



UNIVERSITÄT
KOBLENZ · LANDAU



Institut für
Wirtschaftsinformatik

Fachbereich Informatik
Universität Koblenz-Landau

ULRICH FRANK
BODO L. VAN LAAK

ANFORDERUNGEN AN SPRACHEN ZUR MODELLIERUNG VON GESCHÄFTSPROZESSEN

Januar 2003



UNIVERSITÄT
KOBLENZ · LANDAU



**Institut für
Wirtschaftsinformatik**

Fachbereich Informatik
Universität Koblenz-Landau

ULRICH FRANK
BODO L. VAN LAAK

ANFORDERUNGEN AN SPRACHEN ZUR MODELLIERUNG VON GESCHÄFTSPROZESSEN

Januar 2003

Arbeitsberichte des Instituts für Wirtschaftsinformatik

Nr. 34

Die Arbeitsberichte des Instituts für Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i.d.R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

The "Arbeitsberichte des Instituts für Wirtschaftsinformatik" comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

Alle Rechte vorbehalten. Insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen - auch bei nur auszugsweiser Verwertung.

All rights reserved. No part of this report may be reproduced by any means, or translated.

Anschrift der Verfasser

Address of the authors:

Prof. Dr. Ulrich Frank,
Dipl. Inform. Bodo L. van Laak,
Institut für Wirtschaftsinformatik
Universität Koblenz-Landau

**Arbeitsberichte des Instituts für
Wirtschafts- und Verwaltungsinformatik**

Herausgegeben von / Edited by:

Prof. Dr. Ulrich Frank
Prof. Dr. J. Felix Hampe

Bezugsquelle / Source of Supply:

Institut für Wirtschaftsinformatik
Universität Koblenz-Landau
Rheinau 1
56075 Koblenz
Tel.: 0261-287-2520
Fax: 0261-287-2521



**Institut für
Wirtschaftsinformatik**

Fachbereich Informatik
Universität Koblenz-Landau

Inhaltsverzeichnis

<i>Inhaltsverzeichnis</i>	5
<i>Abbildungsverzeichnis</i>	7
<i>Tabellenverzeichnis</i>	9
<i>Zusammenfassung</i>	11
1. Einleitung	13
2. Abgrenzung des Untersuchungsgegenstands	17
2.1. Begriffliche Grundlagen	17
2.2. Zur wirtschaftlichen Bedeutung der Geschäftsprozessmodellierung	21
3. Generelle Anforderungen an Modellierungssprachen	25
3.1. Formale Anforderungen	26
3.1.1. Korrektheit und Vollständigkeit.....	26
3.1.2. Einheitlichkeit und Redundanzfreiheit.....	26
3.1.3. Wiederverwendbarkeit und Wartbarkeit: Abstraktion	27
3.2. Anwenderbezogene Anforderungen	28
3.2.1. Einfachheit	28
3.2.2. Verständlichkeit und Anschaulichkeit	29
3.3. Anwendungsbezogene Anforderungen	31
3.3.1. Mächtigkeit und Angemessenheit	31
3.3.2. Operationalisierbarkeit.....	33
4. Geschäftsprozessmodellierung	35
4.1. Grundlegende Anforderungen an einschlägige Modellierungssprachen	35
4.1.1. Abstraktionsebenen.....	35
4.1.2. Flexibilität/Anpassbarkeit	40
4.1.3. Unterstützung von Sichten	43
4.2. Betriebswirtschaftliche und organisatorische Konzepte	45
4.2.1. Prozess	45
4.2.2. Ereignis	48
4.2.3. Ziele von Geschäftsprozessen	51
4.2.4. Ressourcen	61
4.2.5. Prozessgegenstand	68
4.2.6. Modellierung der Organisationsstruktur	70
4.2.7. Konzepte zur Modellierung von Rollen	74

4.2.8. Konzepte zur Modellierung unternehmensübergreifender Geschäftsprozesse.....	75
4.2.9. Berücksichtigung weiterer betriebswirtschaftlicher Aspekte	77
4.3. Kontrollstrukturen.....	77
4.3.1. Sequenz.....	77
4.3.2. Bedingungen und Regeln	78
4.3.3. Alternative Ausführung.....	78
4.3.4. Parallelität	79
4.3.5. Nebenläufigkeit.....	79
4.3.6. Unspezifizierte Reihenfolge.....	80
4.3.7. Iteration.....	80
4.3.8. Synchronisation.....	81
4.3.9. Transaktion	82
4.4. Ausnahmen	83
4.5. Integritätsbedingungen.....	83
4.5.1. Kardinalitäten.....	84
4.5.2. Preconditions.....	84
4.5.3. Postconditions	85
4.5.4. Prozesstypinvarianten	86
4.6. Unterstützung der Entwicklung von Informationssystemen.....	88
4.6.1. Integration mit software-technischen Abstraktionen.....	88
4.6.2. Unterstützung der Entwicklung von Workflow-Management Systemen.....	89
4.7. Unterstützung individueller Anpassungen.....	92
4.8. Dokumentation	92
4.8.1. Motivation und Einsatzzweck: Differenzierung von Benutzergruppen	93
4.8.2. Sprachbeschreibung	93
4.9. Spezifikation	94
5. Einbettung in eine Methode.....	97
5.1. Projektspezifische Rollen und Ressourcen	97
5.2. Vorgehensmodell.....	98
6. Abschließende Bemerkungen	99
Literaturverzeichnis	101
Bisherige Arbeitsberichte.....	107

Abbildungsverzeichnis

<i>Abbildung 1: Verhältnis von Geschäftsprozessen und Workflows</i>	<i>20</i>
<i>Abbildung 2: Beispiel für Anschaulichkeit</i>	<i>30</i>
<i>Abbildung 3: Beispiel für eingeschränkte Anschaulichkeit</i>	<i>30</i>
<i>Abbildung 4: Geschäftsprozessmodellierung Abstraktionsebenen</i>	<i>36</i>
<i>Abbildung 5: Abstraktionsebenen bei der Modellierung von Ressourcen</i>	<i>38</i>
<i>Abbildung 6: „Typdifferenzierung“ durch Eigenschaften von Instanzen</i>	<i>41</i>
<i>Abbildung 7: Mögliche Typen von Zielen</i>	<i>54</i>
<i>Abbildung 8: Organigramme stellen verschiedene Sichten dar</i>	<i>73</i>
<i>Abbildung 9: Ableitung eines Workflow-Schemas aus einem Geschäftsprozessmodell unter Verwendung einer standardisierten Spezifikationssprache</i>	<i>91</i>

Tabellenverzeichnis

<i>Tabelle 1: Bewertung von Ansätzen zur Erhöhung der Modellierungsflexibilität.....</i>	<i>43</i>
<i>Tabelle 2: Arten von Ereignissen</i>	<i>49</i>
<i>Tabelle 3: Projektion von Produktionsfaktoren auf Geschäftsprozesse</i>	<i>62</i>
<i>Tabelle 4: Vorbedingungen</i>	<i>85</i>
<i>Tabelle 5: Nachbedingungen.....</i>	<i>86</i>
<i>Tabelle 6: Invarianten.....</i>	<i>87</i>

Zusammenfassung

Die Geschäftsprozessmodellierung spielt in Unternehmen eine zentrale Rolle: sie dient der Dokumentation des Unternehmens und seiner Geschäftsprozesse, der Analyse und Optimierung der Organisation, sowie als Grundlage zur Entwicklung prozessunterstützender Software (was die Softwareentwicklung als auch die Vorbereitung zum Einsatz von Workflow-Managementsystemen einschließt).

Es existiert bereits eine Fülle von Sprachen zur Geschäftsprozessmodellierung. Oftmals offenbart die Anwendung der Sprachen jedoch Mängel, die die Effizienz der Modellierung negativ beeinflussen, weil bestimmte Sachverhalte nicht oder nur schwierig auszudrücken sind. Je Modellierungszweck sind dabei unterschiedliche Anforderungen an Modellierungssprachen zu stellen.

In diesem Beitrag entwickeln wir Anforderungen an Modellierungssprachen im allgemeinen sowie Sprachen und Methoden zur Geschäftsprozessmodellierung im besonderen. Darunter finden sich neben den offenkundig mit Prozessmodellen zu assoziierenden Konzepten wie „Prozess“ und „Ereignis“ sowie Kontrollstrukturen vor allem betriebswirtschaftliche Konzepte (z.B. Ressourcen, Organisationsstrukturen, Organisationsziele). Wir stellen uns in diesem Arbeitsbericht neben praktischen vor allem akademischen Fragestellungen wie beispielsweise der nach Abstraktionsebenen der Konzepte „Prozess“ oder „Ressource“.

Der vorliegende Beitrag entstand im Rahmen einer Evaluation von Werkzeugen zur Unternehmensmodellierung im Allgemeinen bzw. zur Geschäftsprozessmodellierung im speziellen. Neben der Entwicklung eines umfassenden theoretischen Bezugsrahmens für die Evaluation von Unternehmensmodellierungswerkzeugen ist dieser Arbeitsbericht Grundlage zur Einführung neuer Konzepte in die am Institut für Wirtschaftsinformatik der Universität Koblenz entwickelte Sprache MEMO-OrgML (Organisation Modeling Language).

1. Einleitung

Seit ungefähr zehn Jahren wird in vielen Veröffentlichungen (u.a. [JoMc+94], [Kube95], [Ould95], [Bern97], [Möbi99]) die Bedeutung der Organisation von Geschäftsprozessen für die Verbesserung der Effizienz und Wettbewerbsfähigkeit von Unternehmen betont. Dabei wird – verbunden mit Schlagworten wie „Business Process Re-Engineering“ ([HaCh93]), „Process Innovation“, „Business Process Improvement“ ([Harr91]) oder „Process-Centred Organization“ ([Hamm96]) – einerseits dazu geraten, bewusst von traditionellen Formen der Arbeitsteilung und Koordination zu abstrahieren, um die Voraussetzungen für umfassende, nachhaltig effizienzfördernde Reorganisationsmaßnahmen zu schaffen. Auf der anderen Seite wird dabei i.d.R. die herausragende Rolle der Informations- und Kommunikationstechnologie zur Realisation effizienter Geschäftsprozesse unterstrichen (besonders nachdrücklich in [Glas94], [Jaco94], [Hoch96], [Möbi99]). Traditionelle Anwendungssysteme bilden typischerweise betriebliche Funktionen (Finanzbuchhaltung, Fakturierung, Lagerverwaltung etc.) isoliert ab und bieten deshalb keine überzeugende Unterstützung von Geschäftsprozessen, in denen jeweils u.U. mehrere betriebliche Funktionen zu berücksichtigen sind. Im Unterschied dazu sind *Workflow Management Systeme* darauf gerichtet, die Durchführung und gegebenenfalls das Controlling von Geschäftsprozessen durchgängig zu unterstützen. Mittlerweile ist am Markt eine Reihe einschlägiger Systeme verfügbar. Mit der zunehmenden Bedeutung des Internet zur Anbahnung und Durchführung geschäftlicher Transaktionen erhält die Prozessorientierung zusätzliche Bedeutung. Während der zurückliegenden „E-Commerce“-Euphorie war es häufig hinreichend, Produkte über attraktiv gestaltete WWW-Seiten anzubieten, um mitunter grotesk wirkende Prognosen über die künftige Unternehmensentwicklung zu befördern. Die Zeiten haben sich geändert. Um die Potentiale des Internet für nachhaltige Wettbewerbsvorteile zu nutzen, ist es häufig unerlässlich, eine durchgängige Prozessunterstützung entlang unternehmensübergreifender Wertketten zu realisieren.

Die Reorganisation von Geschäftsprozessen wie auch die Einführung von Workflow Management Systemen erfordern die sorgfältige Analyse betriebswirtschaftlicher und software-technischer Zusammenhänge. Dazu empfiehlt sich eine strukturierte Beschreibung von Geschäftsprozessen unter Berücksichtigung aller für eine Reorganisation bedeutsamen Kriterien. Dabei ist zu berücksichtigen, dass Personen mit unterschiedlicher Qualifikation und mit voneinander abweichenden Zielen an entsprechenden Projekten beteiligt sind - neben Vertretern der Fachabteilungen ist hier an Organisationsexperten, DV-Experten und Software-Entwickler zu denken. Um den hinlänglich bekannten Verständigungsproblemen in solch heterogenen Gruppen entgegenzuwirken, sind anschauliche Darstellungen von Geschäftsprozessen hilfreich. Unter günstigen Voraussetzungen liefern sie den Beteiligten ein Medium für zielgerichtete Diskurse. Schon seit langer Zeit werden zur Darstellung nicht nur der Aufbauorganisation, sondern auch der Ablauforganisation grafisch visualisierte *Modelle* verwendet (ein Überblick findet sich in [Fran00]).

Zur Unterstützung der Erstellung, Auswertung und Pflege von Geschäftsprozessmodellen wird mittlerweile eine beachtliche Zahl von Werkzeugen angeboten. Vor diesem Hintergrund sehen sich viele Unternehmen in einer ambivalenten Situation. So spricht der Wettbewerbsdruck dafür, Geschäftsprozesse zu reorganisieren und in zunehmendem Maße zu automatisieren. Zudem empfiehlt die Komplexität des Vorhabens den Einsatz geeigneter Modellierungswerkzeuge. Gleichzeitig gibt es allerdings eine Reihe von Risiken, die die zuständigen Entscheidungsträger zögern lassen. Das gilt einerseits für die Unwägbarkeiten umfassender Reorganisationsmaßnahmen, die orthogonal zum Tagesgeschäft der Fachabteilungen laufen und deshalb nicht immer die nötige Aufmerksamkeit erfahren. Andererseits ist die Auswahl der Technologie - Modellierungswerkzeuge und Workflow Management Systeme – mit großen Schwierigkeiten verbunden. So handelt es sich dabei i.d.R. um komplexe Systeme, deren Evaluation einen Aufwand erfordert, der die Möglichkeiten vieler Anwenderunternehmen übersteigt. Dies gilt umso mehr, als es mit der Begutachtung eines Systems zumeist nicht getan ist. Vielmehr erfordert eine fundierte Entscheidung die vergleichende Betrachtung einer Reihe von Systemen.

Im vorliegenden Arbeitsbericht werden Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen entwickelt. Dabei verfolgen wir eine doppelte Zielsetzung. Auf der einen Seite soll ein Beitrag zur wissenschaftlich fundierten Modellierung und Analyse von Geschäftsprozessen geleistet werden, auf der anderen Seite soll der resultierende Bezugsrahmen Entscheidungsträgern in der Praxis eine Grundlage für die Bewertung und Auswahl einschlägiger Modellierungssprachen bzw. -methoden liefern. In einem weiteren, darauf aufbauenden Arbeitsbericht [FrLa03] wird ein Bezugsrahmen für die Bewertung von Werkzeugen zur Modellierung von Geschäftsprozessen und Workflows entwickelt.

2. Abgrenzung des Untersuchungsgegenstands

Der in diesem Arbeitsbericht entwickelte Bezugsrahmen ist auf Sprachen gerichtet, die die Erstellung, Verwaltung und Nutzung von Geschäftsprozessmodellen unterstützen. Das empfiehlt zunächst eine Betrachtung der zentralen in diesem Kontext zu berücksichtigenden Begriffe wie Geschäftsprozess, Workflow und Workflow Management System. Um den Aufwand zu rechtfertigen, den eine sorgfältige Evaluation einschlägiger Modellierungssprachen mit sich bringt, werden wir anschließend näher auf die zentrale Rolle der Geschäftsprozessmodellierung für eine nachhaltige Verbesserung der Wettbewerbsfähigkeit von Unternehmen eingehen.

2.1. Begriffliche Grundlagen

In der Literatur findet sich eine Vielzahl von Definitionen des Begriffs ‚Geschäftsprozess‘ (u.a. [HaCh93], [LiSu89], [Gait83]). Letztlich hängt die angemessene Konzeptualisierung des Begriffs vom jeweiligen Betrachtungswinkel ab. Im Vordergrund unserer Untersuchung steht eine betriebswirtschaftlich/organisatorische Perspektive. Das hat den Grund, dass Geschäftsprozesse vor allem im Hinblick auf betriebswirtschaftliche Ziele zu analysieren und zu gestalten sind. In organisatorischer Hinsicht ist ein Geschäftsprozess Bestandteil der *Ablauforganisation* eines Unternehmens. Im Hinblick auf die organisatorische Gestaltung eines Geschäftsprozesses ergeben sich daraus folgende Konsequenzen:

- Organisieren impliziert *Arbeitsteilung und Koordination*. Ein Geschäftsprozess beinhaltet danach Arbeitspakete bzw. Aufgaben und die Regelungen zu deren koordinierter Ausführung. Regelungen betreffen logische, zeitliche, räumliche und quantitative Bedingungen, die mit der Durchführung der Aufgaben eines Prozesses verbunden sind (vgl. [LiSu89], S. 26 ff.). Je nach Programmierungs- und Formalisierungsgrad (im Sinne der organisatorischen Begriffe Programmierung und Formalisierung) können die Regelungen den beteiligten Akteuren mehr oder weniger große Interpretations- bzw. Handlungsspielräume lassen.

- Der Begriff ‚Ablauforganisation‘ bezeichnet eine Abstraktion der Organisation eines Unternehmens. Tatsächlich kann eine Ablauforganisation nicht unabhängig von einer korrespondierenden Aufbauorganisation – auch Organisationsstruktur genannt – existieren. Also erfordert die Konzeptualisierung von Geschäftsprozessen auch die Berücksichtigung des jeweiligen aufbauorganisatorischen Kontextes. Es bleibt zu ergänzen, dass organisatorische Regelungen nicht für einen konkreten Geschäftsprozess getroffen werden, sondern für eine Klasse gleichartiger Prozesse, die auch *Geschäftsprozestyp* genannt wird.
- Die organisatorische Analyse und die Gestaltung eines Geschäftsprozesses erfordert *Ziele*. Neben den operativen Zielen, auf die ein Geschäftsprozess auszurichten ist, ist dabei auch an übergeordnete strategische Ziele zu denken, da diese u.U. auch einen Spielraum für eine Variation der operativen Ziele eröffnen.
- Die Durchführung von Geschäftsprozessen erfordert den Einsatz von *Ressourcen*. Die wirtschaftliche Gestaltung von Geschäftsprozessen empfiehlt deshalb eine angemessene Erfassung und Bewertung von Ressourcen.

Für unsere Untersuchung ergibt sich damit folgende Begrifflichkeit:

Ein *Geschäftsprozess* ist eine wiederkehrende Abfolge von Aktivitäten, die mehr oder weniger rigiden Regelungsmustern genügt. Er ist zielgerichtet und steht in einem direkten Zusammenhang mit der marktgerichteten Leistungserstellung eines Unternehmens¹. Solche Prozesse, die unmittelbar und in bedeutsamem Umfang zur Erzeugung von Erträgen führen und gleichzeitig wettbewerbsrelevante Kompetenzen eines Unternehmens fordern, bezeichnet man als *Kernprozesse*. Die Ausführung von Geschäftsprozessen erfordert den Einsatz knapper Ressourcen. Ein *Geschäftsprozestyp* beschreibt eine Klasse gleichartiger Geschäftsprozesse.

Ein *Geschäftsprozessmodell* ist eine zweckgerichtete Abstraktion eines Geschäftsprozestyps, die häufig – aber nicht notwendig – mit einer grafischen Darstellung einhergeht.

¹ Das schließt z.B. Prozesse wie Gebäudereinigung oder die Essenszubereitung in der Kantine i.d.R. aus.

Wie sich noch zeigen wird, sind diese Begriffe im Hinblick auf die Evaluation von Modellierungswerkzeugen noch weiter zu differenzieren. Der hier gewählte Begriff von Geschäftsprozess schließt Produktionsprozesse in Industriebetrieben nicht aus. Dennoch liegt der Fokus der Betrachtung üblicherweise auf Prozessen im Büro- und Verwaltungsbereich.

Auch wenn die Begriffe „Workflow“ und „Geschäftsprozess“ mitunter synonym verwendet werden, erscheint eine Differenzierung sinnvoll. Wir verstehen unter einem Workflow eine Abstraktion eines Geschäftsprozesses, die vor allem auf den Fluss digitalisierter Dokumente bzw. Objekte gerichtet ist. Menschliche Aktivitäten bzw. Entscheidungen im Rahmen eines Geschäftsprozesses werden dabei weitgehend ausgeklammert bzw. auf Interaktionen mit Anwendungssystemen reduziert. Ein Workflow-Modell stellt einen Workflow-Typ (als Klasse gleichartiger Workflows) dar. Dabei sollte die Darstellung den Anforderungen der letztlich beabsichtigten Automatisierung genügen und kann deshalb Aspekte enthalten, auf die in einem korrespondierenden Geschäftsprozessmodell verzichtet wird. Ein Workflow Management System (WFMS, auch: Vorgangssteuerungssystem) ist ein System zur Ausführung und Kontrolle eines Workflows unter Verwendung vorhandener Software und ggfs. eigener Anwendungsteile. Idealtypisch dient dabei ein Workflow-Modell als Ausführungsschema. Die Leistungsfähigkeit vieler Workflow-Systeme ist allerdings bescheiden. Ihre Funktionalität beschränkt sich auf das Starten und Terminieren von Anwendungen und die Nutzung elementarer Benachrichtigungsdienste. Abbildung 1 veranschaulicht den Zusammenhang zwischen Geschäftsprozessmodellen und Workflow-Modellen.

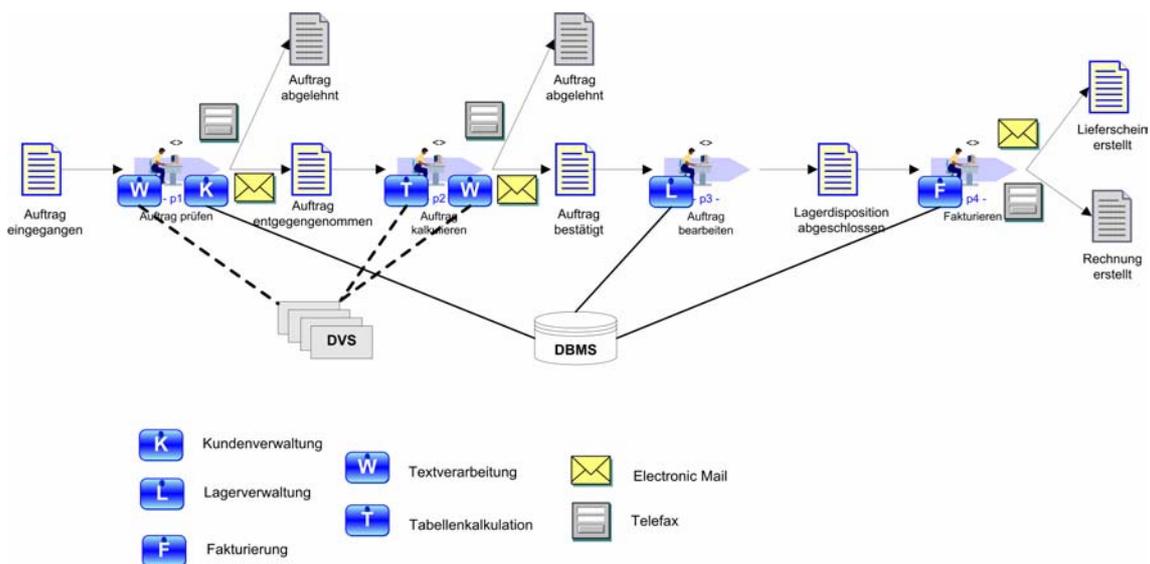


Abbildung 1: Verhältnis von Geschäftsprozessen und Workflows

Ein Modell ist eine sprachliche Konstruktion. Seine Erstellung erfordert deshalb eine *Modellierungssprache*. Eine Modellierungssprache besteht aus einer Menge von Symbolen sowie einer Syntax, die deren zulässige Anordnung beschreibt, und einer Semantik. Dabei kann zwischen der *abstrakten* und der *konkreten* Syntax unterschieden werden. Während die abstrakte Syntax sich auf die Beschreibung der verfügbaren Symbolmengen sowie den darauf aufbauenden Anordnungsregeln beschränkt, wird mit der konkreten Syntax – auch Notation genannt – das konkrete Aussehen der Symbole festgelegt. Eine Modellierungssprache hat eine abstrakte Syntax, kann aber mehrere Notationen aufweisen.

Die Semantik beschreibt einerseits die Bedeutung der verfügbaren Symbole, z.B. durch eine (extensionale) Festlegung der durch sie repräsentierten Mengen, andererseits kann sie die Syntax ergänzen, indem sie syntaktisch korrekte Konstruktionen als semantisch unzulässig kennzeichnet. Bei vereinfachter Betrachtung² können drei Klassen von Modellierungssprachen unterschieden werden: informale, semi-formale und formale. Eine formale Modellierungssprache besteht aus einer eindeutig festgelegten Menge von

² In Kapitel 3 findet sich eine detaillierte Betrachtung der Eigenschaften von Modellierungssprachen.

Symbolen, deren Semantik ebenfalls eindeutig festgelegt ist. Die zulässige Verwendung der Symbole zur Erstellung von Modellen wird durch eine eindeutige Syntax und gegebenenfalls ergänzende eindeutige semantische Integritätsbedingungen präzise festgelegt. Unter einer semi-formalen Modellierungssprache verstehen wir eine Sprache mit einer eindeutig festgelegten Symbolmenge und einer wenigstens in Teilen eindeutig festgelegten Syntax. Demgegenüber gibt es allenfalls eine rudimentäre Beschreibung der Semantik. Eine informale Modellierungssprache schließlich stellt lediglich eine Menge von Symbolen bereit. Zu Syntax und Semantik werden keine expliziten Angaben gemacht. Sie ergeben sich allenfalls indirekt – und i.d.R. mehrdeutig – durch Interpretationen der Betrachter entsprechender Modelle.

2.2. Zur wirtschaftlichen Bedeutung der Geschäftsprozessmodellierung

Die sorgfältige Modellierung von Geschäftsprozessen bzw. von Workflows erfordert einen erheblichen Aufwand. Dabei ist einerseits an den erforderlichen Personaleinsatz zu denken, andererseits an die Verwendung unterstützender Werkzeuge. Das legt die Frage nahe, welcher Nutzen von der Geschäftsprozessmodellierung zu erwarten ist. Tatsächlich gibt es eine Reihe wichtiger Gründe dafür, den Entwurf, die Verwaltung und Pflege von Geschäftsprozessmodellen nicht nur als eine Aufgabe mit hoher Priorität, sondern als eine Kernaufgabe der Unternehmensführung anzusehen. Sie betreffen die Gestaltung leistungsfähiger betrieblicher Informationssysteme, das betriebliche Wissensmanagement, die zielgerichtete Kommunikation, das Controlling und auch die Unternehmenskultur.

Jenseits mancher Überzeichnungen in der einschlägigen Literatur³ scheint es unstrittig, dass die Thematisierung von Geschäftsprozessen geeignet ist, eine für gemeinsames Handeln wichtige *gemeinsame Orientierung* zu schaffen, die gleichzeitig für ökonomische

³ Hammer und Champy tun sich dabei besonders hervor. Ein Beispiel von vielen: „How people and companies did things yesterday doesn't matter to the business reengineer.“ ([HaCh93], S. 2)

misch bedeutsame Erfolgsfaktoren sensibilisiert. Hier ist einerseits daran zu denken, dass die neuere Diskussion der Reorganisation von Geschäftsprozessen i.d.R. mit einer ausgeprägten Kunden- bzw. Marktorientierung verbunden ist. Auf abstrakterem Niveau bedeutet das, dass die Zielgerichtetheit von Tätigkeiten betont wird – was gewiss nicht in allen Unternehmen selbstverständlich ist. Andererseits wird durch die Forderung nach Analyse und Reorganisation von Geschäftsprozessen die stetige Verbesserung von Abläufen nahegelegt. Das dürfte zumindest dafür empfängliche Mitarbeiter dazu motivieren, nach Möglichkeiten zur Erhöhung der Prozesseffizienz – und damit zur Wirtschaftlichkeit des Unternehmens – Ausschau zu halten.

Die Kommunikation zwischen verschiedenen Bereichen eines Unternehmens leidet häufig unter Missverständnissen. Besonders ausgeprägt sind solche Sprachbarrieren i.d.R. zwischen Systementwicklern und Vertretern von Fachabteilungen. Aber auch Mitglieder der Unternehmensleitung und Mitarbeiter auf der operativen Ebene sprechen nicht immer die gleiche Sprache, was nicht zuletzt auf unterschiedliche Aggregations- bzw. Abstraktionsniveaus zurückzuführen ist. Modelle von Geschäftsprozessen versprechen hier insofern Abhilfe als sie eine Sicht auf das Unternehmen betonen, die von allen Mitarbeitern – wenn auch in unterschiedlicher Fokussierung und Detaillierung – geteilt werden kann. Geschäftsprozessmodelle können damit gleichsam die Funktion von *Schnittstellen* zwischen verschiedenen Bereichen eines Unternehmens übernehmen. Darüber hinaus können sie auch als Schnittstelle nach außen dienen. Hier ist z.B. an die Gestaltung unternehmensübergreifender Abläufe zusammen mit Vertretern anderer Unternehmen zu denken.

Der *Entwurf leistungsfähiger betrieblicher Informationssysteme* empfiehlt einerseits eine enge Integration der in verschiedenen Anwendungen genutzten Daten, andererseits eine möglichst reibungsfreie Einbettung in die zu unterstützenden organisatorischen Abläufe. Geeignete Geschäftsprozessmodelle unterstützen beide Ziele. So hat sich gezeigt, Domänenexperten zumeist Geschäftsprozesse beschreiben können, während die unmittelbare Frage nach den im Unternehmen benötigten Informationen keine befriedigenden Antworten erwarten lässt. Wenn aber auf der Grundlage eines Prozessmodells

nach den in diesem Prozesstyp benötigten Informationen gefragt wird, wird eine systematische Erhebung des Informationsbedarfs wirksam unterstützt. Deshalb ist die Betrachtung von Abläufen gut geeignet, um in der Analysephase zusammen mit prospektiven Anwendern Informationen sowie die jeweils zu deren Ablage verwendeten Medien zu erfassen. Auf dieser Grundlage kann einerseits eine Analyse der Effizienz von Prozessen durchgeführt werden – indem etwa Medienbrüche ermittelt werden, andererseits können so schrittweise korrespondierende Daten- bzw. Objektmodelle erstellt werden. Die auf diese Weise entstehenden anwendungsübergreifenden semantischen Referenzmodelle stellen eine gute Grundlage für die Integration der zu entwickelnden Anwendungen dar. Im Hinblick auf die organisatorische Integration betrieblicher Informationssysteme sind Prozessmodelle ebenfalls gut geeignet, weil sie Anwendungen bzw. Sichten auf das Informationssystem im Kontext betrieblicher Aufgaben beschreiben.

Eine wachsende Zahl von Unternehmen hat erkannt, dass mit der zunehmenden Dynamik der Märkte, immer kürzeren Entwicklungszyklen und einem ausgeprägten Wettbewerb um qualifizierte Mitarbeiter das *betriebliche Wissensmanagement* ein hohes Maß an Aufmerksamkeit verdient. Dabei handelt es sich um Verfahren zur Erfassung, Aufbereitung, Verwaltung und Vermittlung von Wissen, das für erfolgreiches Handeln im Unternehmen von Bedeutung ist (vgl. [PrRa+97]). Während dazu häufig strukturierte Dokumente verwendet werden, bieten Geschäftsprozessmodelle, die ggfs. durch multimediale Annotationen angereichert sind, eine gute Ergänzung. So versprechen sie eine anschauliche Darstellung des relevanten Wissens bei gleichzeitiger Betonung einer Ziel- bzw. Kundenorientierung (s.o.). Gleichzeitig unterstützen sie durch geeignete Verknüpfungen mit Darstellungen der Unternehmensstrategie und des Informationssystems die Vermittlung grundlegender Zusammenhänge im Unternehmen ([ScFr01]).

Die Gestaltung von Geschäftsprozessstypen beruht auf einer Reihe von Annahmen. Dazu gehören einerseits die Effizienz der jeweils vorgesehenen Arbeitsteilung und der zugehörigen Koordinationsmaßnahmen, andererseits – durchaus damit zusammenhängend – Anzahl und Umfang entsprechender Prozessinstanzen im Zeitverlauf sowie die Verfügbarkeit und Qualität von Ressourcen. Da diese Annahmen mit Risiken behaftet sind,

empfiehlt sich ihre Überprüfung und ggfs. die Reorganisation der betroffenen Geschäftsprozessstypen – die auch die Redefinition von Zielen beinhalten kann. Insofern ist die Analyse von Geschäftsprozessen eine wichtige Aufgabe des *Controlling*. Im Hinblick auf die Wirtschaftlichkeit von Geschäftsprozessen ist dabei nicht zuletzt an eine verursachungsgerechte Erfassung von Kosten und Leistungen zu denken. Die Prozesskostenrechnung bietet dazu eine Reihe von Verfahren (vgl. [CoKa91], [HoKi+93], [Webe00]), die letztlich entsprechend angereicherte Geschäftsprozessmodelle erfordern.

3. Generelle Anforderungen an Modellierungssprachen

Das Erstellen eines Geschäftsprozessmodells erfordert eine Modellierungssprache. Die Beurteilung von Modellierungssprachen stellt eine erhebliche Herausforderung dar (vgl. dazu [Fran98] und [FrPr97]). Die im Folgenden dargestellten Kriterien lassen zum Teil einen erheblichen Interpretationsspielraum zu. Ihre Anwendung empfiehlt deshalb eine sorgfältige und differenzierte Betrachtung der jeweiligen Sprache. Bevor wir auf spezifische Anforderungen an Konzepte zur Modellierung von Geschäftsprozessen bzw. Workflows eingehen, stellen wir Anforderungen dar, die grundsätzlich von Modellierungssprachen erfüllt werden sollten.

Modellierungssprachen können für eine große Bandbreite unterschiedlicher Zwecke entworfen werden. Für alle Sprachen können generell drei Arten von Anforderungen formuliert werden⁴, die nicht immer unabhängig voneinander sind:

- *Formale* Anforderungen sind dann von besonderer Wichtigkeit, wenn eine (maschinelle) Überprüfung der Integrität von Modellen, die Transformation von Modellen oder die Berechnung von Modelleigenschaften erwünscht sind (vgl. Abschnitt 3.1).
- *Anwenderbezogene* Anforderungen betreffen das Verhältnis des Modellierers zu den bereitgestellten Konzepten und deren Visualisierung (vgl. Abschnitt 3.2).
- *Anwendungsbezogene* Anforderungen sind auf solche Eigenschaften einer Modellierungssprache gerichtet, die allgemein einen Bezug zu Modellierungszwecken und Modellierungsdomänen haben (vgl. Abschnitt 3.3).

⁴ In [FrPr97] findet sich ebenfalls eine umfassende Diskussion von Anforderungen an Modellierungssprachen vor dem Hintergrund der Bewertung objektorientierter Modellierungssprachen.

Alle im Folgenden betrachteten Anforderungen sind für sich betrachtet sinnvoll. Das schließt allerdings Konflikte mit anderen Anforderungen nicht aus. Es ist also nicht in jedem Fall unbedingt angeraten, einzelne Anforderungen maximal zu erfüllen.

3.1. Formale Anforderungen

In der Informatik kommt den formalen Eigenschaften von (Modellierungs-) Sprachen eine herausragende Rolle zu. Dieser Umstand ist darin begründet, dass die formalen Eigenschaften einer Modellierungssprache die Möglichkeiten von Verfahren zur Analyse und Transformation der jeweils zugehörigen Modelle wesentlich prägen.

3.1.1. Korrektheit und Vollständigkeit

Die Spezifikation einer Sprache genügt dem Kriterium der Korrektheit, wenn sie die eindeutige Identifikation (syntaktisch und semantisch) unzulässiger Modelle gewährleistet und gleichzeitig erlaubt, die Menge aller zulässigen Modelle zu generieren. Die Sprachbeschreibung sollte zudem vollständig sein, d.h. alle in der Sprache verwendeten Konzepte sowie die Bedingungen ihrer Verwendung sollten eindeutig definiert sein (vgl. [SüEb97], S. 2 f.). Dabei ist zu berücksichtigen, dass es sich hier lediglich um eine formale Anforderung handelt, nicht darum, die zu beschreibenden Konzepte möglichst detailliert oder anwendungsnah zu beschreiben. Einige Anforderungen sind nur schwer formalisierbar: So wäre beispielsweise ein Konzept wie „Kundenorientierung“ nur dann formal eindeutig definierbar, wenn es als boolescher Wert abgebildet werden könnte.

3.1.2. Einheitlichkeit und Redundanzfreiheit

Eine Sprachspezifikation sollte auch dem Prinzip der Einheitlichkeit bzw. Klarheit genügen. Das umfasst allgemein die Forderung nach einer verständlichen Darstellung der Sprachbeschreibung, konkret die Forderung danach, dass ähnliche Konzepte auch in ähnlicher Weise spezifiziert werden. Schließlich sollte eine Sprachspezifikation so einfach wie möglich dargestellt werden, was u.a. die Vermeidung redundanter Informationen empfiehlt.

3.1.3. Wiederverwendbarkeit und Wartbarkeit: Abstraktion

Das Prinzip der Wiederverwendbarkeit wird schon seit langem in der Softwareentwicklung beachtet, um die Komplexität und den Aufwand bei der Erstellung von Informationssystemen zu reduzieren. Auch Unternehmensmodelle zeichnen sich durch eine hohe Komplexität aus. Daher ist die Wiederverwendbarkeit von bereits modellierten Informationen (in unserem Fall Geschäftsprozessen) bereits von der Sprache zu unterstützen. Wiederverwendet werden können beispielsweise

- Ablaufstrukturen
- Teilmodelle
- Beziehungen zwischen Teilmodellen

Um die Wiederverwendung und Modifikation von Modellen zu unterstützen, sind Sprachkonzepte erforderlich, die es gestatten, Sachverhalte auf einem hohen Abstraktionsniveau zu beschreiben. Indem man von spezifischen Eigenschaften der abzubildenden Sachverhalte bewusst abstrahiert, wird es leichter möglich, diese Teile in anderen Kontexten zu verwenden. Gleichzeitig wird die Wartbarkeit von Modellen unterstützt, weil Änderungen in Modellteilen, von denen abstrahiert wird, die Konsistenz der übrigen Teile eines Modells nicht beeinflussen.

Es gibt eine Reihe bewährter Sprachkonzepte, die die Abstraktion von Modellen fördern:

- *Klassenbildung* erlaubt die Festlegung von Eigenschaften, die für alle Instanzen einer Klasse gelten. Auf diese Weise kann also von konzeptionellen Eigenschaften der Instanzen abstrahiert werden. Die Erzeugung einer neuen Instanz ist damit gleichbedeutend mit der Wiederverwendung der zuvor definierten Klasseneigenschaften. Gleichzeitig wird die Spezifikation und Wartung von Systemen vereinfacht, weil sich der betroffene Code auf die Beschreibung der Klasse beschränkt und die zugehörigen Instanzen im günstigsten Fall nicht berücksichtigt werden müssen.
- „*Information Hiding*“ bezeichnet das gezielte Verbergen von Konkretisierungen einzelner Modellbestandteile. Ein Beispiel dafür liefert die Kapselung von Ob-

jekten. Der Zustand eines Objekts ist von außen nicht direkt zugreifbar. Ebenso ist die Implementierung von Operationen nicht einzusehen. „Information hiding“ unterstützt die Wiederverwendung von Modellen: Durch die gezielte Abstraktion von speziellen Eigenschaften wird die Chance erhöht, dass ein Modell in unterschiedlichen Kontexten anwendbar ist. Gleichzeitig wird die Wartbarkeit unterstützt, weil die Änderung verborgener Modellteile keine Seiteneffekte mit sich bringt.

- *Generalisierung/Spezialisierung* sind essentielle Sprachkonzepte zur Reduktion von Komplexität. Spezialisierung unterstützt die Wiederverwendung von Modellteilen, da durch Hinzufügen weiterer Eigenschaften eine Anpassung an ähnliche Gegebenheiten ermöglicht wird. Gleichzeitig erleichtert Generalisierung die Wartung von Modellen, da sich Änderungen im günstigsten Fall auf die generalisierten Modellteile beschränkt. Im Hinblick auf die Bewertung einschlägiger Modellierungssprachen ist hier allerdings ernüchternd festzustellen, dass es bisher nur wenige Ansätze (z.B. [AaBa97], [WyLe02], [MaCr+99]), für eine Spezifikation von Generalisierungs- bzw. Spezialisierungsbeziehungen zwischen Prozesstypen gibt, die zudem nicht völlig überzeugen ([FrLa02]).

3.2. Anwenderbezogene Anforderungen

Die spezifischen Anforderungen eines Anwenders an eine Modellierungssprache sind auch abhängig davon wie häufig und intensiv er die Sprache benutzt. Daneben sind auch die Betrachter von Modellen, die selbst nicht modellieren, also die Sprache nicht unmittelbar benutzen, nicht völlig zu vernachlässigen.

3.2.1. Einfachheit

Je einfacher eine Sprache ist, desto weniger Fehler sind bei der Modellierung zu erwarten. Auch wenn eine exakte Definition dieses Kriteriums nicht möglich ist, kann Einfachheit tendenziell so charakterisiert werden: Ceteris paribus ist eine Sprache umso einfacher, je geringer die Zahl ihrer Konzepte sowie der sie darstellenden Symbole und je geringer die Zahl der Regeln zur Festlegung ihrer Syntax und Semantik sind.

3.2.2. Verständlichkeit und Anschaulichkeit

Vordergründig gilt, dass eine Modellierungssprache umso verständlicher ist, je einfacher sie ist. Bei näherer Betrachtung wird allerdings deutlich, dass Einfachheit in dem skizzierten Sinn nicht hinreicht, um Verständlichkeit zu garantieren. Vielmehr hängt die Verständlichkeit einer Sprache wesentlich davon ab, wie sehr sie Sprachen ähnelt, die dem Anwender bekannt sind. Eine verständliche Modellierungssprache ist also dadurch gekennzeichnet, dass ihre Konzepte und Symbole direkt mit Begriffen korrespondieren, die dem Anwender vertraut sind. Es liegt auf der Hand, dass dieses Kriterium mit dem Betrachter variiert. Da nicht nur Begrifflichkeiten, sondern auch der Modellierungszweck mit verschiedenen Gruppen von Betrachtern schwanken – so interessieren etwa einen Organisator andere Aspekte als einen Controller oder einen Software-Entwickler – wird die Verständlichkeit von Modellen prinzipiell dadurch gefördert, dass die Modellierungssprache verschiedene Sichten auf ein Modell unterstützt (vgl. [Fran02]). Dazu gehört die Bereitstellung zielgruppenspezifischer Konzepte. In der Geschäftsprozessmodellierung auf hoher Abstraktionsebene können dies betriebswirtschaftliche Fachtermini sein, während bei der Workflow-Modellierung und der Modellierung betrieblicher Informationssysteme die Fachtermini und Konzepte aus der Software-Technik erforderlich sind. In diesem Zusammenhang ist auch zu fordern, dass die Beschreibung von Prozesseigenschaften nicht die Verwendung artifiziell wirkender Konzepte nötig macht, die nicht im Einklang mit gängigen Betrachtungsweisen der Anwender sind. Das Beispiel (in Anlehnung an [Aals01]) in Abbildung 2 und Abbildung 3 verdeutlicht beide Aspekte. Die in der Abbildung dargestellten Petrinetze beschreiben Prozesse zur Prüfung eines Antrags auf Schadenregulierung bei einem Versicherungsunternehmen. Auch wenn die formale Semantik des Modells eindeutig ist – es handelt sich um ein Stellen-Transitionen-Netz – ist seine Interpretation durch einen nicht einschlägig qualifizierten Betrachter mit Tücken behaftet. Das liegt daran, dass die für Stellen und Transitionen verwendeten Bezeichner nicht eindeutig klar machen, ob es sich jeweils um ein Ereignis oder einen Teilprozess handelt. Auch wenn die Wahl der Bezeichner i.d.R. nicht durch die Modellierungssprache vorgeben ist, bleibt hier kritisch anzumerken, dass die ver-

wendeten Symbole keinen Hinweis darauf geben, wie solche Mehrdeutigkeiten aufzulösen sind.

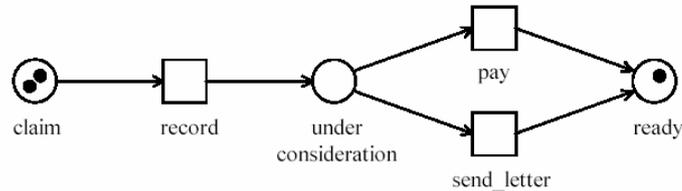


Abbildung 2: Beispiel für Anschaulichkeit

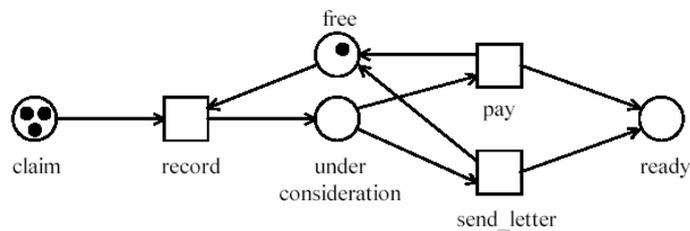


Abbildung 3: Beispiel für eingeschränkte Anschaulichkeit

Das in der Abbildung oben dargestellte Netz erlaubt es, dass mehrere Anträge gleichzeitig geprüft werden (mit anderen Worten: dass die Stelle „under consideration“ ggfs. mehrere Marken enthält). Das modifizierte Netz im unteren Teil der Abbildung stellt sicher, dass jeweils nur ein Antrag geprüft werden kann. Dazu wird eine weitere Stelle („free“) eingeführt, die lediglich den Zweck hat, das Schalten der Transition „record“ so zu limitieren, dass jeweils nur eine Marke an die Stelle „under consideration“ gereicht werden kann. Es liegt auf der Hand, dass eine solche Erweiterung – obwohl sie einen einfachen Sachverhalt beschreibt – schwer verständlich ist.

Neben den bereitgestellten Konzepten und der abstrakten Syntax hat auch die Notation einer Modellierungssprache einen Einfluss auf die Verständlichkeit und Anschaulichkeit von Modellen. Dazu schlägt Rumbaugh (vgl. [Rum96]) eine Liste von Anforderungen an die Notation von Modellierungssprachen vor:

- klare Abbildung der Konzepte auf Symbole,
- nicht zu viele Symbole,
- die Notation muss auch leicht per Hand gezeichnet werden können,
- sie muss gedruckt gut aussehen und sich gut faxen und auf Schwarz-Weiß-Kopierern kopieren lassen,

Während die ersten drei Forderungen plausibel erscheinen, ist die letzte u.E. mit einer einschränkenden Anmerkung zu versehen: Wenn eine farbige Darstellung geeignet erscheint, die Anschaulichkeit eines Modells zu fördern, wird der daraus resultierende Nachteil für schwarz-weiße Darstellungen u.U. deutlich überkompensiert. Grundsätzlich gilt auch für die Notation einer Modellierungssprache, dass die Anschaulichkeit umso größer ist, je ähnlicher die Symbole und ihre Anordnung dem Benutzer bereits vertrauten Darstellungen sind.

3.3. Anwendungsbezogene Anforderungen

Die Anforderungen, die sich aus der Sicht des Anwenders ergeben, sind i.d.R. nicht unabhängig von der jeweiligen Anwendungsdomäne, also den zu modellierenden Gegenständen und den Modellierungszwecken. Dabei ist zu beachten, dass anwenderbezogene Anforderungen teilweise mit anwendungsbezogenen Anforderungen konkurrieren. So bietet eine einfach zu erlernende Modellierungssprache u.U. nicht die Mächtigkeit, die im Hinblick auf die Modellierungszwecke wünschenswert ist.

3.3.1. Mächtigkeit und Angemessenheit

Eine einschlägige Modellierungssprache sollte einen Modellierer in die Lage versetzen, alle als relevant erachteten Eigenschaften eines darzustellenden Sachverhalts in der jeweils gewünschten Detaillierung und Präzision zu beschreiben. Je besser eine Modellierungssprache dieses Ziel unterstützt, desto größer ist ihre Mächtigkeit. Dabei handelt es sich aber offensichtlich nicht um eine präzise Größe, die einer intervallskalierten Messung zugänglich wäre. Gleichzeitig sind wir auch nicht in der Lage, mit Gewissheit zu sagen, welche Konzepte die ggfs. vollständige Beschreibung von Anwendungsdomä-

nen, wie etwa von Geschäftsprozessen, erfordert. Auf Programmiersprachen wird ein eindeutiges Kriterium zur Feststellung der Vollständigkeit angewandt: Berechenbarkeit. Alle Turing-mächtigen Programmiersprachen sind damit vollständig und gleich mächtig. Nun kann man ein formales Vollständigkeitskriterium auch auf Sprachen zur Prozessmodellierung anwenden, wie es ja durchaus auch getan wird – etwa bei Petri-Netzen. Im Hinblick auf die Modellierung von Geschäftsprozessen ist ein solcher Ansatz allerdings nur bedingt geeignet. Das hat mehrere Gründe. Zum einen erfordert die Beschreibung von Geschäftsprozessen u.U. Konzepte, die sich gegen eine formale Spezifikation mit hinreichender Semantik sperren. Hier ist etwa an die Beschreibung der Qualifikation von Mitarbeitern oder allgemein an Zielvorgaben wie ‚Wettbewerbsfähigkeit‘ oder ‚Kostenbewusstsein‘ zu denken. Eine formal vollständige Beschreibungssprache allein garantiert hier keine vollständige Beschreibung, während eine semi-formale oder natürlichsprachliche Modellierung dem menschlichen Betrachter eine informativere und damit ‚vollständigere‘ Darstellung bietet. Daneben gibt es Modellierungszwecke – etwa eine betriebswirtschaftlich ausgerichtete Prozessanalyse – die gar keine formale Beschreibung erfordern.

Dabei ist der Umstand zu berücksichtigen, dass die Konzepte einer Sprache zur Modellierung von Geschäftsprozessen *angemessen* sein sollten. Angemessenheit betrifft das Abstraktionsniveau, den Detaillierungs- und den Formalisierungsgrad. Eine Modellierungssprache bietet ein angemessenes Abstraktionsniveau, wenn ihre Konzepte eine problemadäquate Strukturierung des zu modellierenden Sachverhalts unterstützen. Das bedeutet, dass die Konzepte weitgehend mit den Begriffen einer differenzierten, domänen- und problemgerechten Fachterminologie korrespondieren sollten. Grundformen von Petri-Netzen erfüllen diese Anforderung offenbar nicht. Daneben sollten die Konzepte eine Detaillierung ermöglichen, wie sie nach Maßgabe des jeweiligen Modellierungszwecks wünschenswert erscheint. Umgekehrt empfiehlt das Kriterium ‚Angemessenheit‘, dass eine Modellierungssprache den Modellierer nicht mit unnötigen Detaillierungen belastet. Ähnliches gilt für die Präzision einer Modellierungssprache: Eine präzise, im Grenzfall formale Beschreibung kann für bestimmte Modellierungsziele erfor-

derlich sein, der Zwang zu formaler Präzision kann aber auch für andere Modellierungszwecke dysfunktional sein.

Zusammenfassend können wir festhalten, dass Mächtigkeit und Angemessenheit wichtige Qualitätsmerkmale von Modellierungssprachen darstellen. Gleichzeitig lassen sie sich nicht durch präzise beschreibbare Kriterien beschreiben. Vielmehr sind für eine Klasse von Modellierungssprachen und korrespondierende Modellierungszwecke konkrete Anforderungen an Mächtigkeit und Angemessenheit zu formulieren.

3.3.2. Operationalisierbarkeit

Häufig dienen Modelle nicht nur der Visualisierung eines Sachverhalts, sondern bilden die Grundlage für diverse Analysen und Transformationen. Im Kontext der konzeptionellen Modellierung ist hier vor allem an die Verwendung von Modellen für die Software-Entwicklung zu denken. Daneben kann ein Modell als Grundlage für software-technisch oder betriebswirtschaftlich motivierte Analysen dienen. In diesem Zusammenhang ist außerdem an die Möglichkeit zu denken, Modelle als Grundlage für Simulationen zu verwenden. Die Eignung eines Modells für diverse Analyse- oder Transformationsverfahren wird wesentlich durch die verwendete Modellierungssprache festgelegt. So setzen etwa betriebswirtschaftliche Analysen von Geschäftsprozessmodellen Konzepte voraus, die der Abbildung von Kosten dienen. Eine software-technische Umsetzung, etwa in Form der Generierung von Programmcode aus einem Modell, erfordert die Abbildbarkeit der Konzepte der Modellierungssprache auf korrespondierende Konzepte der Implementierungssprache. Im für die Operationalisierung günstigsten Fall entsprechen die Modellierungskonzepte weitgehend den Implementierungskonzepten. Simulation erfordert spezielle Sprachkonzepte, da die konzeptionelle Beschreibung eines Modells durch die Beschreibung von Instanzenmengen ergänzt werden muss, die für eine Simulation benötigt werden.

4. Geschäftsprozessmodellierung

Jede Modellierungssprache beinhaltet Begriffe (wir können auch sagen: Konzepte). Im Fall der Geschäftsprozessmodellierungssprachen dienen diese Begriffe der Beschreibung von Geschäftsprozessen. Im Hinblick auf die Anforderungen an solche Sprachen ist es deshalb angeraten, den in 2.1 eingeführten, vorläufigen Begriff eines Geschäftsprozessstyps weiter zu differenzieren. Dazu betrachten wir im folgenden eine Reihe von Begriffen, die für die detaillierte Beschreibung von Geschäftsprozessstypen benötigt werden.

4.1. Grundlegende Anforderungen an einschlägige Modellierungssprachen

Es gibt eine Reihe grundlegender Anforderungen, die für viele Sprachen zur konzeptionellen Modellierung gelten. Im folgenden betrachten wir diese Anforderungen im Hinblick auf Sprachen zur Modellierung von Geschäftsprozessen.

4.1.1. Abstraktionsebenen

In der Alltagssprache werden Begriffe häufig mehrdeutig verwendet. Das gilt nicht zuletzt für die mit einem Begriff verbundene Abstraktion. Ein naheliegendes Beispiel aus dem Kontext dieses Arbeitsberichts ist der Begriff „Geschäftsprozess“. Er wird i.d.R. sowohl für die Bezeichnung eines Typs bzw. einer Klasse von Geschäftsprozessen verwendet (z.B. Geschäftsprozess „Auftragsbearbeitung“) wie auch für die Bezeichnung einer bestimmten Instanz. Im Hinblick auf die Erstellung von Modellen, die von Werkzeugen analysiert und ggfs. weiter verarbeitet werden, sind solche Mehrdeutigkeiten nicht zu akzeptieren. Die mit einem Begriff zu verbindenden Abstraktionsebenen sind deshalb sorgfältig zu unterscheiden. Dabei zeigt sich, dass die Anzahl der Abstraktionsebenen mit dem Begriff variieren kann. Betrachten wir zunächst den zentralen Begriff „Geschäftsprozess“. Wie bereits erwähnt, sind hier im wesentlichen zwei Bedeutungen

zu unterscheiden: der Typ oder die Klasse und die Instanz. Der Begriff „Klasse“ wird nicht einheitlich verwendet. Der *intensionale* Klassenbegriff, der mit dem Typbegriff vergleichbar ist, beschreibt die Struktur, die allen Instanzen dieser Klasse gemein ist. Ein Beispiel wäre die Struktur „Nachname“, „Vorname“, „Geschlecht“ für die Klasse „Person“. In *extensionaler* Sicht wird eine Klasse als eine Menge gleichartiger Instanzen aufgefasst. Diese Unterscheidung hat erhebliche Konsequenzen, etwa im Hinblick auf die Semantik von Spezialisierungsbeziehungen: So folgt aus dem extensionalen Klassenbegriff, dass eine Instanz einer Unterklasse gleichzeitig Instanz einer korrespondierenden Oberklasse ist, während der intensionale Klassenbegriff impliziert, dass eine Instanz nur genau einer Klasse zugeordnet ist.

Abbildung 4 zeigt die Abstraktionsebenen, auf denen der Begriff „Geschäftsprozess“ angesiedelt sein kann. Für jede Abstraktionsebene sind beispielhafte Konzepte in einer an das Entity Relationship Modell angelehnte Notation dargestellt.

Abstraktionsniveau	Beispiel
Meta Sprache zur Modellierung von Geschäftsprozessen	$0,^*$ Process_Type $1,1$ $0,^*$ Ressource_Type <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">name: String</div> <div style="border: 1px solid black; padding: 2px;">name: String volume: Integer</div> </div>
Typ Geschäftsprozesstyp	<div style="border: 1px solid black; padding: 5px;"> Auftragsbearbeitung Eingang Date Start Time End Time ... </div>
Instanz konkreter Geschäftsprozess	<div style="border: 1px solid black; padding: 5px;"> Auftragsbearbeitung Eingang 15.08.2001 Start 09:15 (15.08.2001) End - ... </div>

Abbildung 4: Geschäftsprozessmodellierung Abstraktionsebenen

Bei näherer Betrachtung dieser Abstraktionsebenen wird deutlich, dass sie nicht hinreichen, um alle benötigten Konstellationen abzudecken. Das gilt in zweifacher Hinsicht: zum einen für den Prozessbegriff selbst, zum anderen für die verwendeten Ressourcen, deren Konkretisierung in Abbildung 4 nicht ohne Grund ausgespart blieb. Betrachten wir zunächst den Prozessbegriff. Für die Modellierung eines Geschäftsprozessstyps sind folgende Angaben u.U. von Bedeutung: eine ergänzende natürlichsprachliche Beschreibung des Typs, die durchschnittliche Anzahl von Instanzen pro Monat oder die Anzahl der aktiven Instanzen. Diese Angaben können nicht auf der Ebene eines Geschäftsprozessstyps erfasst werden, da ein Typ ja nur solche Eigenschaften repräsentiert, die für alle Instanzen gelten. Hier handelt es sich aber um Eigenschaften, die den Typ (bzw. die Klasse) betreffen. Das erfordert eine weitere Abstraktionsebene, nämlich eine Klasse in extensionaler Bedeutung, die also Eigenschaften der Menge ihre Instanzen beschreibt.

Ressourcen (s. Abbildung 5) werden in Prozessen benutzt bzw. verbraucht. Es handelt sich also um Maschinen, Rohstoffe, Verbrauchsmaterial etc.⁵ Bei der Beschreibung von Ressourcen sind u.U. weitere Abstraktionsebenen zu berücksichtigen. Betrachten wir dazu die Ressource „Laserdrucker“, die von einem Geschäftsprozessstyp benutzt wird.

⁵ Eine ausführliche Betrachtung von Ressourcen findet sich in Abschnitt 4.2.4.

Abstraktionsniveau	Beispiel													
Meta														
Modellierungssprache	<table border="1"> <tr> <td>Resource_Type</td> <td></td> <td>Feature_Type</td> </tr> <tr> <td>name String</td> <td>1,1</td> <td>0,*</td> </tr> <tr> <td></td> <td></td> <td>name String</td> </tr> <tr> <td></td> <td></td> <td>type String</td> </tr> </table>	Resource_Type		Feature_Type	name String	1,1	0,*			name String			type String	
Resource_Type		Feature_Type												
name String	1,1	0,*												
		name String												
		type String												
Typ		Klasse (extensional)												
Ressourcentyp	<table border="1"> <tr> <td>Drucker</td> <td>Drucker</td> </tr> <tr> <td>bezeichnung String</td> <td>bezeichnung String</td> </tr> <tr> <td>seitProMin Integer</td> <td>anzahl Integer</td> </tr> <tr> <td>aufloesung Resolution</td> <td>wartungskosten Float</td> </tr> </table>	Drucker	Drucker	bezeichnung String	bezeichnung String	seitProMin Integer	anzahl Integer	aufloesung Resolution	wartungskosten Float	Fokus auf Mengen				
Drucker	Drucker													
bezeichnung String	bezeichnung String													
seitProMin Integer	anzahl Integer													
aufloesung Resolution	wartungskosten Float													
Initialisierter Typ														
konkreter Ressourcentyp	<table border="1"> <tr> <td>Drucker</td> </tr> <tr> <td>bezeichnung HP-8L</td> </tr> <tr> <td>seitProMin 14</td> </tr> <tr> <td>aufloesung 1200x1200</td> </tr> </table>	Drucker	bezeichnung HP-8L	seitProMin 14	aufloesung 1200x1200									
Drucker														
bezeichnung HP-8L														
seitProMin 14														
aufloesung 1200x1200														
Spezialisierter Typ														
Spezialisierter Ressourcentyp	<table border="1"> <tr> <td>Laserdrucker</td> </tr> <tr> <td>bezeichnung String</td> </tr> <tr> <td>seitProMin Integer</td> </tr> <tr> <td>aufloesung Resolution</td> </tr> <tr> <td>seitProKass Integer</td> </tr> </table>	Laserdrucker	bezeichnung String	seitProMin Integer	aufloesung Resolution	seitProKass Integer								
Laserdrucker														
bezeichnung String														
seitProMin Integer														
aufloesung Resolution														
seitProKass Integer														
"Clone"														
konkreter Ressourcentyp als Erweiterung	<table border="1"> <tr> <td>Drucker</td> </tr> <tr> <td>bezeichnung HP-8LS</td> </tr> <tr> <td>seitProMin 14</td> </tr> <tr> <td>aufloesung 1200x1200</td> </tr> <tr> <td>beidSeitig true</td> </tr> </table>	Drucker	bezeichnung HP-8LS	seitProMin 14	aufloesung 1200x1200	beidSeitig true								
Drucker														
bezeichnung HP-8LS														
seitProMin 14														
aufloesung 1200x1200														
beidSeitig true														
Konkrete Instanz														
konkrete Ressourceninstanz	<table border="1"> <tr> <td>Drucker</td> </tr> <tr> <td>bezeichnung HP-8LS</td> </tr> <tr> <td>seitProMin 14</td> </tr> <tr> <td>aufloesung 1200x1200</td> </tr> <tr> <td>beidSeitig true</td> </tr> <tr> <td>geraeteNr AD-6721</td> </tr> </table>	Drucker	bezeichnung HP-8LS	seitProMin 14	aufloesung 1200x1200	beidSeitig true	geraeteNr AD-6721							
Drucker														
bezeichnung HP-8LS														
seitProMin 14														
aufloesung 1200x1200														
beidSeitig true														
geraeteNr AD-6721														

Abbildung 5: Abstraktionsebenen bei der Modellierung von Ressourcen

Bei näherer Betrachtung der in Abbildung 5 dargestellten Abstraktionsebenen wird zweierlei deutlich. So ist es einerseits leicht vorstellbar noch weitere Ebenen einzuführen. Beispielsweise könnten weitere Druckerklassen wie z.B. „Seitendrucker“ eingeführt werden. Im Hinblick auf die Verwendung entsprechender Konzepte zeigt sich andererseits, dass die Unterscheidung zwischen Spezialisierung und Instanzierung nicht immer intuitiv ist: Ist ein Laserdrucker eine Instanz oder eine Spezialisierung der Klasse „Drucker“? Andererseits legen die Beispiele die Frage nahe, ob überhaupt alle Abstrak-

tionsebenen für die konzeptionelle Modellierung von Geschäftsprozessen benötigt werden. Zunächst kann man tendenziell feststellen, dass die Beschreibung konkreter Instanzen i.d.R. nicht Gegenstand der konzeptionellen Modellierung ist. Vielmehr soll ein konzeptionelles Modell ja bewusst von den Teilen des abgebildeten Realitätsbilds abstrahieren, die absehbaren Änderungen unterliegen. Ähnliches gilt für konkrete Ressourcentypen. Im Einzelfall mag es allerdings wichtig sein, konkrete Ressourcentypen zu erfassen (etwa einen bestimmten Rechnertyp), weil er spezifische Implikationen für die Gestaltung von Geschäftsprozessen mit sich bringt, die nicht vernachlässigt werden dürfen.

Während die Abbildung konkreter Instanzen nicht zu den Anforderungen an eine konzeptionelle Modellierungssprache zählt, sind allerdings Instanzen dennoch in zweifacher Hinsicht zu berücksichtigen. So sollten Sprachkonzepte vorhanden sein, die die Identität von Instanzen auszudrücken erlauben. Wenn beispielsweise zwei Teilprozessen eines Geschäftsprozesstyps ein Sachbearbeiter zugeordnet ist, sollte es möglich sein, auszudrücken, ob es sich dabei jeweils um die gleiche Instanz handeln muss – bzw. nicht handeln darf. Daneben spielen Instanzen für die Vorbereitung von Simulationen eine Rolle. Dabei ist zu berücksichtigen, dass dazu keine konkreten Instanzen verwendet werden. Da betrifft einerseits die Initialisierung (die eben nicht die Abbildung z.B. eines bestimmten Mitarbeiters darstellt), andererseits aber auch die konzeptionelle Ebene. So kann es für eine Simulation wichtig sein, Eigenschaften von Mitarbeitern abzubilden, die bei der Repräsentation tatsächlicher Mitarbeiter keine Rolle spielen bzw. keinen Sinn machen, z.B. durchschnittliche Ausfallzeit pro Jahr, durchschnittliche Kosten pro Stunde etc. Aus diesem Grund sollte eine Sprache zur Modellierung von Geschäftsprozessen ein Konzept enthalten, dass es gestattet prototypische Instanzen für Simulationszwecke zu beschreiben.

Um eine differenzierte Modellierung zu unterstützen, sollte eine Sprache zur Modellierung von Geschäftsprozessen in jedem Fall erlauben, das Abstraktionsniveau der angebotenen Konstrukte explizit zu machen. Die Darstellung von Typen ist unverzichtbar. Im Hinblick auf die Analyse von Prozessen ist ein ergänzendes extensionales Klassen-

konzept wichtig, da so Bewertungen auf der Basis durchschnittlicher bzw. prototypischer oder hypothetischer Ausprägungen möglich werden. Die Identifikation und Unterscheidbarkeit von Instanzen ist ebenfalls eine wichtige Anforderung. Die Einführung von prototypischen Instanzen (oder wie immer ein entsprechendes Konzept im Kontext eines spezifischen Werkzeugs genannt wird) ist lediglich dann wichtig, wenn Simulationen durchgeführt werden sollen.

4.1.2. Flexibilität/Anpassbarkeit

Die Unterscheidung der dargestellten Abstraktionsebenen ist für die Beurteilung von Modellierungswerkzeugen von grundlegender Bedeutung. Sie berührt auch die schwierige Frage nach der Unterscheidung zwischen Sprache und Sprachanwendung bzw. die Frage nach dem Spezialisierungsgrad von Sprachkonzepten. Die Gestaltung dieser Differenzierung hat einen erheblichen Einfluss auf die Flexibilität eines Modellierungswerkzeugs. Um diesen Zusammenhang zu erläutern, betrachten wir beispielhaft die Modellierung von Ressourcen. Wenn wir davon ausgehen, dass die Modellierung von Geschäftsprozessstypen u.a. die Verwendung von Ressourcentypen erfordert, kann diesem Umstand dadurch begegnet werden, dass dem Modellierer eine Menge von Ressourcentypen zur Verfügung gestellt wird. Er wird dadurch von der Last befreit, selbst Ressourcentypen zu spezifizieren. Es werden also Artefakte (Ressourcentypen) bereitgestellt, die eine Anwendung einer Modellierungssprache darstellen. Dieser Ansatz offenbart dann Schwächen, wenn Ressourcentypen benötigt werden, die in der vorgegebenen Menge nicht enthalten sind. Um dem Modellierer eine Chance zu geben, Ressourcentypen auf seinen Bedarf anzupassen, können unterschiedliche Ansätze verwendet werden. Der erste Ansatz beruht auf einer generischen Struktur zur Beschreibung von Ressourcentypen, die Typunterscheidungen lediglich durch die Belegung eines dafür vorgesehenen Attributs erlaubt (siehe Abbildung 6).

Resource	
type:	String
description:	String
category:	Real

Resource	
type:	'Laserdrucker'
description:	'Ein Laserdrucker zeichnet sich durch eine relativ hohe ...'
category:	'Gebrauchsressource'

Abbildung 6: „Typdifferenzierung“ durch Eigenschaften von Instanzen

Die Eigenschaften des Typs werden in einem entsprechenden Attribut in Form eines Textes beschrieben. Während ein solcher Ansatz vordergründig die Beschreibung beliebiger Typen ermöglicht, ist er mit gravierenden Nachteilen verbunden. So handelt es sich hier nicht um Typen im strengen Sinn, sondern lediglich um Instanzen, deren Zustände einen Hinweis auf einen intendierten Typ geben. Damit sind Überprüfungen von Typen und daran anknüpfende Integritätsbedingungen durch das Werkzeug weitgehend ausgeschlossen. Damit zusammenhängend ist eine leistungsfähige maschinelle Auswertung von Typeigenschaften kaum möglich, da die Struktur aller Typen formal gleich ist. Wir können auch sagen: Es wird auf einem niedrigen semantischen Niveau modelliert. Deshalb ist ein solcher Ansatz i.d.R. unbefriedigend. Eine ähnliche Einschränkung gilt für einen weiteren Ansatz, der in der standardisierten Modellierungssprache UML propagiert wird. Sogenannte „*Stereotypes*“ erlauben die Erweiterung der UML um neue Sprachkonzepte (vgl. [OMG99]). Übertragen auf die Spezifikation von Typen würde man etwa den generischen Typ „Ressource“ durch die Zuordnung von Stereotypes kategorisieren. Dadurch wird zwar die maschinelle Identifikation der damit intendierten Typen erleichtert. Das ändert allerdings nichts an dem Umstand, dass die Semantik der so geschaffenen Typen nicht formal beschrieben wird.

Ein weiterer Ansatz basiert auf der Vorgabe generischer Typen, die vom Anwender spezialisiert werden können. Diese Form der Anpassung ist relativ komfortabel. Ihre Flexibilität ist allerdings dadurch eingeschränkt, dass letztlich nur Erweiterungen existierender Typen in Frage kommen. Typen, die nicht sinnvoll als Erweiterungen existierender Typen zu spezifizieren sind, scheiden damit aus. Der dritte Ansatz schließlich bietet das höchste Maß an Flexibilität: Die Bereitstellung einer Sprache zur Spezifikation von Typen. Diese Sprache kann mehr oder weniger speziell sein. So könnte man

beispielsweise eine objektorientierte Modellierungssprache zur Modellierung von Ressourcentypen anbieten. Damit ließe sich ein breites Spektrum von Ressourcentypen beschreiben. Die generischen Sprachkonzepte (etwa elementare Datentypen zur Festlegung der Semantik von Attributen) erlauben allerdings keine komfortable Beschreibung ressourcenspezifischer Eigenschaften. Wenn man etwa davon ausgeht, dass (Gebrauchs-) Ressourcen u.U. durch Nutzungskosten beschrieben werden, wäre ein spezielles, also semantisch reichhaltigeres Sprachkonzept „Kosten“ besser geeignet als etwa der generische Datentyp „Float“.

Schließlich könnte ein Werkzeug ein entsprechendes Attribut leichter als Kosten identifizieren und für dedizierte Analysen verwenden. Der mächtigste Ansatz schließlich besteht darin, dem Anwender die Einführung neuer Sprachkonzepte zu gestatten. Dazu muss ihm letztlich eine Metasprache zur Verfügung gestellt werden. Dieser Ansatz hat allerdings seine Tücken. So dürften die meisten Anwender mit der Spezifikation von Sprachkonzepten überfordert sein. Zudem droht der ursprüngliche Sprachentwurf durch Erweiterung inkonsistent zu werden oder zumindest an Kohärenz einzubüßen.

Konkretisierung der Anforderungen:

Die Ableitung konkreter Anforderungen aus dem dargestellten Zusammenhang gestaltet sich insofern schwierig als die verschiedenen Alternativen spezifische Vor- und Nachteile haben. Ihre Bewertung hängt deshalb von der Varianz möglicher Typen in dem jeweils betrachteten Anwendungsbereich ab. Die folgende Tabelle verdeutlicht diesen Umstand.

	Handhabung von Varianz	Semantisches Niveau (Typsicherheit)	Komfort
flaches Typkonzept	++	---	+
Stereotypes	++	--	+
Mitgelieferte Typen	*	+++	+++
Spezialisierung	*	+++	++
Sprache	+++	++	-
Metasprache	++++	+	--

+	zufriedenstellend
++	gut
+++	sehr gut
*	hängt von Art und Bandbreite ab
-	unbefriedigend
--	problematisch
---	ungenügend

Tabelle 1: Bewertung von Ansätzen zur Erhöhung der Modellierungsflexibilität

Angesichts der ambivalenten Beurteilung einzelner Ansätze ist die Forderung naheliegend, dass eine Modellierungssprache mehrere Alternativen anbieten sollte.

4.1.3. Unterstützung von Sichten

Geschäftsprozesse sind ein zentraler Gegenstand der Betrachtung von Unternehmen. Dabei ist allerdings zu berücksichtigen, dass die Blickwinkel auf Geschäftsprozesse mit dem Betrachter und dem jeweiligen Analysezweck variieren. Diese Varianz betrifft die Detaillierung, die Präzision und den Umfang von Modellen. Die Organisation eines Geschäftsprozessstyps legt einerseits eine detaillierte Darstellung von Kontrollstrukturen und den jeweils benötigten Ressourcen nahe, andererseits können auch Modelle hilfreich sein, die Kompositionshierarchien von Prozessstypen darstellen, also zeigen, wel-

che Prozesstypen ein betrachteter Prozesstyp enthält. Aus der Sicht der Geschäftsleitung ist mitunter eine Black-Box-Darstellung, die durch aggregierte Kosten- und Leistungsgrößen ergänzt wird, hinreichend. Soll ein Geschäftsprozessmodell als Grundlage für die Implementierung dienen, sind ergänzende Angaben – z.B. im Hinblick auf die benötigten Datentypen bzw. Klassen – erforderlich. Im Hinblick auf unternehmensweite Reorganisationsmaßnahmen reicht die isolierte Behandlung einzelner Geschäftsprozessstypen nicht aus. Stattdessen sind auch Beziehungen zwischen Geschäftsprozessstypen zu berücksichtigen. Sie sollten also in einem Werkzeug abzubilden sein. Für all diese Sichten kann der Bedarf an Detaillierung mit der Betrachtungssituation variieren: So gibt es Situationen, in denen ein hohes Maß an Detaillierung erforderlich ist, und andere, in denen es hilfreicher ist, von Details abzusehen.

Der Bedarf an unterschiedlichen Darstellungen markiert insofern eine wichtige Anforderung an Modellierungswerkzeuge, als es sich dabei um verschiedene Sichten auf ein Modell handeln sollte. Wenn für jede Sicht ein eigenständiges Modell zu erstellen wäre, würde dies zu einer erheblichen Redundanz führen – mit wenig erfreulichen Konsequenzen für die Wirtschaftlichkeit der Modellierung und die Integrität der Modelle. Werkzeuge sollten also das Konzept der Sicht auf Geschäftsprozessmodelle beinhalten. Dazu gehören insbesondere folgende Sichten:

- Kontrollstrukturen eines Geschäftsprozessstyps, ggfs. ergänzt um eine ‚light‘-Darstellung
- Kompositionshierarchie eines Prozesstyps
- Beziehungen zwischen Geschäftsprozessstypen
- die in einem Prozesstyp verwendeten Ressourcen
- die in einem Prozesstyp verwendeten Informationen zusammen mit den jeweils genutzten Medien
- die Organisationseinheiten bzw. organisatorischen Rollen, die im Kontext eines Geschäftsprozessstyps zu berücksichtigen sind

4.2. Betriebswirtschaftliche und organisatorische Konzepte

Sprachen zur Geschäftsprozessmodellierung dienen nicht der Darstellung von beliebigen, sondern eben von Prozessen eines Unternehmens oder einer Institution. Daher sollten sie geeignet sein, solche Sachverhalte, die für die betriebswirtschaftliche Analyse eines Geschäftsprozestyps erforderlich sind, darzustellen.

4.2.1. Prozess

„Prozess“ ist das zentrale Konzept der Geschäftsprozessmodellierung. Zugleich ist es ein elementares ontologisches und auch erkenntnistheoretisches Konzept, das kaum weiter aufgelöst werden kann. Deshalb sind Prozessdefinitionen üblicherweise auch selbstbezüglich, wobei häufig auf weitgehend synonyme Begriffe wie „Ablauf“ oder „Vorgang“ zurückgegriffen wird. Für die Modellierung von Geschäftsprozessen ist eine rekursive Begriffsfestlegung durchaus hinreichend: Ein Prozess besteht danach aus einer Menge von (Teil-) Prozessen, für deren Ausführung eine temporale bzw. kausale Ordnung festgelegt ist. Eine entsprechende Dekomposition wird solange durchgeführt bis eine weitere Differenzierung obsolet erscheint.

Ein nicht weiter zerlegter Prozess sollte durch eine Reihe von Eigenschaften beschreibbar sein. So muss es möglich sein, auszudrücken, wann ein Prozess ausgelöst wird und wann er als beendet angesehen werden kann. Allgemein kann man sagen, dass ein Prozess durch ein oder mehrere *Ereignisse* (s.u.) ausgelöst wird. Ein Prozess terminiert, wenn er ein definiertes Ergebnis produziert hat. Dabei kann es sich um mehrere alternative Ergebnisse handeln. Dazu gehört auch das mögliche Scheitern des Prozesses in dem Sinne, dass ein gewünschtes Ergebnis nicht erreicht wird. Ergebnisse sind in einer großen Bandbreite denkbar. Abstrakt betrachtet, kann man das Eintreten solcher Ergebnisse als Ereignis auffassen. Es sollte also möglich sein, einem Prozess sowohl Ereignisse zuzuordnen, die ihn auslösen, als auch solche, die er erzeugen kann. Im Hinblick auf die Durchführung weiterer Analysen oder Simulationen sollte es zudem möglich sein, alternativen Ereignissen, die ein Prozess produzieren kann, *Wahrscheinlichkeiten* oder *Konfidenzmaße* zuzuordnen.

Weitere Angaben, die für die Analyse von Geschäftsprozessen von Interesse sind, umfassen:

- *Zeitliche Dauer*: Hier ist neben einer Zuordnung einer festen Ausführungszeit auch die Angabe eines Intervalls [minimale Dauer ... maximale Dauer] oder einer Durchschnittszeit wünschenswert. Ergänzend dazu kann es hilfreich sein, für ausgewählte Ausführungszeiten Wahrscheinlichkeiten oder Konfidenzmaße angeben zu können.
- *Kosten*: Prinzipiell gilt für die zugewiesenen Kosten das gleiche wie für die Ausführungszeiten. Ergänzend sollte es aber noch möglich sein, weitere Angaben zur Kostenrechnung zu machen, etwa in Form von Hinweisen auf die jeweils beanspruchten Kostenarten (z. B. Sachkosten, Kapitalkosten, Wagniskosten), Kostenstellen (z. B. Verwaltung, Vertrieb) und Kostenträger (z. B. Produkt, Dienstleistung). Dabei ist zu berücksichtigen, dass Kosten, die in Prozessen entstehen, auch indirekt beschrieben werden können, indem Kosten für Nutzung und Verbrauch von Ressourcen (s. 4.2.4) angegeben werden. Eine solche indirekte Beschreibung dürfte aber in vielen Fällen als zu aufwändig angesehen werden.
- *Ausführung*: Um die Ausführung eines Prozesses zu modellieren, sind verschiedene Abstraktions- und Detaillierungsstufen denkbar. So können die innerhalb eines elementaren, nicht weiter dekomponierten Prozesses, auszuführenden Tätigkeiten bzw. Aufgaben benannt und ggfs. natürlichsprachlich beschrieben werden. Ergänzend dazu können Funktionen von Software oder Operationen von Objekten, die in einem assoziierten Objektmodell spezifiziert sind, angegeben werden. Um die zur Steuerung eines Prozesses bzw. eines Workflows zu entwickelnde Software zu spezifizieren, ist es darüber hinaus nötig, geeignete formale Beschreibungsverfahren wie etwa Pseudocode oder Petrinetze zur Verfügung zu haben.
- *Kritische Erfolgsfaktoren*: Falls die Durchführung eines Prozesses eine gewisse Erfahrung erfordert, kann es sinnvoll sein, kritische Erfolgsfaktoren zuordnen zu können. Das können Aspekte der Aufgabenerfüllung sein, die besondere Aufmerksamkeit verdienen oder Hinweise auf einzusetzende Heuristiken, Techniken und Methoden.

Weitere Eigenschaften von Prozessen ergeben sich durch die Zuordnung anderer Konzepte der Organisationsmodellierung, wie Ressourcen, Organisationseinheiten etc. beschrieben.

Nicht zuletzt sollte es möglich sein, Beziehungen zwischen Prozessen auszudrücken, wobei wir an dieser Stelle von solchen Beziehungen absehen, die sich in Kontrollstrukturen ausdrücken:

- *Aggregation*: Eine Aggregationsbeziehung zeigt an, dass ein Prozess aus anderen Prozessen zusammengesetzt ist. Von der Reihenfolge, in der die Teilprozesse ausgeführt werden, wird dabei abstrahiert. Da ein Prozess mehrere Instanzen eines Prozesstyps beinhalten kann, sollte es möglich sein, Kardinalitäten anzugeben.
- *Benutzt*: Eine Benutzt-Beziehung zeigt an, dass ein Prozess einen anderen Prozess benötigt. Auch hier sollte es möglich sein, Kardinalitäten anzugeben.
- *Spezialisierung*: Auch wenn bisher kein überzeugendes formales Konzept für die Semantik der Spezialisierung von Prozesstypen vorliegt, kann es im Hinblick auf die Dokumentationsfunktion von Prozessmodellen hilfreich sein, Spezialisierungsbeziehungen, die aus der Sicht des Modellierers offenkundig sind, als solche ausdrücken zu können. Spezialisierungsbeziehungen dürfen nicht zyklisch sein.
- *Ähnlichkeit*: Falls keine Spezialisierung, wohl aber eine Ähnlichkeit zwischen zwei Prozesstypen vorliegt, kann es zur Unterstützung der Wartung und der Wiederverwendung nützlich sein, Ähnlichkeitsbeziehungen – auch ohne ein formales Konzept von Ähnlichkeit – zwischen zwei Prozesstypen darstellen zu können.

Eigenschaften von Prozesstypen

In 4.1.1 wurden bereits verschiedene Abstraktionsebenen der Geschäftsprozessmodellierung dargestellt. In der Modellierung werden in den meisten Fällen Prozesstypen dargestellt, auch wenn aus Gründen der sprachlichen Vereinfachung von Prozessen die Rede ist. Ein Prozesstyp stellt dabei eine Abstraktion über die Menge seiner möglichen Instanzen dar. Er beschreibt also stellvertretend Eigenschaften, die für seine Instanzen

gelten. Wenn etwa eine Aggregationsbeziehung zwischen zwei Prozesstypen angegeben wird, so handelt es sich tatsächlich um die Beschreibung einer Menge von Beziehungen zwischen Instanzen dieser Prozesstypen. Demgegenüber ist eine Spezialisierungsbeziehung tatsächlich als eine Beziehung zwischen Typen anzusehen.

Bei der Modellierung von Geschäftsprozessen sind aber auch Eigenschaften zu berücksichtigen, die sich nicht auf Instanzen beziehen, sondern Eigenschaften eines Prozess-typs darstellen. Hier ist z. B. an die durchschnittliche Zahl von Instanzen in einem Betrachtungszeitraum zu denken, das Datum, an dem ein Geschäftsprozess eingeführt wurde, oder eine Bewertung seiner Bedeutung für die Wettbewerbsfähigkeit eines Unternehmens.

Identität von Instanzen

Falls innerhalb eines Prozesstyps ein (Teil-) Prozesstyp mehrmals vorkommt, stellt sich die Frage, ob es sich dabei jeweils um eine andere Instanz handeln muss oder ob es ggfs. immer die gleiche Instanz sein muss. Es sollte möglich sein, solche Unterscheidungen auszudrücken.

4.2.2. Ereignis

Ähnlich wie ‚Prozess‘ stellt auch ‚Ereignis‘ ein elementares ontologisches Konzept dar, das kaum auf primitivere Konzepte zurückgeführt werden kann. Ein Ereignis steht für das Eintreten eines Zustands, das für die Durchführung eines Prozesses von Bedeutung ist. Dabei kann es sich u.a. um die Änderung des Zustands von Objekten (z. B.: der Lagerbestand eines Produkts ist kleiner als der Mindestbestand) handeln, das Entstehen von Objekten oder das Löschen von Objekten (z. B.: ein Kunde lässt seinen Auftrag stornieren). Auch wenn sich temporale Ereignisse als Änderungen des Zustands einer Referenzuhr auffassen lassen, macht es Sinn, sie getrennt zu behandeln. Man kann zwei grundsätzliche Arten temporaler Ereignisse unterscheiden: das Eintreten eines bestimmten Zeitpunkts und das Erreichen einer bestimmten zeitlichen Dauer. Tabelle 2 gibt einen Überblick über Arten von Ereignissen. Temporale Ereignisse können auch durch Rückgriff auf nicht-temporale Ereignisse spezifiziert werden. Dabei kann sowohl das

Eintreten eines Zustands oder einer Zustandsänderung als auch das Ausbleiben derselben zu einem entsprechenden Ereignis führen.

Kategorie	Art	Beispiel
Entstehen eines Zustands	Objekt entstanden	neuer Auftrag eingegangen
	Objekt mit bestimmten Zustand entstanden	neuer Auftrag mit einer Auftragssumme von > Euro 10.000.- eingegangen
	Objekt gelöscht	Auftrag storniert
	Objekt mit bestimmten Zustand gelöscht	Auftrag mit einer Auftragssumme von > Euro 10.000.- gelöscht
Zustandsänderung	Zustand eines Objekts ändert sich	Preis eines Produkts geändert
	Zustand eines Objekts ändert sich in bestimmter Weise	Preis eines Produkts < Euro 20.-
	Zustand einer Klasse von Objekten ändert sich	durchschnittliche Auftragssumme > Euro 1.000
	Ausmaß der Zustandsänderung	Preis um mehr als 25% des Höchstpreises gesunken
Anzahl von Einzelereignissen	Anzahl neuer Zustände	mehr als 10 neue Aufträge eingegangen
	Anzahl von Zustandsänderungen	mehr als 100 Aufträge nachträglich modifiziert
temporal	bis zum Zeitpunkt t	Zahlung bis zum Datum d
	bis zu Zeitpunkt, iterierend	Kassenüberprüfung jeden Freitag, 15:00
	relativer Zeitpunkt	an jedem ersten Donnerstag eines Monats
	Zeitdauer	Zahlung innerhalb von zwei Wochen

Tabelle 2: Arten von Ereignissen

Ereignisse können konjunktiv und disjunktiv verknüpft werden. Ereignisse werden i. d. R. als Typen modelliert. Es kann allerdings sein, dass innerhalb eines Prozesstyps ein Ereignis(-typ) mehrmals verwendet wird. In diesen Fällen sollte es möglich sein, die Identität von Instanzen auszudrücken, um ggfs. spezifizieren zu können, dass es sich um gleiche oder unterschiedliche Instanzen handelt.

Es sollte möglich sein, Beziehungen zwischen Ereignistypen zu modellieren. Ähnlich wie bei Prozesstypen sollten Spezialisierungs- und Ähnlichkeitsbeziehungen auszudrücken sein. Im Unterschied zu Prozesstypen scheint es für die Spezialisierung von Ereignistypen sinnvoll, das Kriterium der Ersetzbarkeit anzuwenden: Überall dort, wo eine Instanz eines Ereignistyps einen bestimmten Effekt hat, sollte das Ersetzen dieser Instanz durch eine Instanz eines spezialisierten Typs genau den gleichen Effekt produzieren. Spezialisierung in diesem Sinne könnte also einer logischen Konjunktion der Aussage, die den Supertyp beschreibt und der Aussage, die die zusätzlichen Eigenschaften des Untertyps beschreibt, entsprechen. Es ist allerdings zu prüfen, ob damit dem Kriterium der Ersetzbarkeit genügt wird. Beispiel: „Eine Reklamation ist eingegangen“ (Obertyp); „Eine Reklamation ist eingegangen UND der Kunde fordert die Rückerstattung des Kaufpreises.“ Es sollte möglich sein, Ereignissen eine Wahrscheinlichkeit bzw. ein Konfidenzmaß zuzuordnen.

Prozess und Ereignis sind die zentralen Konzepte der Prozessmodellierung. Im Hinblick auf die einfache und komfortable Anwendbarkeit einer Modellierungssprache ist damit allerdings der Nachteil verbunden, dass die Unterscheidung von Ereignissen, Prozessen und Zuständen nicht trivial ist und mit dem Blickwinkel des Betrachters variieren kann. So ist nach Lorenz ein Ereignis, das „in seinem Verlauf betrachtet wird“, ein Vorgang (synonym mit Prozess) [Lore80]. Die Phasen eines Prozesses wiederum können „... unter Bezug auf einen Gegenstand als Beschreibungen eines Zustands wiedergegeben werden.“ ([Lore96], S. 564)

Wie sollte eine Modellierungssprache die Spezifikation von Ereignistypen unterstützen? Da es sich um ein zentrales Konzept handelt, sollte die Modellierungssprache zur Dar-

stellung von Ereignissen geeignete Symbole anbieten. Neben einem eindeutigen Bezeichner für einen Ereignistyp sollten in jedem Fall natürlichsprachliche Annotationen möglich sein. Eine bessere Voraussetzung für anschließende formale Analysen, etwa zur Unterstützung der Software-Generierung, bieten semi-formale Darstellungen. Dazu ist eine Sprache mit formaler Syntax nötig, wobei die Semantik nicht vollständig spezifiziert ist. Ein Beispiel dafür wären arithmetische Ausdrücke, deren Objekte nicht formal definiert sind („Auftragssumme > durchschnittliche Auftragssumme“). Die besten Voraussetzungen für weitere Auswertungen eines Prozessmodells bieten formalsprachliche Annotationen, deren Semantik formal spezifiziert ist. Das könnte z. B. ein boolescher Ausdruck sein, in dem Objekte, deren Klassen in einem assoziierten Klassendiagramm spezifiziert sind, und deren Dienste verwendet werden: „Auftrag.summe > AuftragClass.durchschnittssumme“. Dabei muss sichergestellt sein, dass die Identität eines Objekts eindeutig nicht mehrdeutig ist.

4.2.3. Ziele von Geschäftsprozessen

Geschäftsprozesse sind kein Selbstzweck. Sie dienen vielmehr der Erreichung von Zielen. Die Beschreibung dieser Ziele ist für die Modellierung von Geschäftsprozessen aus einer Reihe von Gründen erforderlich. So unterstützt die explizite Darstellung von Zielen die *Dokumentationsfunktion* von Geschäftsprozessmodellen und trägt damit zum Verständnis eines Geschäftsprozesstyps bei. Die *(Re-) Organisation von Geschäftsprozessen*, ein wesentlicher Anlass für die Modellierung, ist nur dann sinnvoll möglich, wenn entsprechende Ziele für die Gestaltung vorgegeben sind. Damit zusammenhängend ist die Betrachtung von Zielen auch erforderlich, um *betriebswirtschaftliche Analysen* durchführen zu können. Das gilt sowohl für die Analyse der internen Organisation eines Geschäftsprozesstyps als auch für die Untersuchung der Frage, ob und in welcher Weise ein Geschäftsprozesstyp zur Erreichung übergeordneter Unternehmensziele beiträgt.

Während die Dokumentationsfunktion auch durch natürlichsprachliche Beschreibungen von Zielen erfüllt werden kann, machen es formale und damit ggfs. automatisierbare Analysen zur Unterstützung der Bewertung und Gestaltung von Geschäftsprozessen

nötig, die Bedeutung von Zielen bzw. von Zieltypen formal zu differenzieren. Hier ist z.B. an die Ermittlung von Schwachstellen zu denken. Für das Ziel „Minimierung von Durchlaufzeiten“ würde etwa ein Flaschenhals eine solche Schwachstelle darstellen. Bei hinreichender Beschreibung der relevanten Einflussfaktoren sind auch Optimierungen denkbar, etwa die Minimierung der Kosten, die mit der Durchführung einer Prozessinstanz entstehen. Angesichts der großen Bandbreite möglicher Geschäftsprozessstypen und der mit ihnen verbundenen konkreten Ziele, ist es allerdings ex ante kaum möglich, alle denkbaren Ziele in differenzierter Weise zu definieren. Das liegt nicht zuletzt daran, dass sich Ziele i.d.R. an bestimmten Eigenschaften der in einem Prozess zu erstellenden Produkte und Leistungen festmachen. Solange diese Eigenschaften selbst nicht formalisiert sind, lassen sich also auch die Ziele nicht vollständig formalisieren.

In der Betriebswirtschaftslehre gibt es zahlreiche Ansätze, Kategorien von Zielen zu bilden. Dazu gehört auch die Abgrenzung von Zielen gegenüber Randbedingungen, wie sie vor allem im Operations Research üblich ist. Danach wäre beispielsweise Kostenminimierung ein Ziel, die Einhaltung einer maximalen Ausführungszeit eine Randbedingung. Wenn man die gängige Bedeutung des Begriffs Ziel – das, was erreicht werden soll - zum Maßstab nimmt, ist eine solche Unterscheidung nicht überzeugend: Danach handelt es sich offenbar in beiden Fällen des Beispiels um ein Ziel. Die Unterscheidung ist denn auch eher durch die Struktur formaler Entscheidungsmodelle begründet, die eine Differenzierung von zu optimierenden Größen (Zielen) und zulässigen Wertebereichen für andere erfolgskritische Größen (Randbedingungen bzw. Restriktionen) vorsehen. Im Bereich des Systementwurfs ist es üblich, Restriktionen, denen ein System oder Teile desselben zu genügen haben, in Form von Integritätsbedingungen (Constraints) explizit anzugeben. Vor diesem Hintergrund scheint es für die Modellierung von Geschäftsprozessen sinnvoll, zwischen Zielen als zu optimierenden Größen und Integritätsbedingungen, gegen die bei der Zielverfolgung nicht verstoßen werden darf, zu unterscheiden.

Weitere Unterscheidungen machen sich an grundsätzlichen *Zieldimensionen* fest. Hier ist an *quantitative*, *temporale* und *qualitative* Ziele zu denken. Ein quantitatives Ziel ist

auf die Optimierung quantifizierbarer Größen gerichtet, wie z.B. Ausbringungsmengen, eingehenden Ressourcen, Kosten, Deckungsbeiträge etc. Ein temporales Ziel bezieht sich auf die Optimierung von Zeitintervallen, also etwa die Minimierung der Durchlaufzeit eines Prozesses. Qualitätsziele sind auf die Optimierung von Qualität gerichtet, die bekanntlich schwerer zu fassen ist. Sie können allerdings mitunter auf quantitative Ziele abgebildet werden, beispielsweise in folgender Form: „Minimierung des Ausschusses“, „Minimierung von Reklamationen“.

Neben diesen generischen Zieldimensionen gibt es eine Reihe zumeist quantitativer Ziele, die sich an gängigen betriebswirtschaftlichen Konzepten – vor allem aus dem Rechnungswesen – festmachen. Hier ist etwa an die Optimierung von Umsätzen, Kosten, Deckungsbeiträgen oder Gewinnen zu denken. Dabei ist allerdings zu berücksichtigen, dass die Komplexität und Kontingenz (z.B. der Nachfrage) der Einflussfaktoren (dabei ist auch an Interdependenzen zwischen Geschäftsprozessen zu denken) eine Optimierung häufig nicht möglich machen wird – allenfalls um den Preis drastischer Vereinfachungen wie sie von den Modellen der betriebswirtschaftlichen Produktionstheorie hinreichend bekannt sind. Dessen ungeachtet kann eine formale Kennzeichnung solcher Ziele die formale Analyse von Geschäftsprozessen unterstützen, etwa dadurch, dass Prozesseigenschaften ermittelt werden können, die die Erreichung eines vorgegebenen Ziels potentiell behindern. So könnten z.B. im Hinblick auf das Ziel Kostenminimierung solche Ressourcen ermittelt werden, die durch andere, kostengünstiger substituierbar sind – vorausgesetzt, entsprechende Beziehungen zwischen Ressourcen wurden modelliert. Falls eine Optimierung i.e.S. nicht möglich ist, ist ein entsprechendes Ziel nicht obsolet. Vielmehr kann man in einem solchen Fall „Optimierung“ auch als „Verbesserung“ interpretieren.

Die skizzierten Dimensionen bzw. Kategorien von Zielen lassen sich auch als Zieltypen beschreiben. Das in Abbildung 7 dargestellte semantische Netz zeigt Generalisierungs- bzw. Spezialisierungsbeziehungen zwischen Typen von Zielen, die in einer Sprache zur Geschäftsprozessmodellierung abgebildet werden könnten.

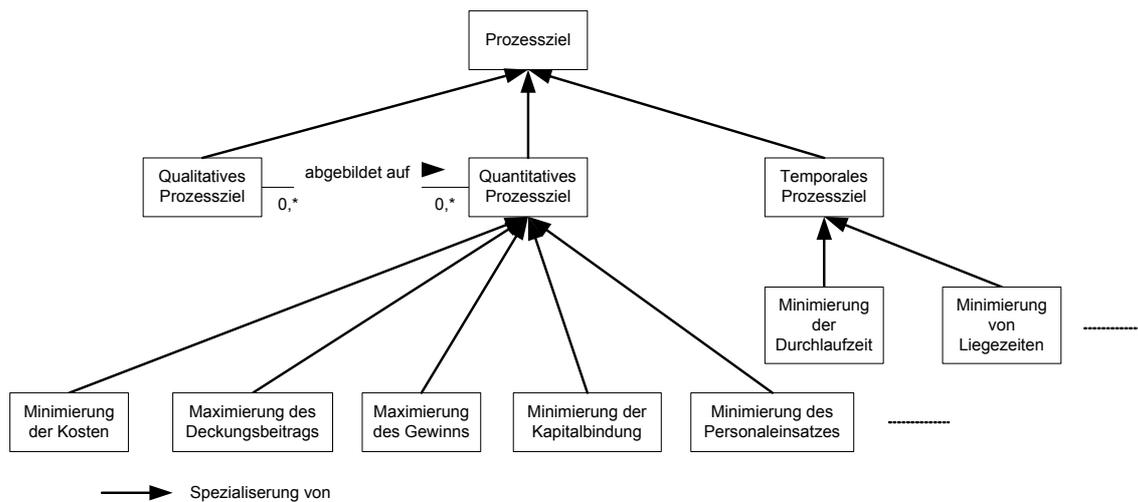


Abbildung 7: Mögliche Typen von Zielen

Die Zieltypen in Abbildung 7 geben zwar einen Hinweis auf eine mögliche Typisierung von Zielen, bei näherer Betrachtung wird allerdings deutlich, dass zunächst eine Reihe von Fragen zu klären ist. Sie betreffen das mit einem Ziel verbundene Abstraktionsniveau, die Formalisierbarkeit von Zielen und die bekannte Frage danach, ob ein darzustellender Sachverhalt als Sprachkonzept oder durch die Anwendung einer Sprache abgebildet werden soll. Zwischen Zielen kann es Beziehungen wie Unabhängigkeit, Konkurrenz, Komplementarität und Widersprüchlichkeit geben. Außerdem kann ein Ziel ein Oberziel von anderen Zielen sein, was bedeutet, dass die Erfüllung von Unterzielen zur Erreichung ihres Oberziels beiträgt.

Abstraktionsniveau

Die für die konzeptionelle Modellierung von Geschäftsprozessen gebräuchliche Abstraktionsebene legt es nahe, Ziele für einen Geschäftsprozesstyp zu formulieren. Dabei ist zwischen solchen Zielen zu unterscheiden, die dem Geschäftsprozesstyp stellvertretend für seine potentiellen Instanzen zugeordnet werden, und solchen, die lediglich auf der Typebene Sinn machen. Zur ersten Kategorie gehören Ziele wie „Minimierung der Durchlaufzeit“ oder „Minimierung der Kosten“, der zweiten Kategorie sind Ziele zuzu-

ordnen wie „Maximierung der Zahl von Prozessinstanzen in einem vorgegebenen Zeitraum“ oder „Minimierung der Zahl fehlerhafter Prozessinstanzen in einem vorgegebenen Zeitraum“. Daneben gibt es Ziele, die sich ohne nähere Betrachtung keiner dieser beiden Kategorien eindeutig zuordnen lassen, wie etwa „Minimierung der durchschnittlichen Durchlaufzeit aller Prozessinstanzen in einem Zeitraum“. Diese Ziel kann eine Minimierung der Durchlaufzeiten in jedem Einzelfall implizieren, muss es aber nicht: Wenn z.B. mehrere Prozessinstanzen parallel – aber nicht unabhängig voneinander – ausgeführt werden, kann es im Hinblick auf die Minimierung der durchschnittlichen Laufzeit angeraten sein, bei einzelnen Instanzen keine Minimierung anzustreben. Gegenüber der Festlegung von Zielen für einzelne Geschäftsprozessstypen lässt sich ein höheres Abstraktionsniveau dadurch erreichen, dass Ziele einer Menge von Geschäftsprozessstypen zugeordnet werden, also bspw.: „Für alle Geschäftsprozessstypen, an denen der Vertrieb beteiligt ist, ist die Durchlaufzeit zu minimieren.“ Dabei stellt sich die an dieser Stelle nicht weiter zu untersuchende Frage, ob Ziele eines Geschäftsprozessstyps an spezialisierte Prozessstypen vererbt werden.

In allen bisher diskutierten Fällen werden Ziele auf der Typebene formuliert und betreffen dann u.U. alle Instanzen der jeweiligen Typen. Es ist aber auch denkbar, dass Ziele nur ausgewählte Instanzen betreffen, z.B. „Minimierung der Durchlaufzeit bei Premiunkunden“. Entsprechende Ziele werden also erst dann wirksam, wenn die Eigenschaften einer konkreten Instanz einem bestimmten Muster entsprechen. Dazu gehört auch der Fall, dass Ziele lediglich für solche Instanzen verfolgt werden, die von einem verantwortlichen Mitarbeiter ausgewählt wurden. Für die Modellierung von Prozessstypen haben solche auf ausgewählte Instanzen gerichtete Ziele insofern Bedeutung als ihre Umsetzung z.B. Möglichkeiten zur dynamischen Allokation von Ressourcen erfordert.

Für alle genannten Fälle ist eine weitere Schwierigkeit zu berücksichtigen, die mit dem je gewählten Abstraktionsniveau zusammenhängt. Um die langfristige Verwendung von Prozessmodellen zu fördern und um ggfs. eine flexible technische Unterstützung von Prozessen zu ermöglichen, ist es i.d.R. angeraten, von unterschiedlichen Ausprägungen der Instanzen von Ressourcen zu abstrahieren. Wenn etwa in einem Geschäftsprozess

ein Rechner benötigt wird und Mobilität kein zwingendes Kriterium ist, kann davon abstrahiert werden, ob es sich um ein Notebook oder um ein Desktop handelt. Vor allem aber ist davon zu abstrahieren, welches konkrete Rechnermodell eingesetzt wird. Für die Rechenzeiten oder die damit verbundenen Kosten ist es aber u.U. wesentlich, welches Modell verwendet wird. Es ist dabei denkbar, dass die Unterschiede zwischen möglichen Instanzen – man denke an Sprünge in der technologischen Entwicklung – so groß sind, dass sie für die Optimierung eines Geschäftsprozesses von Bedeutung sind. In solchen Fällen ist eine Optimierung auf Typebene nicht möglich. In solchen Fällen kann allenfalls eine Simulation (s. Abschnitt 3.3.2) weiter helfen.

Auch bei der Betrachtung von Beziehungen zwischen Zielen ist das Abstraktionsniveau bzw. der jeweilige Kontext zu berücksichtigen. So können die Ziele „Minimierung von Durchlaufzeiten“ und „Minimierung der Kosten“ in einem Kontext unabhängig sein, in einem anderen konkurrierend.

Die vorausgegangen Überlegungen führen uns zu einem weiteren Problem, das zum Teile für Ziele spezifisch zu sein scheint, zum Teil an die Ausführungen in 4.1.1 anknüpft: die Unterscheidung von Zieltypen und Zielinstanzen. Zunächst kann man sagen, dass eine Instanz aus einem Typ erzeugt wird, also in jedem Fall eine eigene Identität hat. Während dieser Umstand etwa für die Unterscheidung der Klasse ‚Person‘ und einer ihrer Instanzen leicht nachvollziehbar ist, ist es nicht offensichtlich, wie sich etwa das Ziel „Minimierung der Durchlaufzeit“ in Typ und Instanz differenzieren lässt. Bei intensionaler Definition legt ein Typ die zulässigen Zustände seiner möglichen Instanzen fest. Eine Instanz unterscheidet sich also von ihrem Typ dadurch, dass sie einen spezifischen Zustand hat. Auf diese Weise ergibt sich die Möglichkeit, eine Zielinstanz von einem Zieltyp zu unterscheiden. So könnte man einer Zielinstanz etwa einen Zielerreichungsgrad zuordnen, der sich im Zeitverlauf ändern könnte. Es ist allerdings nicht immer offenkundig, wann man einen Zieltyp verwenden sollte, wann eine Instanz. So können wir den Zieltyp „Minimierung der Durchlaufzeit“ nicht nur einem Prozesstyp zuordnen, sondern mehreren. Handelt es sich dabei jeweils um den gleichen Zieltyp, um unterschiedliche Typen oder um verschiedene Instanzen eines Typs? Bei einer restrikti-

ven Modellierung von Zieltypen wäre es denkbar, den genannten Zieltyp in verschiedene Zieltypen, die jeweils nur für eine eingeschränkte Menge von Prozesstypen gelten würden, zu differenzieren. Beispielsweise könnte eine Assoziation eines speziellen Zieltyps mit dem Ressourcentyp „Transportmittel“ verpflichtend sein, für andere nicht. Davon abgesehen bleibt die Frage, ob einem Prozesstyp eine Zielinstanz oder ein Zieltyp zugeordnet werden soll. Ist das Ziel, das einer konkreten Prozessinstanz zugeordnet wird ein anderes als das, das einer anderen Instanz bzw. dem zugehörigen Prozesstyp zugeordnet wird? Wenn wir davon ausgehen, dass ein Ziel einen Zustand haben kann, z.B. in Form eines Zielerreichungsgrads, dann würde man einer Prozessinstanz Zielinstanzen zuordnen. Ein Prozesstyp wäre dann mit einem Zieltyp assoziiert, der aber prinzipiell auch einen Zustand haben kann, etwa in Form eines Zielgewichts. Damit ergäben sich die in oben bereits ausführlich behandelten Probleme, da man üblicherweise nur zwei Abstraktionsebenen zur Verfügung hat. Das Problem wird allerdings dadurch entkräftet, dass wir bei der Modellierung i.d.R. keine Instanzen beschreiben, uns also auf Ziele für Prozesstypen beschränken können. Allenfalls zur Vorbereitung von Simulationen wären entsprechende Konzepte nötig. Die Frage kann allerdings auch aus einem anderen Blickwinkel untersucht werden, der mit der Differenzierung von Sprachkonzept und Sprachanwendung zu tun hat (s.u.): Wenn die Modellierungssprache einen allgemeinen Zieltyp bereitstellt, würde für einen bestimmten Prozesstyp daraus eine Instanz zu bilden sein, für die ein Name, eine Beschreibung und ggfs. weitere Eigenschaften zu benennen wären. Das einem Prozesstyp zugeordnete Ziel wäre also formal eine Instanz, konzeptionell aber ein Typ.

Formalisierbarkeit

Es gibt eine Reihe von Zielen, die sich konzeptionell sehr präzise beschreiben lassen. Dazu gehören zeitbezogene Ziele wie die Minimierung von Durchlaufzeiten oder mengenbezogene Ziele wie die Minimierung von Kosten. Dessen ungeachtet kann eine Formalisierung entsprechender Ziele daran scheitern, dass die Erfassung der Zielgrößen im Einzelfall zu aufwändig wäre. So erfordert etwa die Konzeptualisierung der variablen Kosten eines Geschäftsprozesses die Erfassung aller durch den Prozess verursachten Kosten. Sieht man von der Schwierigkeit ab, die damit verbunden sein kann, variable Kosten und Fixkosten im Einzelfall sauber zu trennen – bspw. bei der Beanspruchung von Personal oder Rechnern, bleibt immer noch ein erheblicher Modellierungsaufwand, um alle Kostenarten zu beschreiben, aus denen sich die gesamten Prozesskosten zusammensetzen. Ähnliches gilt für den Zusammenhang zwischen Beschäftigungsänderungen und den daraus resultierenden Kosten. Wir können zusammenfassen, dass die Formalisierung von Zielen daran scheitern kann, dass der Aufwand für eine präzise Konzeptualisierung zu groß ist.

Ein noch gravierenderes Hindernis der Formalisierung liegt dann vor, wenn die Konzeptualisierung, also die formale Rekonstruktion der Semantik eines Ziels mit Schwierigkeiten verbunden ist. Dies dürfte häufig für Qualitätsziele gelten oder allgemein für solche Ziele, die auf Begriffen mit einem deutlich intensionalen⁶ Sinngehalt beruhen. Auch wenn im Einzelfall nicht bewiesen werden kann, dass ein Begriff nicht zu formalisieren ist, zeigt die Erfahrung doch, dass es Begriffe gibt, die für einen menschlichen Betrachter eine gehaltvolle Bedeutung besitzen, sich jedoch gegen eine befriedigende Formalisierung sperren. Beispiele dafür sind Ziele wie „ausgeprägte Kundenorientierung“ oder „hohe Qualität der erbrachten Leistungen“. Ähnliche Schwierigkeiten kön-

⁶ intensional ist hier nicht im Sinn der intentionalen Spezifikation von Klassen oder Typen zu verstehen, sondern im Sinn einer hermeneutischen Sichtweise, wonach die Bedeutung mancher Begriffe nicht authentisch durch eine extensionale Beschreibung aller sinngebenden Eigenschaften zu beschreiben ist. Vielmehr seien dazu auch Assoziationen zu Wahrnehmungen und Erlebnissen erforderlich. Von Wright spricht in diesem Zusammenhang von „Einfühlung oder innerem Nachvollzug der geistigen Atmosphäre“ ([Wri74], S. 20).

nen sich bei der Beschreibung von Restriktionen ergeben, die bei der Zielerreichung einzuhalten sind: „Humane Arbeitsbedingungen müssen gewährleistet sein.“ Bei solchen Begriffen ist im Einzelfall zu entscheiden, ob eine Rekonstruktion mit Hilfe einer begrenzten Zahl von Indikatoren zu brauchbaren Ersatzbegriffen führt oder ob damit unzumutbare Vereinfachungen verbunden sind.

Sprachbestandteil oder Sprachanwendung

Während es unstrittig ist, dass Geschäftsprozessmodelle die Beschreibung von Zielen erfordern, ist es nicht offenkundig, ob Ziele als Konzepte der Modellierungssprache vorgegeben werden sollten oder aber erst mit Hilfe der Modellierungssprache beschrieben werden sollten. Auf den ersten Blick mag es naheliegend erscheinen, Ziele mit Hilfe von Klassen zu beschreiben. Eine entsprechende Klasse wäre durch eine Reihe von Attributen wie z.B. ‚Name‘ oder ‚Beschreibung‘ gekennzeichnet. Ein bestimmtes Ziel würde dann durch eine Instanz dieser Klasse repräsentiert werden. Alternativ dazu könnte für jedes Ziel eine eigene Klasse eingeführt werden. Während ein solcher Ansatz ein hohes Maß an Flexibilität aufweist – irgendwie kann man so alle Ziele beschreiben, hat er gegenüber der Darstellung von Zielen mittels Sprachkonzepten zwei Nachteile. Der erste Nachteil ist darin zu sehen, dass der Sprachanwender gezwungen ist, zentrale Begriffe selbst zu spezifizieren. Er betrifft also den Aufwand bzw. den Komfort der Sprachnutzung. Der zweite Nachteil ist darin zu sehen, dass in der Syntax und Semantik der Sprache die Bedeutung der sprachlich gebildeten Konzepte nicht angemessen berücksichtigt werden. Auf diese Weise können Modelle erstellt werden, die nach Maßgabe der verwendeten Modellierungssprache syntaktisch korrekt sind und auch nicht gegen semantische Integritätsbedingungen verstoßen, die aber offenkundig gegen die spezifischen semantischen Integritätsbedingungen verstoßen, die mit der Beschreibung von Zielen verbunden ist. Um einige Beispiele zu geben: Eine Klasse kann mit beliebigen anderen Klassen – auch mit sich selbst – assoziiert sein. Ein Ziel kann aber nicht gleichzeitig Unterziel und Oberziel eines anderen Ziels sein. Ein Ziel, das mit einem anderen Ziel im Widerspruch steht, kann nicht gleichzeitig eine komplementäre Beziehung zu diesem Ziel haben.

Konkretisierung der Anforderungen:

In jedem Fall sollte eine Sprache zur Modellierung von Geschäftsprozessen Konzepte beinhalten, die eine explizite Darstellung von Zielen ermöglichen. Um der Dokumentationsfunktion zu genügen, reicht es hin, wenn dazu neben der Bezeichnung eines Ziels eine Beschreibung hinterlegt werden kann. Darüber hinaus sollten Beziehungen zwischen Zielen angegeben werden können. Während dies auch der Dokumentation dient, unterstützt die Bereitstellung spezieller Beziehungstypen – wie etwas „ist Oberziel von“ oder „konkurriert mit“ – formale Analysen eines Modells. Falls eine formale Analyse von Geschäftsprozessmodellen als wichtig angesehen wird und der zusätzliche Modellierungsaufwand – auch für die Beschreibung von Ressourcen – akzeptabel ist, sind Zieltypen als Bestandteil der Modellierungssprache hilfreich. Dabei sollte es sich um eine Liste handeln, die wichtige Zieltypen beinhaltet, aber prinzipiell erweiterungsfähig ist. Beispiele für vorgegebene Zieltypen sind:

- Minimierung von Fixkosten
- Minimierung der (variablen) Prozesskosten
- Verbesserung der Kundenorientierung
- Minimierung der Kapitalbindung
- Verbesserung der Produktivität
- Verbesserung der Wirtschaftlichkeit
- Minimierung der Durchlaufzeiten
- Steigerung des Output
- Steigerung der Qualität des Ergebnisses

Integritätsbedingungen, die Beziehungen zwischen Zieltypen betreffen, sollten Sprachbestandteil sein, also z.B., dass die Beziehung „ist Oberziel von“ niemals zyklisch sein darf.

Alternativ ist ein Kompromiss zwischen beiden Ansätzen denkbar: Es wird eine kleine Auswahl von Zieltypen vorgegeben, etwa temporal und quantitativ, die dann durch die Initialisierung der jeweiligen Attribute angepasst werden können.

4.2.4. Ressourcen

Unter Ressourcen sind alle Objekte zu verstehen, die benötigt werden, um einen Prozess auszuführen – und die knapp sind, deren Verwendung also mit Kosten verbunden ist.

In der Betriebswirtschaftslehre wird der Begriff „Ressource“ nur selten verwendet. Stattdessen ist es üblich – vor allem in der betriebswirtschaftlichen Produktionstheorie – von *Produktionsfaktoren* zu sprechen. Auch wenn – wie sich noch zeigen wird – eine solche Terminologie für Geschäftsprozesse nicht überzeugend ist, betrachten wir dennoch zunächst gängige Kategorisierungen von Produktionsfaktoren, um Hinweise für eine Kategorisierung von Ressourcen zu erhalten. I.d.R. wird zwischen Werkstoffen, Betriebsmitteln und menschlicher Arbeit unterschieden (z. B. [BILü02], S. 148 ff.). Werkstoffe gehen in die zu erstellenden Produkte ein. Sie umfassen Rohstoffe, Hilfsstoffe und Betriebsstoffe. Anders als im umgangssprachlichen Gebrauch ist der Begriff „Rohstoff“ kontextabhängig: Zu den Rohstoffen eines Produktionsprozesses gehören alle Güter, die von außen in den Prozess eingehen. Es kann sich dabei also durchaus um Konstruktionsteile handeln, die zugeliefert werden. Hilfsstoffe sind Werkstoffe, die in ein Produkt eingehen, aber nicht als Bestandteil bzw. Komponenten des Produkts angesehen werden, also z.B. Farben, Klebstoffe oder Nägel. Betriebsstoffe dienen dazu, Betriebsmittel zu nutzen. Sie umfassen Kraftstoffe, elektrische Energie, Schmiermittel etc. Zu den Betriebsmitteln gehören Grundstücke, Gebäude, Produktionsanlagen, Maschinen, Werkzeuge etc. Mitunter werden aber auch immaterielle Güter wie „Rechte und das Wissen des Betriebes“ ([BILü02], S. 149) dazu gezählt. Menschliche Arbeit wird unterschieden in ausführende Arbeit und dispositive Arbeit, also solche, die auf die Gestaltung von Produktionsprozessen und die Führung von Mitarbeitern gerichtet sind. Im Hinblick auf die Wirtschaftlichkeit von Produktionsprozessen ist der Verbrauch an Produktionsfaktoren ein wesentliches Kriterium. Bei Betriebsmitteln wie etwa Maschinen wird der Verbrauch alternativ in Abhängigkeit vom Niveau des Outputs oder von der technischen Leistung gemessen.

Produktionstheorie		Geschäftsprozessmodellierung
Produktionsfaktor	Beispiele	Beispiele
<i>Werkstoffe</i>		
Rohstoffe	Konstruktionsteile, Komponenten	Papier (?)
Hilfsstoffe	Farben, Klebstoffe ...	Tinte, Farbband, Diskette ...
Betriebsstoffe	Kraftstoffe, elekt. Energie ...	Elekt. Energie
<i>Betriebsmittel</i>	Grundstücke, Gebäude, Maschinen, Werkzeuge ...	Gebäude, Rechner, Software, Kopierer, Drucker ...
	Rechte, Wissen, Information	Rechte, Wissen, Information
<i>Menschliche Arbeit</i>		
Ausführende Arbeit	Lagerarbeiter, Fließbandarbeiter	Sachbearbeiter
Dispositive Arbeit	Produktionsleiter	Gruppenleiter, Prozessverantwortlicher

Tabelle 3: Projektion von Produktionsfaktoren auf Geschäftsprozesse

Tab. 3 zeigt die Projektion von Produktionsfaktoren auf Geschäftsprozesse. Dabei wird deutlich, dass Produktionsfaktoren zur Differenzierung von Ressourcen in Geschäftsprozessen nur bedingt geeignet sind. Das hat im wesentlichen zwei Gründe. So schließt die Modellierung von Geschäftsprozessen Produktionsprozesse zwar nicht grundsätzlich aus, fokussiert aber doch eher auf Prozesse im Büro- und Verwaltungsbereich, in dem i. d. R. andere Ressourcen zu berücksichtigen sind. Im Unterschied zu Produktionsprozessen geht es dabei nicht um die Erstellung eines physischen Produkts. Vielmehr spielt die Nutzung und Erzeugung von Information bzw. Wissen eine zentrale Rolle. Aus diesem Grund ist es auch fraglich, ob in einem Geschäftsprozess, der auf die Erstellung eines Kaufvertrags zielt, das Papier, auf dem der Vertrag dokumentiert ist, Bestandteil des zu erstellenden Ergebnisses ist oder ob es nicht eher eine Funktion hat, die der einer Verpackung entspricht. Auch ist die Unterscheidung zwischen Werkstoffen und Betriebsmittel im Fall von Information, die in einem Geschäftsprozess benötigt wird, nicht überzeugend. Während Produktionsfaktoren in einem Produktionsprozess verbraucht

oder durch Gebrauch abgenutzt und damit wirtschaftlich abgeschrieben werden, ist ein entsprechender Verzehr für Information oder Wissen nicht zu verzeichnen. Man kann im Gegenteil davon ausgehen, dass prozessrelevantes Wissen durch seine Anwendung und Verfeinerung zunimmt.

Zudem ist der Zweck der Modellierung von Geschäftsprozessen ein anderer als der, der der Modellierung von Produktionsprozessen in der Produktionstheorie zugrunde liegt: Die Modellierung von Geschäftsprozessen ist darauf gerichtet, den Ablauf einschließlich der Kontrollstrukturen eines Prozesses so detailliert zu beschreiben, dass nicht nur aussagekräftige Wirtschaftlichkeitsanalysen möglich sind, sondern dass auch die Gestaltung eines realen Prozesses angeleitet wird. Im Unterschied dazu sind die Modelle der Produktionstheorie darauf gerichtet, mögliche Zusammenhänge zwischen Input- und Outputgrößen in idealisierter, also stark vereinfachter Form zu veranschaulichen. Dabei wird i. d. R. davon abstrahiert, welcher Typ eines Produktionsfaktors, z. B. ein bestimmter Maschinentyp, in einem Produktionsprozess eingesetzt wird. Auch wird von einer Beschreibung des eigentlichen Prozesses weitgehend abstrahiert.

Vor diesem Hintergrund schlagen wir vier Kategorien von Ressourcen für die Modellierung von Geschäftsprozessen vor: *Betriebsressourcen i.e.S.*, *Software*, *Information* und *menschliche Arbeit*.

Betriebsressourcen i.e.S.

Betriebsressourcen i.e.S. umfassen Güter, deren Benutzung zur Durchführung eines Geschäftsprozesses nötig ist. Dazu gehören u.a. Gebäude, Räume, (Büro-) Maschinen, Kommunikationsverbindungen, Transportmittel, Rechner und periphere Geräte. Sie werden in einem Prozess benutzt. Ihr Gebrauch verursacht Kosten. Dabei handelt es sich um Mieten, Abschreibungen, Energiekosten sowie Kosten für den Verbrauch von Hilfsstoffen. Die explizite Modellierung von Hilfsstoffen wie etwa von Druckertinte als Ressource wird i. d. R. wenig sinnvoll sein, da sie für die gängigen Zwecke der Geschäftsprozessmodellierung nicht nötig ist. Stattdessen sollte es möglich sein, die mit der Nutzung der Ressourcen verbundenen Kosten zu erfassen. Im einfachen Fall könnte

dies etwa durch die Zuordnung von Stundensätzen von Maschinen geschehen. Falls sich die zeitbezogenen Kosten mit der Last der Maschine nennenswert ändern, müsste dies u.U. auch auszudrücken sein.

Software

Software gehört auch zu den Betriebsressourcen i.w.S. Es gibt aber gute Gründe dafür, Software getrennt zu behandeln. Im Hinblick auf die Automatisierung von Geschäftsprozessen kommt Software eine zentrale Rolle zu. Dabei sind spezifische Eigenschaften wie Plattformen, Schnittstellen etc. zu berücksichtigen, die für andere Betriebsressourcen nicht gelten. Im Hinblick auf eine differenzierte Erfassung von Software ist an Kategorien wie Betriebssysteme, Middleware, Anwendungssoftware oder Komponente zu denken. Es sollte möglich sein, wesentliche software-technische (z.B. Implementierungssprache, Datenbanknutzung etc.) und wirtschaftliche (z.B. Nutzungskosten, extern oder intern erstellt ...) Eigenschaften zu erfassen.

Information

Daten, Information und Wissen können ebenfalls als Betriebsressourcen aufgefasst werden. Da sie nicht nur für die Organisation und Durchführung von Geschäftsprozessen eine große Bedeutung haben, sondern auch für die Vorbereitung der Entwicklung von Software aus Geschäftsprozessmodellen, ist es in jedem Fall angeraten, sie gesondert zu behandeln. Die Differenzierung von Daten, Information und Wissen wird in der Literatur nicht einheitlich gehandhabt. Die folgende Begriffsabgrenzung ist u.E. brauchbar: Daten sind symbolische Repräsentationen, lediglich durch eine Syntax und eine rudimentäre, formale Bedeutung – die sich z.B. darin äußert, dass für Zahlentypen andere Operationen definiert sind als für Zeichenketten - gekennzeichnet sind. Information liegt vor, wenn Daten mit einer Semantik versehen sind, die den jeweils repräsentierten realweltlichen Sachverhalt kennzeichnet. Im Unterschied dazu besteht (explizites) Wissen nicht aus Aussagen über singuläre Instanzen, sondern über ganze Klassen, die zudem prinzipiell fehlbar sind und deshalb begründet sein sollten.

Es bleibt allerdings die Frage, ob eine solche Differenzierung im Hinblick auf die Zwecke der Geschäftsprozessmodellierung überhaupt benötigt wird. Daten im Sinne der vorgeschlagenen Unterscheidung werden in Geschäftsprozessmodellen wohl kaum abgebildet. Die subtilere Unterscheidung zwischen Information und Wissen erfordert eine menschliche Interpretation. Dies kann leicht dazu führen, dass der Sprachanwender sich überfordert sieht, zumal einige Informationsquellen, wie etwa Handbücher, sowohl Information als auch Wissen enthalten. Es scheint daher nicht notwendig, explizit zwischen Information und Wissen zu unterscheiden. Um eine differenzierte Modellierung zu unterstützen, die eine gehaltvolle Grundlage für die Analyse von Geschäftsprozessen unterstützt, ist allerdings eine Unterscheidung verschiedener Informationsarten und der mit ihnen einhergehenden (Re-) Präsentation angeraten. Dazu muss es möglich sein, verschiedene *Arten von Dokumenten* zu unterscheiden, also z.B. Formulare, Briefe, Handbücher, Fachzeitschriften, Nachschlagewerke etc. Um Medienbrüche feststellen zu können, ist es wichtig, dass für jeden Dokumenttyp die jeweilige Repräsentationsform angegeben wird. Dazu gehört das verwendete Medium (z.B. Papier, Mikrofiche, digitaler Datenträger ...), sowie ggfs. der Aufbewahrungsort, das Dateiformat und die verwendete Zeichencodierung. Ergänzend sollte es möglich sein, zu erfassen, ob es sich um interne oder externe Quellen handelt und welche Bedeutung ihnen im Hinblick auf die Wettbewerbsfähigkeit zukommt.

Die Strukturen von *Daten* oder *Objekten*, also Entitätstypen oder Klassen, sollten getrennt von Dokumenten beschrieben werden, weil ihre Darstellung besonderen Anforderungen der Software-Entwicklung genügen muss. Sinnvollerweise sollte es dazu möglich sein, Daten- bzw. Objektmodelle in geeigneter Form zu referenzieren. Um die Software-Entwicklung angemessen zu unterstützen, sollte eine differenzierte Referenzierung von Attributen oder Operationen möglich sein. Daten- bzw. Objektmodelle dienen der Beschreibung von Informationen, die im internen Informationssystem residieren. Da der rechnergestützten Kommunikation mit externen Akteuren eine zunehmende Bedeutung zukommt, sollte es möglich sein, entsprechende *Schnittstellen* und *Kommunikationsprotokolle* zu beschreiben.

Das mit der Ressourcenmodellierung verbundene Abstraktionsniveau ist bereits diskutiert worden. Grundsätzlich hängt die Wahl des Abstraktionsniveaus vom jeweiligen Modellierungszweck und dem möglichen Modellierungsaufwand ab. Häufig wird die Beschreibung von Typen hinreichen. Es mag aber sein, dass man – wie in Abb. 5 dargestellt – auch Instanzen von Typen darstellen möchte, die ebenfalls Typen sind. In Einzelfällen mag auch der Verweis auf konkrete Instanzen angemessen erscheinen – etwa bei der Erfassung der URL von Verzeichnisdiensten (sog. „registries“), die aktuelle Schnittstellenbeschreibungen bereitstellen. Die Modellierungssprache sollte es also erlauben, Ressourcen auf möglichst vielen unterschiedlichen Abstraktionsebenen zu beschreiben.

Menschliche Arbeitskraft

Leistung und Qualität von Geschäftsprozessen hängen i. d. R. wesentlich von den beteiligten Menschen ab. Gleichzeitig erfordert die Beschreibung der Ressource „menschliche Arbeitskraft“ ganz spezifische Kriterien. Deshalb ist eine getrennte Modellierung dieser Ressource sinnvoll. Dabei sollte es möglich sein, die nötige Qualifikation der beteiligten Personen zu erfassen sowie ggfs. anzugeben, ob es sich um Mitarbeiter, Freiberufler oder Mitarbeiter externer Institutionen handelt. Um eine differenzierte Modellierung zu erlauben und gleichzeitig die Organisation eines Geschäftsprozessstyps detailliert angeben zu können, empfiehlt es sich, dazu Rollen bzw. Organisationseinheiten (i. d. R. Stellen) angeben zu können. Da Rollen und Organisationseinheiten in mehreren Prozessstypen genutzt werden können, sollte es möglich sein, ein entsprechendes Rollen bzw. Organisationsmodell (s. 4.2.7) zu referenzieren. In diesem Modell können dann auch Angaben über Nutzungskosten abgelegt werden.

Kapazitäten

Die Analyse und (Re-) Organisation von Prozessen erfordert u. U. die Berücksichtigung der Kapazität von Ressourcen. Darunter ist allgemein das Leistungsvermögen pro Zeiteinheit zu verstehen. So kann etwa die Kapazität eines Sachbearbeiters bei einer Bank darin bestehen, pro Stunde zwei Kreditanträge zu prüfen. Wenn ein Unternehmen über 10 Sachbearbeiter verfügt, können pro Stunde maximal zwanzig Anträge geprüft wer-

den. Ein Datenbankserver mag 5.000 Lesezugriffe pro Sekunde zulassen, ein Drucker zwanzig Seiten pro Minute drucken etc. Die Zuordnung von Kapazitäten zu Ressourcen eröffnet die Möglichkeit, Ausführungszeiten von Prozessen zu berechnen. Dabei sind allerdings gravierende Einschränkungen zu berücksichtigen. So ist die Kapazität einzelner Ressourcen möglicherweise nicht unabhängig vom Einsatz anderer Ressourcen. So kann die Kapazität eines Mitarbeiters u. U. erheblich gesteigert werden, wenn zur Durchführung seiner Aufgaben Software verwendet. Das impliziert einen erheblichen Modellierungs- und vor allem Pflegeaufwand: Durch die Veränderung von Ressourcen sind die Kapazitäten der davon betroffenen anderen Ressourcen u. U. anzupassen. Gleichzeitig ist zu berücksichtigen, dass sich die Kapazitäten einzelner Ressourcen, hier ist vor allem an Rechner und periphere Geräte zu denken, im Zeitverlauf häufig ändern mögen. Erschwerend kommt noch hinzu, dass Ressourcen i. d. R. nicht nur von einer Prozessinstanz genutzt werden, sondern von mehreren Instanzen eines Typs oder auch von Instanzen verschiedener Prozesstypen. Die Festlegung der tatsächlich verfügbaren Kapazität von Ressourcen für eine Prozessinstanz wird zusätzlich dadurch erschwert, dass Ressourcen zum Teil nicht permanent verfügbar sind: Maschinen mögen zeitweise defekt sein, Mitarbeiter fallen durch Urlaub oder Krankheit aus. Wegen des erheblichen Aufwands macht die Erfassung von Kapazitäten nur Sinn, wenn ein entsprechender Nutzen zu erwarten ist – durch die Optimierung der Organisation über verschiedenen Prozesstypen hinweg bzw. durch die Durchführung von Simulationen. Im Hinblick auf die Durchführung von Simulationen sollte die Modellierung prototypischer Instanzen unterstützt werden (vgl. [Fran94]).

Konkretisierung der Anforderungen:

Um den Modellierungsaufwand zu reduzieren und gleichzeitig die Integrität von Modellen zu fördern, sollte eine Modellierungssprache wichtige Ressourcentypen bereitstellen. Diese können Sprachbestandteil oder Sprachanwendung sein. Angesichts der Besonderheiten von Information und menschlicher Arbeit ist es wünschenswert, dass spezielle Konzepte zur Modellierung dieser Ressourcenarten bereitgestellt werden. Insbesondere ist zu fordern, dass Objekt- bzw. Datenmodelle sowie Rollen- bzw. Organisationsmodelle referenziert werden können. Darüber hinaus ist eine branchenspezifische

Ressourcentypisierung denkbar, die je nach Branche unterschiedliche Typen von Ressourcen vordefiniert. Ergänzend dazu sollte es möglich sein, spezielle Ressourcentypen einzuführen. Um eine konsistente Modellierung von Beziehungen, an denen Ressourcentypen beteiligt sind, zu unterstützen, sollte die Modellierungssprache geeignete Beziehungstypen bereitstellen. Das gilt zum einen für Beziehungen zwischen Ressourcentypen. Neben Generalisierungs- bzw. Spezialisierungsbeziehungen ist hier an Aggregations-, Benutzungs-, Substitutionsbeziehungen u.ä. zu denken, die jeweils soweit wie möglich sinnvoll mit einer formalen Semantik versehen sein sollten, um formale Analysen zu unterstützen. Auf der anderen Seite müssen Beziehungen zwischen Ressourcentypen und anderen Konzepten der Geschäftsprozessmodellierung möglich sein, z.B. Benutzungs- oder Verbrauchsbeziehungen zu Prozesstypen. Um eine Grundlage für die Berechnung von Prozesskosten zu schaffen, sollte es möglich sein, Ressourcentypen Kosten zuzuordnen. Dabei kann es sich beispielsweise um durchschnittliche, minimale und maximale Kosten pro Zeiteinheit handeln oder aber um durchschnittliche, minimale und maximale Kosten pro verbrauchter Instanz. Voraussetzung dafür sind allerdings gehaltvolle Informationen über die jeweiligen Kostenstrukturen, also etwa über das Verhältnis von variablen und fixen Kosten, da sonst eine Kalkulation wenig Sinn macht. Kapazitäten beziehen sich auf bestimmte Leistungen, die innerhalb bestimmter Prozesse zu erbringen sind. Sie können deshalb einer Ressource nicht unabhängig von einer Leistung zugeordnet werden. Wenn die Modellierung von Kapazitäten erforderlich ist (was eher selten der Fall sein dürfte), muss es deshalb möglich sein, Beziehungen zwischen Prozessen und Ressourcen durch Angaben über Kapazitäten anzureichern. Da Ressourcentypen mehreren Prozesstypen zugeordnet werden können, sollte es ggfs. möglich sein, eine maximale Zahl von Prozessinstanzen angeben zu können, an denen eine Prozessinstanz gleichzeitig beteiligt sein darf.

4.2.5. Prozessgegenstand

Ein Geschäftsprozess ist i.d.R. auf einen Gegenstand gerichtet. Dabei kann es sich um eine zu erstellende Leistung bzw. ein Produkt handeln oder um die Vorbereitung und Abwicklung einer Transaktion. Beispiele: „Darlehensvertrag“, „PCs“. Da es sich dabei

nicht um eine Ressource handelt, ist ein entsprechendes Sprachkonzept sinnvoll. Ein Prozessgegenstand hat eine enge Beziehung zum jeweiligen Prozesstyp, stellt allerdings eine andere Abstraktion dar. So kann etwa der Prozessgegenstand ‚Darlehensvertrag‘ in den Prozesstypen ‚Abschluss eines Darlehensvertrags‘ und ‚Überprüfung eines Darlehensvertrags‘ vorkommen. Da Prozessgegenstände in großer Bandbreite variieren können, ist es sinnvoll, die Definition von Prozessgegenstandstypen zu erlauben. Ähnlich wie auch bei der Definition von Ressourcentypen können entsprechende Sprachkonzepte mehr oder weniger speziell sein. So kann eine Sprache vordefinierte Typen von Prozessgegenständen bereitstellen (als Sprachanwendung) oder die Spezifikation solcher Typen erlauben. Im Hinblick auf die Formulierung von Prozesszielen kann der Qualität eines Prozessgegenstands eine zentrale Bedeutung zukommen. Insofern sollte eine Beschreibung der Qualität unterstützt werden.

Im Unterschied zu Ressourcen ist u.E. die Typisierung von Prozessgegenständen i.d.R. weniger wichtig, da es zumeist für einen Prozesstyp nur einen Prozessgegenstand gibt, weshalb ein Gegenstandstyp etwa für die Formulierung von Integritätsbedingungen innerhalb des jeweiligen Prozesstyps nicht relevant ist. Wenn man allerdings ein System mehrerer Geschäftsprozesse unterschiedlichen Typs betrachtet, können Gegenstandstypen geeignet sein, Regeln über Prioritäten von Prozessinstanzen auszudrücken.

Konkretisierung der Anforderungen

Es sollte in jedem Fall möglich sein, einen Geschäftsprozessgegenstand zu definieren. Anders als bei Ressourcen ist dazu für die meisten Anwendungsfälle ein semantisch flaches Konzept hinreichend. Eine größere Ausdrucksmächtigkeit wird durch Sprachkonzepte ermöglicht, die die Spezifikation domänenspezifischer Gegenstandstypen erlauben. Die Bereitstellung einer Menge vordefinierter Gegenstandstypen unterstützt demgegenüber Komfort und Sicherheit der Benutzung. Um die Qualität von Prozessgegenständen beschreiben zu können, sollte in jedem Fall eine entsprechende natürlich-sprachliche Annotation möglich sein. Wenn es möglich ist, Qualität mittels messbarer Kriterien zu operationalisieren, sollte es möglich sein, diese Kriterien abzubilden. Da der Prozessgegenstand i. d. R. auch bei der Formulierung von Prozesszielen eine zentra-

le Rolle spielt, sollte es möglich sein, entsprechende – ggfs. qualifizierbare – Beziehungen zwischen Prozessgegenstand und Prozessziel auszudrücken.

4.2.6. Modellierung der Organisationsstruktur

Um die in einem Prozesstyp erforderliche Beteiligung menschlicher Akteure zu beschreiben, ist die bereits erwähnte Ressource ‚Arbeitskraft‘ i.d.R. nicht sinnvoll. Vielmehr sind die beteiligten Arbeitskräfte differenzierter und zugleich abstrakter zu beschreiben. Ein Ansatz dazu bietet die Organisationsstruktur bzw. die Aufbauorganisation. Die Aufbauorganisation besteht aus Organisationseinheiten (z.B. Bereiche, Abteilungen, Gruppen), die wiederum andere Organisationseinheiten beinhalten können. Die kleinste Organisationseinheit ist die Stelle. Die Berücksichtigung der Aufbauorganisation ist nicht zuletzt für die Modellierung von Prozessen, die später über Workflow-Management-Systeme abgewickelt werden, erforderlich, da diese Systeme den Einsatz von Personal und Applikationen in Bezug auf die Prozesse koordinieren (vgl. bspw. [GeGa99]).

Zur Darstellung der Aufbauorganisation werden häufig Organigramme verwendet. Sie stellen Organisationseinheiten und zwischen diesen bestehende Beziehungen grafisch dar. Sowohl in der Praxis wie auch in Teilen der Fachliteratur finden sich ungenaue bzw. mehrdeutige Organigramme. Das ist zumeist darauf zurückzuführen, dass die Beziehungen zwischen den Organisationseinheiten nicht sorgfältig differenziert werden (vgl. [Fran01], S. 44 ff.). Um eine differenzierte Beschreibung von Organisationseinheiten zu ermöglichen, sollte eine Modellierungssprache deshalb eine Typisierung von Beziehungen unterstützen. Zwischen Organisationseinheiten können zunächst zwei Arten von Beziehungen unterschieden werden: Aggregations- und Weisungsbeziehungen. Eine Aggregationsbeziehung liegt vor, wenn eine Organisationseinheit in einer anderen enthalten ist, etwa eine Stelle in einer Abteilung. Weisungsbeziehungen können weiter unterschieden werden in funktionsgebunden, disziplinarisch, fachlich und objektgebunden (vgl. [Groc82], S. 173 f.). Um eine konsistente Verwendung der Beziehungstypen zu unterstützen, sollten die entsprechenden Sprachkonzepte durch eine angemessene Semantik gekennzeichnet sein. So dürfen Aggregations- und Weisungsbeziehungen auf

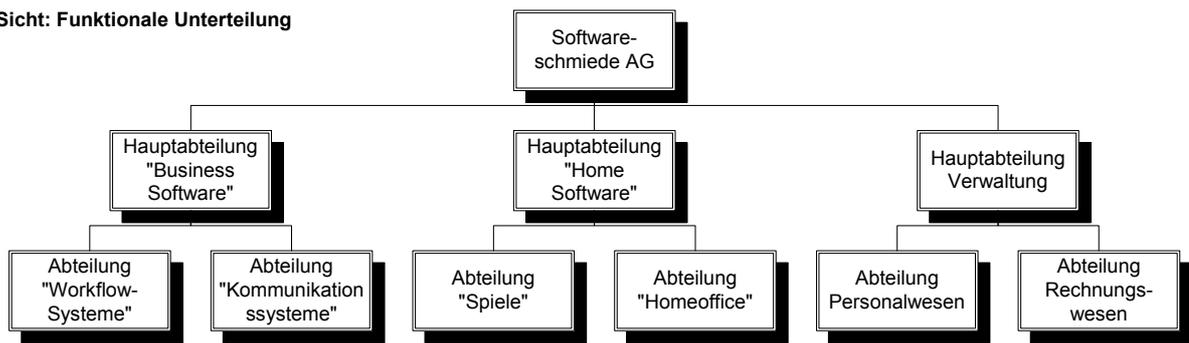
der Instanzenebene nicht zyklisch sein. Weitere Integritätsbedingungen für die Verwendung von Weisungsbeziehungen können sich dadurch ergeben, dass gewisse Gestaltungsregeln einzuhalten sind. So ist im *Einliniensystem* eine Organisationseinheit immer nur genau einer anderen untergeordnet, während es das *Mehrliniensystem* erlaubt, dass eine Organisationseinheit Weisungen von mehreren übergeordneten Einheiten erhält. Ein weiterer Gestaltungsaspekt ergibt sich durch das jeweils angewandte Prinzip der Arbeitsteilung. Bei funktionsorientierter Arbeitsteilung werden die Organisationseinheiten durch die jeweils zu erfüllende Funktion differenziert, z.B.: Abteilung ‚Rechnungswesen‘. Eine objektorientierte Differenzierung liegt dann vor, wenn der zu bearbeitende Gegenstand im Vordergrund steht – z.B.: Hauptabteilung ‚Lebensversicherung‘. Mehrdimensionale Organisationsstrukturen wie die Matrixorganisation (vgl. [Schr98], S. 180 ff.) kombinieren objektorientierte Arbeitsteilung mit funktionsorientierter. Dabei ist allerdings zu berücksichtigen, dass eine formale Differenzierung von Funktion und Objekt nur dann möglich ist, wenn eine Organisationseinheit explizit ausgezeichnet ist. Dabei bietet eine entsprechende Typisierung (etwa durch die Untertypen ‚Funktionseinheit‘ und ‚Objekteinheit‘) den Vorteil, dass daran anknüpfende Integritätsbedingungen von einem Werkzeug früher überprüft werden können.

Während die Modellierung i. d. R. auf Typen von Geschäftsprozessen gerichtet ist, werden bei der Modellierung der Organisationsstruktur zumeist konkrete Instanzen beschrieben, also z.B. eine Marketing-Abteilung. Man könnte hier zwar einwenden, dass es sich dabei um die einzige Instanz eines Typs handelt, das ist aber nicht überzeugend. Wenn etwa die Marketing-Abteilung eine Subeinheit der Hauptabteilung ‚Vertrieb‘ ist, dann handelt es sich dabei offenbar nicht um eine ‚typische‘ Eigenschaft von Marketing-Abteilungen, sondern um eine spezielle Konstellation im Einzelfall. Außerdem liegt es nahe, ein Organisationsmodell auch als Dokumentation des aktuellen Zustands eines Unternehmens zu verwenden. In diesem Fall würde man etwa den Namen des aktuellen Abteilungsleiters zuordnen wollen. Offensichtlich handelt es sich dabei um eine Eigenschaft, die nur auf Instanzenebene Sinn macht. Neben den Fällen, in denen eine bestimmte Organisationseinheit nur einmal im Unternehmen existiert, gibt es auch Organisationseinheiten, für die es mehrere Instanzen eines Typs zu verzeichnen sind,

etwa die Stelle ‚Kreditsachbearbeiter‘. Hier würde man sinnvollerweise einen Typ beschreiben, um nicht jede Instanz einzeln erfassen zu müssen. Mit dem Typ sollte es möglich sein, Angaben zur Zahl der verfügbaren Instanzen zu machen.

Die Visualisierung der Aufbauorganisation kann dabei aus unterschiedlichen Blickwinkeln erstellt werden. So können z.B. Weisungsbefugnissen oder Aggregationsbeziehungen zwischen Organisationseinheiten im Unternehmen dargestellt werden. Im Hinblick auf die Notation ist festzustellen, dass Organigramme (s. Beispiele in Abbildung 8) wegen ihrer Verbreitung und dem daraus abzuleitenden Beitrag zur Anschaulichkeit von Modellen zwar prinzipiell empfehlenswert sind, dass aber die mit ihnen einhergehende Mehrdeutigkeit aufgelöst werden sollte. Außerdem ist zu berücksichtigen, dass einfache Organigramme nicht hinreichen, beliebige Organisationsformen (beispielsweise der Matrixorganisation) darzustellen, da sie lediglich 1:1 oder 1:n-Beziehungen zwischen den Organisationseinheiten erlauben (vgl. BeRo96). Die entsprechende Notation sollte auch die Darstellung von n:m-Beziehungen erlauben.

Sicht: Funktionale Unterteilung



Sicht: Weisungsbefugnisse

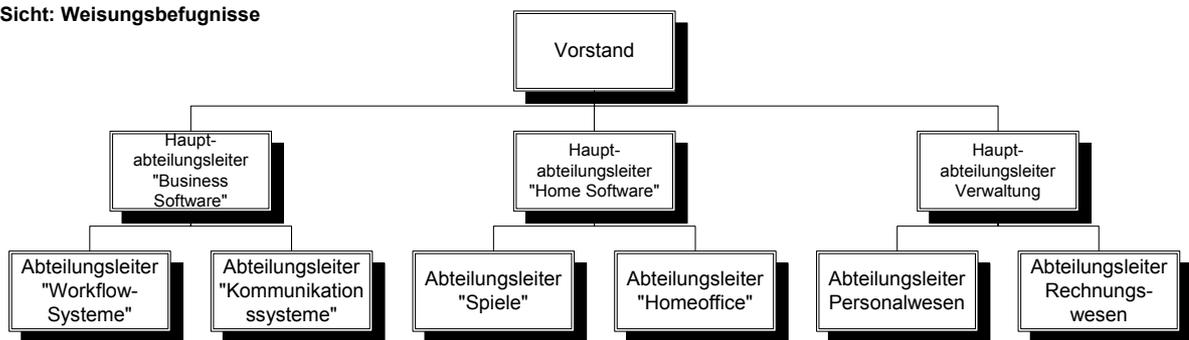


Abbildung 8: Organigramme stellen verschiedene Sichten dar

Konkretisierung der Anforderungen

Eine einschlägige Modellierungssprache sollte Konzepte zur Modellierung von Organisationseinheiten bereitstellen. Dazu gehören einerseits Strukturen, die wesentliche Eigenschaften von Organisationseinheiten. Im Hinblick auf die Unterstützung eines effizienten Personalmanagements gehört dazu nicht zuletzt eine detaillierte Struktur zur Beschreibung von Stellenprofilen. Dabei sollte es möglich sein, zwischen Ist-Profil und Wunsch-Profil zu unterscheiden. Angesichts der Vielfalt möglicher Organisationseinheiten muss es möglich sein, diese Strukturen an individuelle Anforderungen anzupassen. Andererseits sollten Konzepte zur Abbildung von Beziehungen zwischen Organisationseinheiten vorhanden sein. Insbesondere sollten Aggregations- und Weisungsbeziehungen unterstützt werden. Die Sprachspezifikation sollte eine unsachgemäße Verwendung von Beziehungen, z. B. zyklische Aggregationsbeziehungen, ausschließen. Auch

das Angebot vordefinierter Typen von Organisationseinheiten trägt zur Konsistenz bei. Wenn etwa Typen wie ‚Hauptabteilung‘, ‚Abteilung‘, ‚Gruppe‘ Bestandteil der Sprache sind bzw. als Sprachanwendung zusammen mit der Sprache angeboten werden, dann verstößt eine Beziehung wie ‚Gruppe *beinhaltet* Hauptabteilung‘ entweder gegen die Sprachsemantik oder aber gegen explizite Integritätsbedingungen. Konkrete Organisationseinheiten wie etwa die oben erwähnte Marketing-Abteilung wären dann Instanzen der vorgegebenen Typen. Dem subtilen Unterschied zwischen Typ und Instanz einer Organisationseinheit kann dadurch Rechnung getragen werden, dass für jede Organisationseinheit optional die Zahl ihrer Instanzen angegeben werden kann. Gleichzeitig sollte es möglich sein, die Identität einzelner Instanzen auszudrücken, um etwa angeben zu können, dass an den Instanzen eines Prozesstyps jeweils zwei Stellen des Typs „Kreditsachbearbeiter“ beteiligt sein müssen. Da ggfs. nicht nur unternehmensinterne Organisationseinheiten zu berücksichtigen sind, sollte es möglich sein, das jeweilige Unternehmen bzw. die Institution mit zu erfassen.

Im Hinblick auf die Anschaulichkeit von Modellen der Organisationsstruktur sollte auf jeden Fall eine Visualisierung unterstützt werden, die an den gängigen Organigrammen orientiert ist.

4.2.7. Konzepte zur Modellierung von Rollen

Ergänzend zu den organisatorischen Einheiten, die an der Ausführung von Prozessen eines bestimmten Typs beteiligt sind, sind *Rollen* zu berücksichtigen. Eine Rolle ist eine Abstraktion, die mit der Zuständigkeit für eine Aufgabe oder Funktion verbunden ist. Sie ist orthogonal zur Stelle als kleinster organisatorischer Einheit. Rollen können von personellen wie auch von maschinellen Aufgabenträgern (Maschinen, Software) übernommen werden. Personelle Aufgabenträger unterscheiden sich in unternehmensinterne und unternehmensexterne. Beispiele für externe Rollen sind Lieferant, Wettbewerber, Kunde. Beispiele für interne Rollen sind Qualitätsbeauftragter, Betriebsratsmitglied, Sicherheitsbeauftragter. Dabei ist allerdings zu berücksichtigen, dass z.B. Kunde auch als interne Rolle angesehen werden kann. Die detaillierte Beschreibung von Rollen erfordert u.U. die explizite Festlegung spezifischer Integritätsbedingungen. Beispielswei-

se könnte es nötig sein, festzulegen, dass die Rolle eines Sicherheitsbeauftragten nur von einem Gruppenleiter (Stelle) oder nur von einem Mitarbeiter, der die Qualifikation ‚Diplomingenieur‘ aufweist, wahrgenommen werden darf. Um eine komfortable und sichere Anwendung einer Sprache zu unterstützen, sollten Konzepte zur Abbildung wiederholt verwendeter Beziehungen zwischen Rollen in die Sprache integriert sein. Hier ist z. B. an Stellvertreter-Beziehungen zwischen Rollen, aber auch an Beziehungen zwischen Rollen und Stellen, etwa um auszudrücken, dass eine Rolle nur von Inhabern eines bestimmten Stellentyps wahrgenommen werden kann. Da in einem Unternehmen eine Rolle eines Typs mehrfach vorkommen kann, sollten die entsprechenden Sprachkonzepte an Typen orientiert sein, es also z. B. erlauben auszudrücken, wie viele Instanzen es minimal und/oder maximal geben darf bzw. wie viele es tatsächlich gibt.

Konkretisierung der Anforderungen

Es ist eine große Zahl ganz unterschiedlicher Rollentypen denkbar. Auch vordergründig in vielen Unternehmen vorkommende Rollentypen wie ‚Kunde‘, ‚Qualitätsbeauftragter‘ oder ‚Sicherheitsingenieur‘ sind nicht einheitlich festgelegt. Es scheint deshalb wenig sinnvoll, entsprechenden Rollen als Typen in die Sprache zu integrieren. Vielmehr sollte es den Sprachanwendern möglich sein, individuelle Rollentypen zu definieren, indem Eigenschaften sowie zulässige Beziehungen zu anderen Konzepten der Organisationsmodellierung angegeben werden. Konzepte zur Abbildung von Rollen sollten formal Typen sein. Die Unterscheidung zwischen Typen und Instanzen kann gegenüber dem Sprachanwender aber weitgehend verborgen werden.

4.2.8. Konzepte zur Modellierung unternehmensübergreifender Geschäftsprozesse

Integrierte, unternehmensübergreifende Prozesse haben in den letzten Jahren eine zunehmende Bedeutung erlangt. Das gilt z. B. für die Beschaffung („supply chain management“) oder auch für den Vertrieb – hier ist etwa an Electronic Commerce zu denken. Prinzipiell werden unternehmensübergreifende Geschäftsprozesse genauso modelliert wie unternehmensinterne. Es gibt allerdings eine Reihe zusätzlicher Aspekte, deren Darstellung eine Modellierungssprache unterstützen sollte. Zunächst muss es möglich

sein, Teilprozesse, die von Externen ausgeführt werden, als solche zu kennzeichnen. Die Funktion einer externen Organisation (z. B. ‚Kunde‘, ‚Logistikpartner‘ etc.) sollte durch die Zuordnung von Rollen darstellbar sein. Da bei unternehmensübergreifenden Prozessen häufig ein Daten- bzw. Dokumentaustausch zwischen verschiedenen Informationssystemen unterstützt werden muss und dazu aus gutem Grund nach Möglichkeit auf Standardschnittstellen zurückgegriffen werden sollte, sollte die Modellierungssprache nicht nur, wie bereits in 4.2.4 gefordert, Konzepte wie ‚Schnittstelle‘ oder ‚Protokoll‘ beinhalten, sondern auch eine Bibliothek wichtiger Standards, etwa aus dem Bereich des elektronischen Geschäftsverkehrs (vgl. [Fran00]) oder des elektronischen Zahlungsverkehrs. Da bei unternehmensübergreifenden Prozessen auch Logistikprozesse, also etwa die Auslieferung von Waren an Kunden, zu berücksichtigen sind, ist es wünschenswert, die damit verbundenen Besonderheiten, wie etwa das eingesetzte Transportmittel, abbilden zu können. Schließlich können in Geschäftsprozessen, an denen externe Partner beteiligt sind, spezifische Störungen bzw. Ausnahmen auftreten, also z. B. Problem bei der Durchführung von Finanztransaktionen oder der Ausfall von Transportmitteln bei einem Logistikpartner. Auch wenn die Behebung solcher Störungen zu meist bei den zuständigen externen Partnern liegen dürfte, kann es doch wünschenswert sein, eine einheitliche Ausnahmebehandlung zu definieren – etwa in Form einer standardisierten Benachrichtigung des externen Partners.

Konkretisierung der Anforderungen

Eine Sprache zur Organisationsmodellierung sollte spezifische Konzepte und anschauliche Symbole für die Modellierung unternehmensübergreifender Geschäftsprozesse bereitstellen. Dazu gehören Konzepte zur Beschreibung von Logistikprozessen und solche zur Darstellung spezifischer Ausnahmen. Daneben sollten dem Modellierer die Beschreibungen wichtiger Standards für den unternehmensübergreifenden Datenaustausch mitgeliefert werden.

4.2.9. Berücksichtigung weiterer betriebswirtschaftlicher Aspekte

Modelle von Geschäftsprozessen sollten die wirtschaftliche Analyse eines Unternehmens unterstützen. Zu diesem Zweck sollte es möglich sein, das Modell eines Geschäftsprozestyps mit Angaben für das Controlling anzureichern. Dazu gehört etwa die Möglichkeit, das Verfahren zur Vereinbarung und Überprüfung von Prozesszielen zu beschreiben – i. d. R. wohl natürlichsprachlich. Auch sollte eine Beschreibung der jeweils angewandten Kostenrechnungsmethode möglich sein. Um die Wirtschaftlichkeit eines Prozesstyps beurteilen zu können, sind auch Angaben über die mittels eines Prozesses erbrachten Leistungen erforderlich. Hier ist etwa an Deckungsbeiträge zu denken, die pro Instanz eines Prozesstyps durchschnittlich erwirtschaftet werden. Eine Modellierungssprache sollte es ermöglichen, entsprechende Angaben auszudrücken.

Im Hinblick auf das Prozessmanagement ist es hilfreich, wenn einem Prozessmodell kritische Erfolgsfaktoren, also Faktoren, die für das Erreichen der Prozessziele eine herausragende Bedeutung haben, zugeordnet werden können. Um eine kontinuierliche Steigerung der Prozesseffizienz zu unterstützen, sind Erfahrungen bei der Durchführung von Prozessen systematisch auszuwerten. Dazu sollte es möglich sein, solche Erfahrungen in strukturierter Form an das Modell eines Prozesstyps zu annotieren.

4.3. Kontrollstrukturen

Kontrollstrukturen sind der Kern aller Prozessmodellierungssprachen, da sie zeitliche und logische Abfolgen einzelner Teilprozesse innerhalb eines (Gesamt-) Prozesses darstellen. Angesichts der großen Varianz von Geschäftsprozessen sollte eine Prozessmodellierungssprache möglichst alle denkbaren Kontrollstrukturen darzustellen erlauben.

4.3.1. Sequenz

Die einfachste Abfolge ist die lineare Abfolge (Sequenz). Eine Sequenz umfasst mindestens zwei Teilprozesse A und B. Sie stellt eine Reihenfolge zwischen den Teilprozessen A und B her (A vor B oder B vor A).

4.3.2. Bedingungen und Regeln

Um Prozessverläufe ausdrücken zu können, die mit dem Zustand einer konkreten Instanz variieren, ist es nötig, entsprechende Regeln ausdrücken zu können. Eine Regel legt also fest, welcher Prozessverlauf (Aktionsteil der Regel) in Abhängigkeit vom Zutreffen einer Bedingung (Bedingungsteil der Regel) auszuwählen ist. Allgemein ausgedrückt ist eine Bedingung eine Aussage, die, wenn sie sich als wahr erweist, eine andere Aussage impliziert. Eine differenzierte Betrachtung des Begriffs findet sich in [Kamb80]. Für unsere Zwecke ist eine leicht modifizierte Begriffsfassung hilfreicher: Bedingungen sind Aussagen,

- deren Wahrheitswert sich mit den Instanzen eines Geschäftsprozessstyps bzw. während der Lebenszeit einer Instanz ändern kann.
- von deren Wahrheitswert die Durchführung eines Prozesses abhängen kann.

Es sind viele Regeln zur Festlegung der Kontrollstrukturen von Geschäftsprozessen denkbar. Die Überprüfung einer Bedingung während der Laufzeit eines Prozesses führt zu einem Ereignis (entweder die Bedingung trifft zu oder die Bedingung trifft nicht zu). Bedingungen sind also vergleichbar mit all denen Aussagen, die zur Beschreibung von Ereignissen verwendet werden (s. 4.2.2). Insofern sind auch die Anforderungen an die Formulierung von Bedingungen mit denen, die wir an die Formulierung von Ereignissen gestellt haben, vergleichbar. Der Aktionsteil einer Regel legt den Prozess fest, der im Fall des Zutreffens der Bedingung ausgeführt werden soll. Ergänzend kann er auch noch den Prozess festlegen, der auszuführen ist, wenn die Bedingung nicht zutrifft.

4.3.3. Alternative Ausführung

Immer dann, wenn nach Beendigung eines Teilprozesses alternativ zwischen mehreren Folgeprozessen gewählt werden kann, wird die alternative Ausführung benötigt. Die möglichen Alternativen werden durch eine Menge von Folgeprozessen dargestellt. Die alternative Ausführung besagt, dass genau ein Folgeprozess aus dieser Menge auszuwählen ist, um den Prozess fortzusetzen. Das Kriterium, das die Wahl eines Folgeprozesses bestimmt, kann, muss aber nicht explizit angegeben werden. In vielen Fällen

reicht es aus, sich auf die Wahl zwischen zwei Alternativen zu beschränken. Dazu genügt ein boolescher Ausdruck. Mitunter sind allerdings alternative Ausführungen zu modellieren, in denen mehr als zwei Fälle zu unterscheiden sind. Sie können zwar auf eine Menge dualer Alternativen, die jeweils über boolesche Ausdrücke geregelt werden, abgebildet werden. Im Hinblick auf den Modellierungskomfort und die Übersichtlichkeit von Modellen ist es allerdings wünschenswert, dass eine Modellierungssprache auch die Darstellung von Fallunterscheidungen mit mehr als zwei Fällen unterstützt.

4.3.4. Parallelität

Unter Parallelität im engeren Sinne versteht man die zeitgleiche oder sich zeitlich überschneidende Ausführung von Prozessen. Die Parallelisierung von Prozessen ist ein häufig eingesetztes Mittel zur Optimierung von unternehmerischen Abläufen. Vor allem zur Spezifikation von Soll-Modellen einer Ablauforganisation, in denen explizit klar gemacht werden soll, dass Prozesse (z.B. aus Gründen der Prozessoptimierung) zugleich ausgeführt werden sollen ist es notwendig, die Parallelität mit einer Prozessmodellierungssprache abbilden zu können. Auf für die Überprüfung der Korrektheit von Prozessen kann die explizite Modellierung der Parallelität einen sinnvollen Dienst erweisen: Parallele Prozesse dürfen nicht die gleichen Ressourcen(-instanzen) verwenden.

4.3.5. Nebenläufigkeit

Parallelität wird häufig mit Nebenläufigkeit verwechselt. Nebenläufigkeit bedeutet, dass zwei oder mehr unabhängige Prozesse parallel, oder in beliebiger Reihenfolge hintereinander ablaufen können. Die Parallelität ist daher nur ein Spezialfall der Nebenläufigkeit. Falls die Modellierung von nebenläufigen Prozessen A und B durch die Modellierungssprache nicht unterstützt wird, kann sie dennoch dargestellt werden, in dem explizit alle erlaubten Ausführungsreihenfolgen (A vor B, B vor A , A parallel zu B) als nicht deterministisch auszuwählenden Alternativen modelliert werden. Dies führt jedoch schnell zu unnötig unübersichtlichen und schwer verständlichen Modellen.

4.3.6. Unspezifizierte Reihenfolge

Es ist denkbar, dass für Teile eines Geschäftsprozesses die Ausführungsreihenfolge einer Menge von (Teil-) Prozessen nicht spezifiziert wird. Das kann darin begründet sein, dass man in einer frühen Phase der Prozessanalyse bewusst von einzelnen Details der Kontrollstrukturen abstrahieren möchte. Daneben kann es sein, dass es keine zwingenden Gründe für eine bestimmte Reihenfolge gibt und die Vorgabe einer Reihenfolge die Flexibilität der Prozessausführung im Einzelfall zu sehr einschränken würde. Das kann z. B. dann der Fall sein, wenn einzelne Ressourcen in mehreren Geschäftsprozessen eingesetzt werden. Dann kann es angeraten sein, die Reihenfolge einer Menge von Prozessen ad hoc im Einzelfall festzulegen. Es sollte deshalb möglich sein, für eine Menge von Teilprozessen offen zu lassen, in welcher Reihenfolge sie ausgeführt werden. Die unspezifizierte Reihenfolge entspricht der Nebenläufigkeit ohne erlaubte Parallelität.

4.3.7. Iteration

Teile von Prozessen müssen manchmal mehrfach durchlaufen werden. Sie iterieren. Die Anzahl der Iterationen kann auf verschiedene Weisen ausgedrückt werden. Es gibt unterschiedliche formale Möglichkeiten die Anzahl der Iterationen auszudrücken. Diese Möglichkeiten sind bereits aus gängigen höheren Programmiersprachen bekannt:

- Angabe eines absoluten oder variablen Wertes der Anzahl der Wiederholungen
- Vorgabe einer Bedingung, bei deren Eintreten keine weitere Wiederholung mehr stattfindet, oder
- Der Teilprozess wird solange wiederholt, wie eine bestimmte Bedingung erfüllt ist.

Die angegebenen Darstellungsformen korrespondieren mit den bekannten Programmierkonstrukten für Schleifensteuerungen:

- `FOR Zaehler = x TO y DO Prozess`
- `DO Prozess WHILE Bedingung`
- `REPEAT Prozess UNTIL Bedingung`

Die sog. Zählschleife ist dabei redundant, da sie durch jeweils eine der beiden anderen Iterationskonstrukte äquivalent ausgedrückt werden kann. Dessen ungeachtet ist ein entsprechendes Konstrukt hilfreich, um diese Art von Wiederholungen komfortabel ausdrücken zu können. Zur Darstellung einer Iteration muss es möglich sein, den Teilprozess, der wiederholt werden soll, abzugrenzen, die Art der Schleifensteuerung und die jeweils verwendete Bedingung anzugeben. Ähnlich wie bei Ereignistypen ist es hilfreich, wenn bestimmte Arten von Iterationen, etwa solche, die durch temporale Bedingungen festgelegt werden, durch eigens dafür vorgesehene Symbole dargestellt werden können.

4.3.8. Synchronisation

Werden in einem Prozess parallele oder nebenläufige Teilprozesse aufgespannt, so muss festgelegt werden, wann ein Prozess ausgeführt wird, der sich an den nebenläufigen bzw. parallelen Prozess anschließt. Die Teilprozesse müssen synchronisiert werden. Im Unterscheid dazu können alternative Prozesse zwar auch in gemeinsame Folgeprozesse münden. Sie werden aber nicht synchronisiert, da ja im Fall einer konkreten Prozessinstanz immer nur einer der alternativen Prozesse ausgeführt wird.

Für die Synchronisation von parallelen und nebenläufigen Prozessen sind verschiedene Bedingungen denkbar. Zumeist reichen dazu zwei Arten von Bedingungen aus. Zur Veranschaulichung gehen wir davon aus, dass zwei parallele Teilprozesse A und B in den gemeinsamen Folgeprozess C konvergieren:

- Der Prozess wird bei C nur dann fortgeführt, wenn die Teilprozesse A und B beendet sind.
- Der Prozess wird bei C bereits dann fortgeführt, wenn einer der Teilprozesse A und B beendet ist.

Dies gilt nicht nur für zwei, sondern auch für eine beliebige Zahl von Teilprozessen. Dabei wird im zweiten Fall gewöhnlich angenommen, dass die noch nicht beendeten Prozesse mit der Synchronisation abgebrochen werden. Falls diese Prozesse weiter laufen soll, ist ein Sprachmittel erforderlich, um dies auszudrücken. Daneben kann es vor-

kommen, dass die Art der Synchronisation sich nicht allein daran festmacht, dass ein oder mehrere Teilprozesse terminieren, sondern vielmehr davon abhängt, welche Resultate die Teilprozesse produzieren. Wenn etwa im Rahmen einer Auftragsbearbeitung die Prüfung der Bonität eines Kunden und die Prüfung der Lieferfähigkeit parallel verlaufen, ist der weitere Verlauf offensichtlich vom Ergebnis der Prüfungen abhängig: Nur wenn beide Prüfungen keine Einwände ergeben, kann der Auftrag weiter verarbeitet werden. Wenn bereits eine Prüfung zu einer Ablehnung führt, mag auch die Auftragsbearbeitung zu terminieren sein.

4.3.9. Transaktion

In Geschäftsprozessen werden i. d. R. Veränderungen an Informationen und auch an physischen Objekten vorgenommen. So wird etwa bei einer Auftragsbearbeitung der verfügbare Lagerbestand eines Gutes um die bestellte Menge reduziert. Zudem wird die gesamte Auftragssumme des Unternehmens entsprechend erhöht. Zur Vorbereitung eines Reparaturauftrags wird z. B. eine schwere Bohrmaschine auf einen LKW geladen. Anschließend werden der Monteur und das Fahrzeug im Informationssystem als reserviert eingetragen. Wenn nun ein Prozess vorzeitig terminiert, etwa weil eine Ausnahme auftritt, ist es erforderlich, alle in einem Zusammenhang stehenden Veränderungen an Objekten wieder rückgängig zu machen. In der Datenbanktheorie wird in diesem Zusammenhang das Konzept der Transaktion vorgeschlagen. Eine Transaktion besteht aus mehreren (Teil-) Prozessen, die nur dann Sinn machen, wenn auch der letzte Teilprozess ordnungsgemäß terminiert. Die sich daraus ergebenden Anforderungen für die Ausführung von Transaktionen, etwa die Atomarität, sind für eine Modellierungssprache nicht von Bedeutung. Eine Modellierungssprache sollte es ermöglichen, eine Transaktion als solche zu kennzeichnen. Dabei ist vorstellbar, dass der gesamte Prozess als eine Transaktion angesehen wird.

4.4. Ausnahmen

Ausnahmen sind mögliche Ereignisse innerhalb eines Prozesses, die hinreichend selten auftreten, um bei der Spezifikation der Kontrollstrukturen nicht explizit berücksichtigt werden zu müssen aber dennoch Konsequenzen für die weitere Ausführung eines Prozesses haben. Es handelt sich also um Störungen des gewöhnlichen Ablaufs. Diese reichen von alltäglichen Störungen wie etwa dem unsanften Abbruch einer von einem Kunden initiierten elektronischen Bestellung, dem Absturz eines Rechners oder dem Defekt eines peripheren Gerätes über Störungen, die sich durch Engpässe bei den benötigten Ressourcen ergeben – etwa durch den Ausfall eines Mitarbeiters – bis hin zu Unfällen oder gar Katastrophen. Ausnahmen sind Ereignisse und sollten auch genauso beschrieben werden wie Ereignisse. Da sie innerhalb von (Teil-) Prozessen auftreten, sollten sie mit Prozessen assoziiert werden können. Ergänzend dazu sollte es möglich sein, für einen Ausnahmetyp die jeweils erforderlichen Maßnahmen anzugeben. Es liegt nahe, dazu entsprechende Prozessmodelle zu verwenden. Durch die mehrfache Verwendung von Ausnahmetypen in einem Prozesstyp und über mehrere Prozesstypen hinweg wird eine einheitliche Ausnahmebehandlung bis hin zu einer unternehmensweiten „Business Continuity Plan“ (vgl. z.B. [Biwe03]) unterstützt – mit entsprechenden Vorteilen durch Wiederverwendung und organisatorische Integrität. Die Spezifikation einer Ausnahmebehandlung ist vergleichbar mit Regeln, die sich etwa in einem Organisationshandbuch finden. Mitunter ist in diesem Zusammenhang auch von Geschäftsregeln („business rules“) die Rede.

4.5. Integritätsbedingungen

Die Kontrollstrukturen eines Prozesstyps werden durch eine Reihe expliziter und impliziter Bedingungen ausgedrückt, etwa solchen, die zur Spezifikation alternativer Ausführungspfade oder zur Spezifikation der Ausnahmebehandlung verwendet werden. Dadurch werden jeweils prozedurale Ausführungsregeln ausgedrückt. Ergänzend dazu sind auch deklarative Integritätsbedingungen sinnvoll. Deklarative Integritätsbedingungen legen Propositionen fest, die entweder in bestimmten Situationen der Prozessausführung

oder grundsätzlich während der gesamten Lebenszeit eines Prozesses garantiert sein müssen. Anders formuliert legen sie Regeln fest, denen Prozesse genügen müssen. Dabei erlauben sie ein höheres Abstraktionsniveau als die erwähnten prozeduralen Regeln, da durch sie nicht explizit festgelegt wird, wie die Wahrung der Regeln erreicht wird.

4.5.1. Kardinalitäten

Häufig ergibt sich die Bedeutung eines Modellelements durch die Assoziation mit anderen Modellelementen. Für die Semantik solcher Assoziationen ist u.U. die Kardinalität der beteiligten Modellelemente von Bedeutung. Wenn etwa einem Geschäftsprozessstyp eine Stelle („Sachbearbeiter“) zugeordnet ist, kann durch die Zuordnung der Kardinalität 1,1 festgelegt werden, dass immer genau ein Sachbearbeiter für eine Prozessinstanz zuständig ist.

Konkretisierung der Anforderung:

Kardinalitäten sind ein wichtiges Ausdrucksmittel für die Modellierung von Geschäftsprozessen. Dabei ist eine min-max Notation wünschenswert, bei der für die minimalen und maximalen Werte beliebige natürliche Zahlen eingesetzt werden können (solange die minimale kleiner oder gleich der maximalen ist).

4.5.2. Preconditions

Preconditions oder Vorbedingungen sind Propositionen, die *vor* der Ausführung eines (Teil-) Prozesses wahr sein müssen, um einen konsistenten Prozessablauf zu ermöglichen. Sie können sowohl formal als auch informal formuliert sein, wie die folgenden Beispiele in unten abgebildeter Tabelle zeigen. Grundsätzlich können beliebige Bedingungen für die Formulierung von Preconditions verwendet werden. Preconditions kommt bei Prozessen, die von externen Partnern durchgeführt werden, eine besondere Bedeutung zu, da sie sich gut zur Abbildung vertraglicher Zusicherungen eignen.

Prozess	Vorbedingung	Formalisierbar?
Durchführung eines Ver-	- Der Name des Käu-	- Allenfalls indirekt

kaufsgesprächs	<p>fers ist dem Verkäufer bekannt.</p> <ul style="list-style-type: none"> - Der Verkäufer hat sich über persönliche Vorlieben / Präferenzen des Käufers informiert. 	<p>(z. B. Checkliste)</p> <ul style="list-style-type: none"> - Allenfalls indirekt (z. B. Checkliste)
Auftrag bestätigen	Lieferung zum Wunschtermin ist sichergestellt	Ja
Auslieferung bestellter Ware	<ul style="list-style-type: none"> - LKW ist verfügbar. - Kundenadresse ist ins Navigationssystem des LKW übertragen 	<ul style="list-style-type: none"> - eingeschränkt - ja

Tabelle 4: Vorbedingungen

Falls während der Durchführung eines Prozesses eine Precondition verletzt wird, sollte eine Ausnahme erzeugt werden. Für den entsprechenden Ausnahmetyp kann dann eine Ausnahmebehandlung festgelegt werden.

4.5.3. Postconditions

Postconditions oder Nachbedingungen sind Propositionen, die *nach* der Ausführung eines (Teil-) Prozesses wahr sein müssen, um einen konsistenten Prozessablauf zu ermöglichen. Für die Formulierung von Postconditions gilt das gleiche wie für die Spezifikation von Preconditions. Auch für Postconditions gilt, dass sie sich besonders für extern durchgeführte Prozesse eignen, da auf diese Weise überprüft werden kann, ob der

externe Partner die (vertraglichen) Zusicherungen eingehalten hat. In untenstehender Tabelle sind Beispiele für Postconditions dargestellt.

Prozess	Nachbedingung	Formalisierbar?
Durchführung eines Verkaufsgesprächs	<ul style="list-style-type: none"> - Der Name des Käufers ist dem Verkäufer bekannt. - Der Verkäufer hat sich über persönliche Vorlieben/Präferenzen des Käufers informiert. 	<ul style="list-style-type: none"> - Allenfalls indirekt (z. B. Checkliste) - Allenfalls indirekt (z. B. Checkliste)
Auftrag bestätigen	Lieferung zum Wunschtermin ist sichergestellt	Ja
Auslieferung bestellter Ware	Kundenadresse ist ins Navigationssystem des LKW übertragen	Ja

Tabelle 5: Nachbedingungen

Ebenso wie bei Preconditions sollte die Verletzung einer Postcondition zu einer Ausnahme führen, für die eine Ausnahmebehandlung definiert werden kann.

4.5.4. Prozesstypinvarianten

Pre- und Postconditions sind an bestimmte Prozesstypen gebunden und werden nur in bestimmten Ausführungssituationen überprüft. Im Unterschied dazu gibt es auch Propositionen, die während der gesamten Lebenszeit der Prozesse eines Typs erfüllt sein müssen. Man kann bei dieser Art von Integritätsbedingungen in Anlehnung an Klasseninva-

rianten ([Meye97], S. 146) von Prozessstypinvarianten sprechen. Sie sind gegenüber Pre- und Postconditions auf einem höheren Abstraktionsniveau angesiedelt und immer dann zu bevorzugen, wenn eine Integritätsbedingung mehrere (Teil-) Prozesse – sowie ggfs. zukünftige Modifikationen eines Prozessstyps – umfassen. Die Formulierung von Prozessstypinvarianten erfolgt analog zu der von Pre- und Postconditions. Auch hier sollte es möglich sein, die Verletzung einer Invariante als Ausnahme mit entsprechender Ausnahmebehandlung zu modellieren. Untenstehende Tabelle zeigt Beispiele für Prozessstypinvarianten.

Prozess	Invariante	Formalisierbar?
Auftragsbearbeitung	Eine Prozessinstanz darf nur von max. einem Sachbearbeiter bzw. dessen Stellvertreter betreut werden. (Ggfs. auch mittels Kardinalitäten auszudrücken)	Ja, Überprüfung allerdings problematisch
Verkaufsverhandlung	Der Verkaufspreis eines Produktes garantiert immer einen positiven Deckungsbeitrag.	Ja
Auslieferung bestellter Ware	Der Kunde kann sich zu jedem Zeitpunkt über den Zustand des Prozesses informieren.	Eingeschränkt

Tabelle 6: Invarianten

4.6. Unterstützung der Entwicklung von Informationssystemen

Eine typische Anwendung der Unternehmensmodellierung ist die Vorbereitung der Software-Entwicklung. Modelle von Geschäftsprozessen kommt dabei in mehrfacher Hinsicht eine besondere Bedeutung zu. So stellen sie einerseits eine Grundlage für den systematischen Entwurf von Daten- bzw. Objektmodellen dar, indem für jeden Prozess-typ der Informationsbedarf erfasst wird. Daneben bieten sie eine konzeptionelle Basis für den Entwurf von prozessunterstützenden Systemen – etwa Workflow Management Systemen. Um die Software-Entwicklung zu unterstützen, muss eine Sprache zur Modellierung von Geschäftsprozessen mit software-technischen Abstraktionen integriert werden.

4.6.1. Integration mit software-technischen Abstraktionen

Ein Geschäftsprozessmodell bietet eine gute Grundlage für die Entwicklung bzw. schrittweise Verfeinerung eines Daten- oder Objektmodells. Dazu ist für jeden Teilprozess festzustellen, welche Informationen jeweils benötigt oder erzeugt werden. Sofern diese Informationen mittels Objekten abgebildet werden, sollte es möglich sein, die jeweils benötigten Teilinformationen (etwa das Gehalt oder das Alter eines Mitarbeiters) durch die Zugriffsmethoden anzugeben. Der Verweis auf Attribute stellt eine Alternative dar, die allerdings gegen das Prinzip der Verkapselung verstößt – also weniger abstrakt ist – und zudem weniger mächtig ist (ein Zugriff auf das Alter wäre so nicht auszudrücken). Ergänzend zur Referenzierung von Methoden sollte es möglich sein, Objektkonstruktoren (bzw. entsprechende Klassenmethoden) zu referenzieren sowie das Löschen (bzw. die Freigabe) eines Objekts auszudrücken. Wenn innerhalb eines nicht weiter dekomponierten Prozesses nicht triviale Zugriffe auf Objekte erfolgen, sollte es möglich sein, die entsprechenden Kontrollstrukturen zu hinterlegen und Parameterlisten zu spezifizieren, z. B. in Pseudocode. Da u. U. in einem Prozess mehrere unterschiedliche Objekte einer Klasse benötigt werden, sollte es möglich sein, zwischen verschiedenen Objektidentitäten zu unterscheiden.

Die Behandlung von Ausnahmen erfolgt u. U. in automatisierter Form durch das Aufrufen einer Objektmethode. Deshalb sollte es möglich sein, Ausnahmen bzw. allgemein Ereignissen eine jeweils aufzurufende Objektmethode zuzuordnen.

Da die Datenmodellierung immer noch eine nicht unerhebliche Rolle spielt, sollte auch die Integration von Datenmodellen unterstützt werden. Dazu muss es vor allem möglich sein, Attribute von Entitätstypen zu referenzieren. Ergänzend dazu ist vorstellbar, dass Funktionen eines assoziierten Funktionenmodells referenziert werden, um einen abstrakteren Zugriff auf Daten ausdrücken zu können.

Konkretisierung der Anforderungen

Es sollte möglich sein, einem Prozess Attribute der Entitätstypen eines assoziierten Datenmodells bzw. – vorzugsweise - Methoden von Objekten eines assoziierten Objektmodells zuzuordnen. Auch muss es möglich sein, auszudrücken, dass Objekte gelöscht bzw. freigegeben werden sollten. Ergänzend dazu sollten Objektkonstruktoren bzw. entsprechende Klassenmethoden aufrufbar sein. Letzteres erfordert ein Klassen- bzw. Metaklassenkonzept, um Klassenmethoden eindeutig von Objektmethoden unterscheiden zu können. Die in einem Prozess verwendeten Objekte sollten bei Bedarf identifizierbar sein, um mehrere Instanzen unterscheiden zu können bzw. Integritätsbedingungen, die die zulässige Anzahl von Instanzen betreffen, ausdrücken zu können. Um eine software-technische Ausnahmebehandlung zu unterstützen, sollten Ausnahmen und auch Ereignissen eine jeweils aufzurufende Objektmethode zuzuordnen sein.

4.6.2. Unterstützung der Entwicklung von Workflow-Management Systemen

Um die Ausführung eines Geschäftsprozesses durch Software zu unterstützen, bieten sich u. a. Systeme an, die den Ablauf eines Prozesses nach Maßgabe der im Prozessmodells spezifizierten Kontrollstrukturen unterstützen. Dabei ist vor allem an Workflow-Management Systeme zu denken. Ein Workflow-Management System unterstützt die Ausführung und Überwachung von Prozessen i. d. R. durch eine Reihe von Maßnahmen. So werden die mit Beginn eines (Teil-) Prozesses erforderlichen Anwendungen an den dafür vorgesehenen Arbeitsplätzen bereitgestellt. Die beteiligten Anwender werden

über Benachrichtigungsdienste oder spezielle Anwendungen von relevanten Ereignissen unterrichtet und ggfs. zu Eingaben aufgefordert. Zudem kann während der Ausführung eines Prozesses der aktuelle Zustand abgefragt werden. Um dem Bedürfnis nach Investitionsschutz entgegenzukommen, sind Workflow-Management Systeme zumeist so konzipiert, dass die im Rahmen eines Geschäftsprozesses eingesetzten Anwendungen weiter verwendet werden können. Das ist allerdings mit einer eingeschränkten Funktionalität solcher Workflow-Management Systeme verbunden, da sich der Zugriff auf die Anwendungen weitgehend auf die Nutzung von Betriebssystemfunktionen wie Starten und ggfs. Terminieren einer Anwendung beschränkt.

Da ein Workflow-Management System eine Vielzahl von Geschäftsprozessstypen unterstützen soll, muss ihm im Einzelfall die Spezifikation eines Geschäftsprozessstyps in geeigneter Form übergeben werden. Um dabei nicht in eine unangenehme Abhängigkeit von einem Anbieter zu gelangen, ist es dabei wünschenswert, dass die entsprechende Spezifikation in einer standardisierten Sprache erstellt werden kann. Auf diese Weise ergibt sich eine deutliche Parallele zu relationalen Datenbanken, deren Schema mittels SQL beschrieben wird, was den Wechsel des Datenbank-Management Systems erheblich erleichtert. Seit einigen Jahren ist die Workflow Management Coalition bestrebt, eine standardisierte Sprache für die Spezifikation von Workflows zu schaffen. Zunächst wurde dazu die Workflow Process Definition Language (WPDL, [Work96]) definiert. Seit einiger Zeit setzt die gleiche Organisation mit Wf-XML auf einen XML-basierten Ansatz ([Work96]). Abbildung 9 veranschaulicht, wie die Spezifikation eines Workflows auf der Grundlage eines Geschäftsprozessmodells erstellt werden kann.

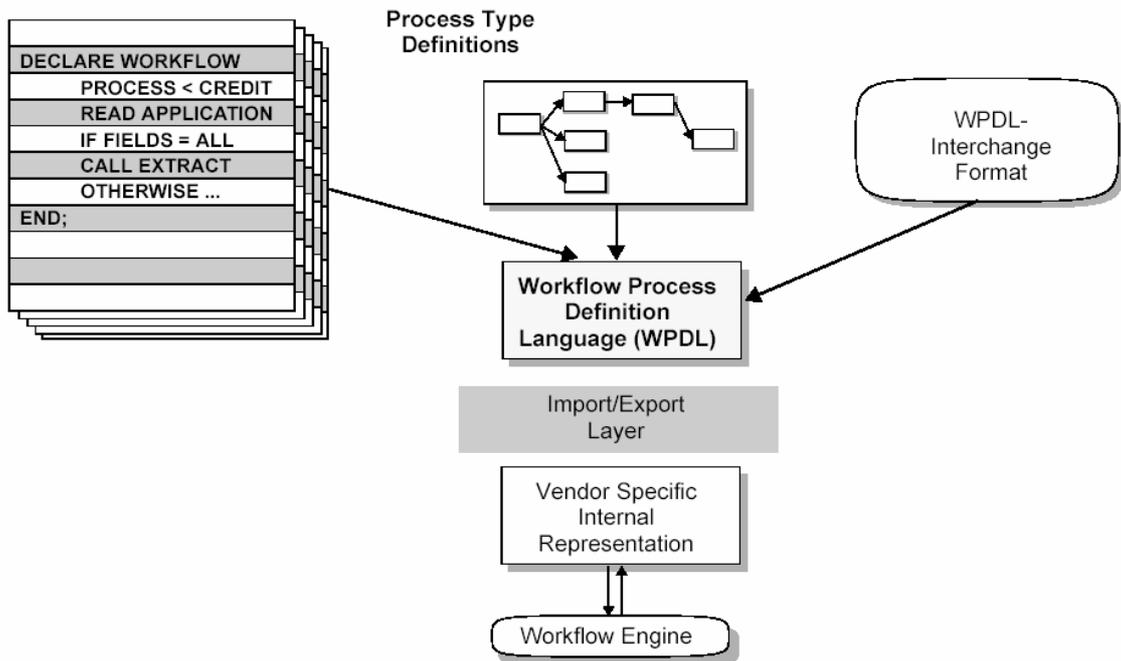


Abbildung 9: Ableitung eines Workflow-Schemas aus einem Geschäftsprozessmodell unter Verwendung einer standardisierten Spezifikationssprache

Um die Ableitung eines Workflow-Schemas aus einem Geschäftsprozessmodell zu unterstützen, sollten möglichst alle Angaben, die für das Schema benötigt werden, mit der Modellierungssprache zu beschreiben sein. Neben den Kontrollstrukturen ist dabei vor allem an die Beschreibung von Anwendungen zu denken. Die Beschreibung von Anwendungen ist bereits im Zusammenhang mit der Ressourcenmodellierung gefordert worden (s. Abschnitt 4.2.4). Im Hinblick auf Workflow-Management Systeme sind ggfs. noch spezifische Angaben zu ergänzen, die sich aus der jeweils verwendeten Spezifikationsprache ergeben. Dabei ist vor allem an die Unterstützung von standardisierten Sprachen zu denken. Das folgende Fragment aus einer Schemabeschreibung veranschaulicht, welche Angaben zu Anwendungen im Schema eines Workflow Management-Systems gemacht werden müssen.

```
Workflow Application List> ::=
    APPLICATION          <generic tool>
    [DESCRIPTION         <description>]
    [TOOLNAME           <tool name>]
```

```
[PARAMETERS      <parameter list>]
[IN_PARAMETERS   <parameter list>]
[OUT_PARAMETERS  <parameter list>]
[<extended attribute list>]
END_APPLICATION
```

Bei den Parametern handelt es sich i. d. R. um Dateien, die als Ressourcen auf einem angemessenen Abstraktionsniveau (also nicht etwa allein durch einen Dateinamen) beschreibbar sein sollten.

4.7. Unterstützung individueller Anpassungen

Selbst eine komplexe Sprache zur Geschäftsprozessmodellierung mag im Einzelfall nicht hinreichen, um zufriedenstellende Modelle zu entwerfen. Für solche Fälle sind Anpassungs- bzw. Erweiterungsmöglichkeiten sinnvoll. Im einfachsten Fall genügt eine Anpassung der Notation (konkrete Syntax), die lediglich die Ersetzung von Symbolen erfordert. Eine solche Ersetzung kann von einem sachkundigen Anwender ohne nennenswerte Probleme vorgenommen werden, ist aber mit dem Aufwand verbunden, geeignete Symbole zu kreieren. Insofern ist es wünschenswert, dass eine Modellierungssprache ergänzend zu den vorgeschlagenen Symbolen eine Reihe domänen- bzw. zweckgebundener Ersatzsymbole bietet. Anpassungen der (abstrakten) Syntax oder der Semantik einer Modellierungssprache sind demgegenüber anspruchsvoller, aber auch fragwürdiger. Letztlich würde dies dazu führen, verschiedene Varianten einer Modellierungssprache anzubieten. Falls es zwischen verschiedenen Einsatzdomänen – etwa der Modellierung von Prozessen im Verwaltungsbereich und der Modellierung von Produktionsprozessen – allzu große Unterschiede gibt, kann dies allerdings eine Option sein.

4.8. Dokumentation

Das Erlernen und Anwenden einer Modellierungssprache erfordert eine angemessene Dokumentation der Sprachkonzepte sowie ihrer Anwendung. Eine umfassende Unter-

stützung der Sprachanwendung wird durch eine die Sprache ergänzende Vorgehensweise, also die Einbettung der Sprache in einer Methode, gewährt (s. Kapitel 5).

Da verschiedene Personengruppen die Werkzeuge aus unterschiedlicher Motivation heraus und mit unterschiedlichen Zielen im Blick verwenden, ist eine Dokumentation, die explizit die verschiedenen Zielgruppen anspricht, positiv zu bewerten (z.B. könnte es für ein Konzept „XY“ die Dokumentationen „XY für Manager“, „XY für Systemanalysten“, „XY für Systementwickler“, „Benutzerhandbuch für XY“ geben).

4.8.1. Motivation und Einsatzzweck: Differenzierung von Benutzergruppen

Mehr noch als andere Sprachen zur konzeptionellen Modellierung richten sich Sprachen zur Modellierung von Geschäftsprozessen an Anwender mit weit streuenden Profilen – im Hinblick auf die jeweiligen Modellierungsziele und die jeweilige fachliche Kompetenz. Hier ist u. a. an Organisatoren, Controller, Vertreter von Fachabteilungen, Führungskräfte, aber auch Systemanalytiker und Software-Entwickler zu denken. Eine gute Dokumentation sollte diesem Umstand Rechnung tragen, indem die Beschreibung der Sprachkonzepte nach wichtigen Anwendergruppen differenziert. Für jede Gruppe sollte zunächst die Motivation der Sprachverwendung erläutert werden: Welche Vorteile sind von Geschäftsprozessmodellen im Hinblick auf anwendergruppenspezifische Ziele zu erwarten. Anschließend sollten die spezifischen Eigenschaften - Besonderheiten, Vorteile, Nachteile - der jeweiligen Sprache erläutert werden.

4.8.2. Sprachbeschreibung

Auch die Sprachbeschreibung sollte für unterschiedliche Anwendergruppen aufbereitet werden. Sie sollte einerseits eine vollständige Liste der verfügbaren Konzepte umfassen, andererseits mittels geeigneter Beispiele ein angemessenes Verständnis der Konzepte fördern. Je nach Umfang der Sprache kann die Liste der Sprachkonzepte differenziert werden in häufig und eher selten benötigte Konzepte. Jedes Konzept sollte in einer strukturierten Weise - z. B.: Bedeutung, grafische Repräsentation, kleines Beispiel – beschrieben werden. Um mit den verwendeten Beispielen eine anschauliche Darstellung

der Sprachkonzepte zu fördern, sollten die Beispiele solche Geschäftsprozesse abbilden, die vielen Anwendern, wenn auch aus unterschiedlichen Perspektiven, vertraut sind – also etwa eine Auftragsbearbeitung. Dabei sollten in Abhängigkeit von typischen Zielen einzelner Anwendergruppen Analyse- und Entwurfsentscheidungen verdeutlicht sowie sinnvolle Auswertungsmöglichkeiten aufgezeigt werden.

4.9. Spezifikation

Für die meisten Benutzergruppen ist eine präzise Spezifikation einer Modellierungssprache nicht hilfreich bzw. sogar verwirrend. Dessen ungeachtet ist eine Spezifikation erforderlich, um Modellierungswerkzeuge entwerfen zu können bzw. um Modelle mathematisch auswerten zu können. Die Spezifikation umfasst die abstrakte Syntax und Semantik sowie die grafische Notation (konkrete Syntax).

Semantik und abstrakte Syntax

Die abstrakte Syntax sollte möglichst formal beschrieben werden, da es sonst nicht immer möglich ist, syntaktisch korrekte von syntaktisch fehlerhaften Modellen eindeutig zu unterscheiden. Im Hinblick auf die weitere Analyse von Modellen bietet auch eine formale Spezifikation der Semantik deutliche Vorteile. Allerdings kann sich eine formale Spezifikation aller Konzepte als zu aufwändig bzw. zu restriktiv erweisen. Es gibt verschiedene Ansätze zur Spezifikation einer Modellierungssprache (vgl. [FrPr97]). In jedem Fall erfordert die Beschreibung einer Sprache die Verwendung einer *Metasprache*. Eine Metasprache kann eingesetzt werden, um eine Grammatik zu beschreiben, die die Regeln zum Erzeugen korrekter Sätze aus einer Menge vorgegebener Symbole festlegt. Die Symbole werden in Terminal- und Nichtterminalsymbole unterteilt. Ein Satz bzw. eine Satzform über einem endlichen Alphabet wird in der Informatik Wort genannt. Auf diese Weise kann eine Sprache auch aufgefasst werden als die Menge aller Wörter von Terminalsymbolen, die durch eine zugehörige Grammatik erzeugt werden. Während dabei nur die Syntax einer Sprache berücksichtigt wird, kann daneben auch die Semantik der einzelnen Wörter und Sätze einer Sprache definiert werden. Hier ist z.B. an axiomatische oder operationale Semantiken zu denken, wie sie mitunter zur

Spezifikation von Programmiersprachen verwendet werden ([Kow96], S. 372 ff.). Alternativ dazu kann eine Metasprache selbst wieder als Modellierungssprache ausgelegt sein. Auf diese Weise kann eine Sprachbeschreibung als Modell dargestellt werden. Man spricht dann von einem *Metamodell*.

Welche Form der Sprachspezifikation zu bevorzugen ist – Metamodell oder Grammatik – ist zum Teil eine Frage subjektiver Präferenzen. Eine Grammatik Vorteile für die automatische Überprüfung der Syntax eines Modells. Auf der anderen Seite spricht für ein Metamodell der Umstand, dass kein Paradigmenwechsel zwischen Objekt- und Metaebene erforderlich ist, was das Verständnis der Sprachbeschreibung fördern dürfte. Im Hinblick auf den Entwurf eines Modellierungswerkzeugs hat ein Metamodell den Vorteil, dass es unmittelbar als Objektmodell rekonstruiert werden kann.

Falls die Modellierungssprache zusätzlich zu den grafischen Bestandteilen auch eine formale Sprache zur Beschreibung von Integritätsbedingungen enthält, ist auch diese zu spezifizieren. Dazu bietet sich eine Grammatik an.

Notation

Die Spezifikation der Notation sollte die vorgeschlagenen grafischen Symbole den jeweiligen Sprachkonzepten präzise zuordnen. Um den angemessenen Einsatz der Sprache in verschiedenen Domänen zu unterstützen, sollten alternative Darstellungsformen für einzelne Symbole angeboten werden. Ergänzend zur Verwendung grafischer Symbole erfordert die Modellierung von Geschäftsprozessen i. d. R. die Zuordnung textueller Bezeichner. Da diese gewissen Konventionen bzw. auch syntaktischen Einschränkungen unterliegen, ist dazu eine formale Spezifikation, etwa unter Rückgriff auf die Backus-Naur Form, angeraten.

5. Einbettung in eine Methode

Eine Modellierungsmethode ist eine Methode, die die Erstellung und Pflege einer Klasse von Modellen unterstützt. Sie umfasst i.d.R. eine oder mehrere Modellierungssprachen, ein Verzeichnis projektspezifischer Rollen und Ressourcen, ein Vorgehensmodell sowie ggfs. Kriterien oder Metriken zur Evaluierung von Modellen und Anwendungsbeispiele. Eine Methode ist also nicht Bestandteil einer Modellierungssprache und stellt deshalb auch kein Bewertungskriterium für Sprachen i. e. S. dar. Dessen ungeachtet wertet die Verfügbarkeit einer geeigneten Methode eine Modellierungssprache u. U. erheblich auf, weshalb wir im folgenden kurz auf die Bestandteile einer Modellierungsmethode eingehen.

5.1. Projektspezifische Rollen und Ressourcen

Das Management von Modellierungsprojekten erfordert organisatorische Vorbereitungen. Dazu gehört die Auswahl von Mitarbeitern und sonstiger Ressourcen. Zur Unterstützung der Projektverantwortlichen sollte ein Methode kommentierte Verzeichnisse mit Rollen und damit verbundenen Anforderungsprofilen bereitstellen. Dazu gehören auch Vorschläge für die Gestaltung von Kommunikationsbeziehungen, etwa durch den Einsatz von IuK-Technologien bzw. durch die Durchführung gemeinsamer Sitzungen. Darüber hinaus sollten die mit den Rollen verbundenen Aufgaben sowie Koordinations- und Qualitätssicherungsmaßnahmen dargestellt werden. Für die Ausstattung - und betriebswirtschaftliche Kalkulation – von Projekten sind zudem die benötigten Ressourcen zu erfassen und zu bewerten. Unterstützung dafür bietet ein Verzeichnis wichtiger Ressourcentypen (Kommunikationsmedien, Rechner, Software, etc.), das durch Erläuterungen der jeweiligen Einsatzziele und –bedingungen sowie wesentlicher Auswahlkriterien ergänzt ist.

5.2. Vorgehensmodell

Ein Vorgehensmodell dient dazu, die gesamte mit der Durchführung eines Projekts verbundene Aufgabe in überschaubare Teilaufgaben bzw. Phasen zu gliedern und diese in eine kausale bzw. temporale Ordnung zu bringen. Für jede Teilaufgabe sollten auf einem angemessenen Abstraktionsniveau typische Ziele, kritische Erfolgsfaktoren, die involvierten Rollen und ihre Aufgaben, die benötigten Informationen und Kommunikationsbeziehungen sowie die anzustrebenden Ergebnisse beschrieben werden. Ergänzend dazu sind spezifische Heuristiken, Techniken und Qualitätskriterien anzugeben. Angesichts der vielfältigen Zielsetzungen, die mit der Erstellung von Geschäftsprozessmodellen verbunden sind, ist ein allgemeines Vorgehensmodell allenfalls auf einem sehr hohen, im Einzelfall nur begrenzt hilfreichen Abstraktionsniveau denkbar. Deshalb sollte ein einschlägiges Vorgehensmodell in Abhängigkeit von typischen Zielen und Randbedingungen der Modellierung verschiedene Varianten beinhalten. Um die Vorgehensweise zu veranschaulichen bieten sich neben Beispiele vor allem Fallstudien an, in denen die verschiedenen Aspekte der Modellierung von Geschäftsprozessen dargestellt werden.

6. Abschließende Bemerkungen

Der vorliegende Arbeitsbericht bietet eine detaillierte und umfassende Darstellung allgemeiner Anforderungen, die mit der Modellierung von Geschäftsprozessen verbunden sind. Dabei wurde von den Besonderheiten spezieller Geschäftsprozessstypen, wie sie etwa in der industriellen Produktion, der Logistik oder im Bereich der öffentlichen Verwaltung zu verzeichnen sind, abstrahiert. Die Untersuchung der mit der Modellierung von Geschäftsprozessen verbundenen Ziele und Randbedingungen war nicht nur darauf gerichtet, klar definierte Anforderungen an einschlägige Modellierungssprachen zu formulieren, sondern auch darauf, die Beziehungen zwischen verschiedenen Anforderungen sowie die mit ihrer Umsetzung teilweise verbundenen erheblichen wissenschaftlichen Herausforderungen zu erfassen. Hier ist beispielsweise an die Vielzahl von Abstraktionsebenen zu denken, die bei der Modellierung zu berücksichtigen sind, und die gleichzeitig mit erheblichen Schwierigkeiten beim Entwurf von Modellierungssprachen verbunden sind. Da die Diskussion der Anforderungen auch bewusst akademische Probleme einschließt, ist der Arbeitsbericht nur bedingt geeignet, Entscheidungsträgern in der Praxis eine Unterstützung bei der Auswahl von Modellierungssprachen zu geben. Deshalb werden die Anforderungen in einem weiteren Arbeitsbericht [FrLa03] aufbereitet und in Form eines praxisgerechten Bezugsrahmens präsentiert. Der vorliegende Arbeitsbericht bietet zudem die Grundlage für die gegenwärtig laufende (Re-) Spezifikation der MEMO Organisation Modelling Language (MEMO-OrgML), die in Kürze ebenfalls als Arbeitsbericht veröffentlicht wird.

Literaturverzeichnis

- [AaBa97] van der Aalst, W.M.P; Basten, T.: Lifecycle Inheritance – A Petri-Net-based approach. In: Azema, P.; Balbo, G. (Hrsg): Application and Theory of Petri Nets 1997, Lecture Notes in Computer Science, Band 1248, Springer, Berlin 1997.
- [BSc02] Bea, F. X.; Schweitzer, M.: Einleitung: Leistungsprozess. In: (Hg.): Allgemeine Betriebswirtschaftslehre: Bd. 3: Leistungsprozess. 8. Neubearb. und erw., Stuttgart: Lucius & Lucius 2002, S. 1-7.
- [Bern97] Berndt, R. (Hg.): Business Reengineering: Effizientes Neugestalten von Geschäftsprozessen. Berlin, Heidelberg, New York, et al.: Springer 1997
- [BeRo96] Becker, J; Rosemann, M. (Hg.): Workflow-Management – State of the Art aus Sicht von Theorie und Praxis. Nr. 47 aus der Reihe Arbeitsberichte des Instituts für Wirtschaftsinformatik, Universität Münster, 1996.
- [Biwe03] Biwer, M.: Eine Methode zur Erstellung von Business Continuity Plans. Diplomarbeit am Institut für Wirtschaftsinformatik der Universität Koblenz. Erscheint 2003.
- [BILü02] Bloech, J.; Lücke, W.: Produktionswirtschaft. In: Allgemeine Betriebswirtschaftslehre: Bd. 3: Leistungsprozess. 8. Neubearb. und erw., Stuttgart: Lucius & Lucius 2002, S. 1-212.
- [BKN00] Bustard, D.; Kawalek, P.; Norris, M. (Hg.): Systems Modeling for Business Process Improvement. Boston, et al.: Artech House 2000
- [CoKa91] Cooper, R.; Kaplan, R.: Profit priorities from activity-based costing. In: Harvard Business Review, 0. Jg., Nr. 3, 1991, S. 130-137
- [Fran94] Frank, U.: Multiperspektivische Unternehmensmodellierung -- Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg Verlag, 1994.
- [Fran98] Frank, U.: Evaluating Modelling Languages: Relevant Issues, Epistemological Challenges and a Preliminary Research Framework. Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 15, Koblenz 1998.
- [Fran00] Frank, U.: Vergleichende Betrachtung von Standardisierungsvorhaben zur Realisierung von Infrastrukturen für das E-Business. Arbeitsberichte

des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau.
Nr. 22, Universität Koblenz-Landau 2000

- [Fran01] Frank, U.: Organising the Corporation: Research Perspectives, Concepts and Diagrams. Arbeitsberichte des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau. Nr. 25, Universität Koblenz-Landau 2001.
- [Fran02] Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), Honolulu 2002.
- [FrLa02] Frank, U.; van Laak, B. L.: A Method for the multi-perspective Design of versatile E-Business Systems. In: Proceedings of the Americas Conference on Information Systems (AMCIS) 2002, Dallas 2002.
- [FrLa03] Frank, U.; van Laak, B. L.: Ein Bezugsrahmen zur Evaluation von Werkzeugen zur Unternehmensmodellierung. Wird erscheinen in der Reihe Arbeitsberichte des Instituts für Wirtschaftsinformatik, Koblenz 2003.
- [FrPr97] Frank, U.; Prasse, M.: Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen - veranschaulicht am Beispiel von OML und UML. Arbeitsberichte des Institut für Wirtschaftsinformatik der Universität Koblenz-Landau, Nr. 7, 1997
- [FrSc01a] Frank, U.; Schauer, H.: Potentiale und Herausforderungen des Wissensmanagements aus der Sicht der Wirtschaftsinformatik. In: Schreyögg, G. (Hg.): Wissen in Unternehmen: Konzepte - Maßnahmen - Methoden. Berlin: Erich Schmidt 2001, S. 163-182
- [FrSc01b] Frank, U.; Schauer, H.: Software für das Wissensmanagement: Einschlägige Systeme und deren Einführung. In: Das Wirtschaftsstudium (wisu), 30. Jg., Nr. 5, 2001, S. 718-726
- [Gait83] Gaitanides, M.: Prozeßorganisation. München: Vahlen 1983
- [GeGa99] Gehring, H.; Gadatsch, A.: Ein Rahmenkonzept für die Modellierung von Geschäftsprozessen und Workflows. Fachbereichsbericht Nr. 274 aus der Reihe Diskussionsbeiträge der Fachbereichs Wirtschaftswissenschaft, Fernuniversität Hagen, September 1999.
- [Glas94] Glasson, B. C. (Hg.): Business process re-engineering: information systems opportunities and challenges: Proceedings of the IFIP TC8 Open

- Conference, Queensland Gold Coast. Amsterdam, et al.: North Holland 1994
- [Groc82] Grochla, E.: Grundlagen der organisatorischen Gestaltung. Stuttgart 1982.
- [Hamm96] Hammer, M.: Beyond Reengineering: How the Process-Centred Organization is Changing our Work and our Lives. Harper Collins 1996
- [HaCh93] Hammer, M.; Champy, J.: Reengineering The Corporation: A Manifesto for Business Revolution. New York: Harper Business 1993
- [Harr91] Harrington, J. H.: Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity and Competitiveness. McGraw-Hill 1991
- [Hein77] Heinen, E.: Einführung in die Betriebswirtschaftslehre. 6. verbesserte und erw., Wiesbaden: Gabler 1977
- [HeSc+01] Herrmann, T.; Scheer, A.; Weber, H. (Hg.): Verbesserung von Geschäftsprozessen mit flexiblen Workflow-Management-Systemen 4: Workflow-Management für die lernende Organisation. Einführung, Evaluierung und zukünftige Perspektiven. Heidelberg, et al.: Physica-Verlag 2001
- [Hoch96] Hoch, T.: Einsatz der Informationsverarbeitung bei Business Process Reengineering. DUV 1996
- [HoKi+93] Horvath, P.; Kieninger, M.; Mayer, R.; Schimank, C.: Prozeßkostenrechnung - oder wie die Praxis die Theorie überholt - Kritik und Gegenkritik. In: DBW, 53. Jg., Nr. 5, 1993, S. 609-628
- [Jaco94] Jacobson, I.: The Object Advantage - Business Process Reengineering with Object Technology. Addison-Wesley 1994
- [JoMc+94] Johansson, H. J.; McHugh, P.; Pendlebury, A. J.; Wheeler III, W. A.: Business Process Reengineering: BreakPoint Strategies for Market Dominance. Wiley 1994
- [Kamb80] Kambartel, F.: Bedingung. In: Mittelstraß, J. (Hg.): Enzyklopädie Philosophie und Wissenschaftstheorie. Mannheim: B I Wissenschaftsverlag (Bibliographisches Institut) 1980

- [Kow96] Kowalk, W. P.: System – Modell – Programm. Heidelberg, Berlin, Oxford: Spektrum 1996
- [Kube95] Kubeck, L. C.: Techniques for Business Process Redesign: Tying It All Together. Wiley 1995
- [LiSu89] Liebelt, W.; Sulzberger, M.: Grundlagen der Ablauforganisation. Gießen: Dr. Götz Schmidt 1989
- [Lore80] Lorenz, K.: Ereignis. In: Mittelstraß, J. (Hg.): Enzyklopädie Philosophie und Wissenschaftstheorie. Mannheim: B I Wissenschaftsverlag (Bibliographisches Institut) 1980
- [Lore96] Lorenz, K.: Vorgang. In: Mittelstraß, J. (Hg.): Enzyklopädie Philosophie und Wissenschaftstheorie. Mannheim: B I Wissenschaftsverlag (Bibliographisches Institut) 1996, S. 564-564
- [MaCr+99] Malone, T.W.; Crowston, K.; Lee, J.; Pentland, B.; Dellarocas, C.; Wyner, G.; Quimby, J.; Osborne, C.S.; Bernstein, A.; Herman, G.; Klein, M.; Donnel, E.: Toward a handbook of organizational processes. In: Management Science 45(3), March 1999, S. 425-443.
- [Meye97] Meyer, B.: Object Oriented Software Construction. 2. Ed., Prentice Hall 1997
- [Möbi99] Möbius, D.: Geschäftsprozessoptimierung durch Workflow-Management: Grundlagen, Fallbeispiel, Umsetzung. Bankakademie-Vlg. 1999
- [Mühl96] zur Mühlen, M.: Der Lösungsbeitrag von Metamodellen und Kontrollflussprimitiven beim Vergleich von Workflowmanagementsystemen. Diplomarbeit, Universität Münster, 1996.
- [Ober96a] Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen. Teubner 1996
- [Ober96b] Oberweis, A.: An Integrated Approach for the Specification of Processes and Related Complex Structured Objects in Business Applications. In: Decision Support Systems, 17.Jg., Nr. 1, 1996, S. 31-53
- [OMG99] OMG (Hrsg.): OMG Unified Modeling Language Specification, Version 1.3, 1999.

- [OSS94] Oberweis, A.; Scherrer, G.; Stucky, W.: INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. In: Information Systems, 19. Jg., Nr. 8, 1994, S. 643-660
- [Ould95] Ould, M. A.: Business Processes: Modelling and Analysis for Re-Engineering and Improvement. Wiley 1995
- [PrRa+97] Probst, G.; Raub, S. P.; Romhardt, K.: Wissen Managen. Wie Unternehmen ihre wertvollsten Ressourcen optimal nutzen. Frankfurter Allgemeine Zeitung, Frankfurt/Main 1997.
- [Rum96] Rumbaugh, J.: Notation notes: Principles for choosing notation. In: Journal of Object-Oriented Programming. Mai 1996, S. 11-14
- [Schr98] Schreyögg, G.: Organisation. 2. überarb. Auflage, Gabler Verlag, Wiesbaden 1998.
- [Schw02] Schweitzer, M.: Innovationsmanagement. In: (Hg.): Allgemeine Betriebswirtschaftslehre: Bd. 3: Leistungsprozess. 8. neubearb. und erw., Stuttgart: Lucius & Lucius 2002, S. 9-76.
- [ScFr01] Schauer, H.; Frank, U.: Software für das Wissensmanagement: Einschlägige Systeme und deren Einführung. In: WISU – das Wirtschaftsstudium, Heft 5 / 2001, S. 718-726
- [SüEb97] Süttenbach, R.; Ebert, J.: A Booch Metamodel. Fachberichte Informatik 5/97, Universität Koblenz-Landau, 1997.
- [Troß02] Troßmann, E.: Beschaffung und Logistik. In: (Hg.): Allgemeine Betriebswirtschaftslehre: Bd. 3: Leistungsprozess. 8. neubearb. und erw., Stuttgart: Lucius & Lucius 2002, S. 77-144.
- [Wri74] von Wright, G. H.: Erklären und Verstehen. Frankfurt/M.: Athenäum 1974
- [Webe00] Weber, J.: Stichworte: Balanced Scorecard, Controlling, Prozesskostenrechnung. In: (Hg.): Gabler's Wirtschaftslexikon. 15. Aufl., Wiesbaden: Gabler 2000
- [Work02] Workflow Management Coalition: Workflow Standard-Interoperability Wf-XML Binding: WfMC-TC-1023, Version 1.1
- [Work96] Workflow Management Coalition: Workflow Standard - Interoperability Abstract Specification: WfMC-TC-1012, Version 1.0

[WyLe02] Wyner, G.; Lee, J.: Defining Specialization for Process Models. CSS No. 216, Sloan No. 4159, 2002.

Bisherige Arbeitsberichte

- Hampe, J. F.; Lehmann, S.: Konzeption eines erweiterten, integrativen Telekommunikationsdienstes. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 1**, Koblenz 1996
- Frank, U.; Halter, S.: Enhancing Object-Oriented Software Development with Delegation. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 2**, Koblenz 1997
- Frank, U.: Towards a Standardization of Object-Oriented Modelling Languages? Arbeitsbericht des Instituts für Wirtschaftsinformatik, **Nr. 3**, Koblenz 1997
- Frank, U.: Enriching Object-Oriented Methods with Domain Specific Knowledge: Outline of a Method for Enterprise Modelling. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 4**, Koblenz 1997
- Prasse, M.; Rittgen, P.: Bemerkungen zu Peter Wegners Ausführungen über Interaktion und Berechenbarkeit, Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 5**, Koblenz 1997
- Frank, U.; Prasse, M.: Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen - veranschaulicht am Beispiel vom OML und UML. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 6**, Koblenz 1997
- Klein, S.; Zickhardt, J.: Auktionen auf dem World Wide Web: Bezugsrahmen, Fallbeispiele und annotierte Linksammlung. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 7**, Koblenz 1997
- Prasse, M.; Rittgen, P.: Why Church's Thesis still holds - Some Notes on Peter Wegner's Tracts on Interaction and Computability. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 8**, Koblenz 1997
- Frank, U.: The MEMO Meta-Metamodel, Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 9**, Koblenz 1998
- Frank, U.: The Memo Object Modelling Language (MEMO-OML), Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 10**, Koblenz 1998
- Frank, U.: Applying the MEMO-OML: Guidelines and Examples. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 11**, Koblenz 1998
- Glabbeek, R.J. van; Rittgen, P.: Scheduling Algebra. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 12**, Koblenz 1998

- Klein, S.; Güler, S.; Tempelhoff, S.: Verteilte Entscheidungen im Rahmen eines Unternehmensplanspiels mit Videokonferenzunterstützung, Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 13**, Koblenz 1997
- Frank, U.: Reflections on the Core of the Information Systems Discipline. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 14**, Koblenz 1998
- Frank, U.: Evaluating Modelling Languages: Relevant Issues, Epistemological Challenges and a Preliminary Research Framework. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 15**, Koblenz 1998
- Frank, U.: An Object-Oriented Architecture for Knowledge Management Systems. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 16**, Koblenz
- Rittgen, P.: Vom Prozessmodell zum elektronischen Geschäftsprozess. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 17**, Koblenz 1999
- Frank, U.: Memo: Visual Languages for Enterprise Modelling. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 18**, Koblenz 1999
- Rittgen, P.: Modified EPCs and their Formal Semantics. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 19**, Koblenz 1999
- Prasse, M., Rittgen, P.: Success Factors and Future Challenges for the Development of Object Orientation. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 20**, Koblenz 2000
- Schönert, S.: Virtuelle Projektteams - Ein Ansatz zur Unterstützung der Kommunikationsprozesse im Rahmen standortverteilter Projektarbeit. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 21**, Koblenz 2000
- Frank, U.: Vergleichende Betrachtung von Standardisierungsvorhaben zur Realisierung von Infrastrukturen für das E-Business. . Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 22**, Koblenz 2000
- Jung, J.; Hampe, J.F.: Konzeption einer Architektur für ein Flottenmanagementsystem. . Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 23**, Koblenz 2001
- Jung, J.: Konzepte objektorientierter Datenbanken – Konkretisiert am Beispiel GemStone. . Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 24**, Koblenz 2001
- Frank, U.: Organising the Corporation: Research Perspectives, Concepts and Diagrams. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 25**, Koblenz 2001

- Kirchner, L.; Jung, J.: Ein Bezugsrahmen zur Evaluierung von UML-Modellierungswerkzeugen. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 26**, Koblenz 2001
- Botterweck, G.; Hampe, J.: Benutzeroberflächen für WAP-basierte Mobile Commerce Anwendungen. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 27**, Koblenz 2001
- Jung, J.; van Laak, Bodo L.: Flottenmanagementsysteme - Grundlegende Technologien, Funktionen und Marktüberblick. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 28**, Koblenz 2001
- Jung, J.; Kirchner, L.: Logistische Prozesse im Handwerk – Begriffliche Grundlagen und Referenzmodelle. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 29**, Koblenz 2001
- Frank, U.: Forschung in der Wirtschaftsinformatik: Profilierung durch Kontemplation – ein Plädoyer für den Elfenbeinturm. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 30**, Koblenz 2002
- Frank, U.; van Laak, B.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 31**, Koblenz 2002
- Jung, U.: Entwicklung eines elektronischen Fahrtenbuchs – Grundlegender Entwurf, prototypische Implementierung und zukünftige Potentiale. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 32**, Koblenz 2002
- Van Laak, B.; Frank, U.: Eine Struktur zur Beschreibung von Prozessmustern der ECOMOD-Prozessbibliothek. Arbeitsberichte des Instituts für Wirtschaftsinformatik, **Nr. 33**, Koblenz 2002.