Sonderausgabe Oktober, S. 36-42.
- Wenzel, P. (2001): Betriebswirtschaftliche Anwendungen mit SAP R/3®: Eine Einführung inklusive Customizing, ABAP/4, Accelerated SAP (ASAP), Projekt-system (PS). Braunschweig – Wiesbaden.
- Weske, M. (2007): Business Process Management: Concepts, Languages, Architectures. Berlin – Heidelberg – New York.
- Woods, D. (2003): Enterprise Services Architecture. Beijing u. a.

## **Abschließende Erklärung**

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit selbstständig, 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.

Loburg, den 28. Juli 2009