

# Evolution von Organisationsmodellen in Workflow-Management-Systemen

Diplomarbeit

*Fakultät für Informatik  
Abteilung Datenbanken und Informationssysteme  
Universität Ulm*

*vorgelegt von:* Ursula Wiedemuth-Catrinescu  
*Gutachter:* Prof. Dr. Peter Dadam  
Dr. Manfred Reichert

*Juli 2002*

# Inhalt

<b>Inhalt.....</b>	<b>2</b>
<b>Zusammenfassung .....</b>	<b>4</b>
<b>1 Einleitung.....</b>	<b>5</b>
1.1 Problemstellung .....	6
1.2 Ziele und Gliederung der Arbeit .....	7
<b>2 Grundlagen der Organisations-Modellierung .....</b>	<b>9</b>
2.1 Organisations-Metamodell .....	9
2.1.1 Allgemeingültiges Organisations-Metamodell .....	9
2.1.2 Konfiguriertes Organisations-Metamodell .....	12
2.2 Organisationsmodell .....	14
2.2.1 Formale Darstellung .....	14
2.2.2 Beispiel für ein Organisationsmodell .....	17
2.3 Zusammenfassung .....	19
<b>3 Organisationsmodell-Änderungen.....</b>	<b>20</b>
3.1 Anwendungsszenarien.....	20
3.2 Klassifizierung von Organisationsmodell-Änderungen .....	22
3.3 Zusammenfassung .....	24
<b>4 Operationen für Organisationsmodell-Änderungen.....</b>	<b>25</b>
4.1 Anforderungen an Änderungsoperationen.....	25
4.2 Herleitung der Änderungsoperationen.....	26
4.3 Änderungsprimitive .....	28
4.4 Elementare Änderungsoperationen.....	30
4.4.1 Entitäten.....	31
4.4.2 Relationen.....	32
4.4.3 Attribute .....	34
4.5 Komplexe Änderungen und Änderungstransaktionen.....	37
4.5.1 Änderungstransaktionen .....	37
4.5.2 Konsistenzsicherung einer Änderungstransaktion.....	38
4.5.3 Vordefinierte komplexe Änderungen .....	43
4.6 Zusammenfassung .....	50
<b>5 Pflege der Bearbeiterformeln (statischer Fall).....</b>	<b>51</b>
5.1 Grundprinzipien der Bearbeiterformeln .....	51
5.2 Problemstellung .....	53
5.2.1 Bearbeiterformeln .....	53
5.2.2 Bearbeitermengen.....	55
5.3 Lösungsansätze zur Behandlung von Referenzierungs-Problemen.....	56

5.3.1	Welche Bearbeiterzuordnungen müssen bei welchen Änderungsoperationen angepaßt werden? .....	57
5.3.2	Wie können Bearbeiterzuordnungen angepaßt werden? .....	59
5.3.3	Wer führt die Anpassung durch? .....	62
5.3.4	Wann wird die Anpassung durchgeführt? .....	63
5.4	Zusammenfassung .....	64
<b>6</b>	<b>Pflege der Arbeitslisten (dynamischer Fall) .....</b>	<b>65</b>
6.1	Grundlagen .....	65
6.1.1	Arbeitslisten .....	65
6.1.2	Serverseitige Verwaltung der Arbeitslisten .....	66
6.1.3	Klientenseitige Verwaltung der Arbeitslisten .....	68
6.2	Problemstellung .....	68
6.3	Aktualisierung der Arbeitslistenverwaltung .....	72
6.3.1	Aktualisierungsvarianten .....	72
6.3.2	Aktualisierung der Bearbeiterformeln .....	74
6.3.3	Aktualisierung der Workitem- Zuordnung .....	75
6.4	Identifikation kritischer Organisationsmodell- Änderungen .....	77
6.5	Beispiel für organisatorische Änderungen zur Laufzeit .....	80
6.6	Zusammenfassung .....	84
<b>7</b>	<b>Diskussion verwandter Ansätze und Themen .....</b>	<b>85</b>
7.1	Organisationsmodellierung .....	85
7.2	Organisatorische Änderungen zur Laufzeit in existierenden Workflow- Management-Systemen .....	86
7.2.1	Staffware 2000 .....	87
7.2.2	MQSeries Workflow .....	88
7.2.3	ADEPT .....	88
7.3	Verwandte und weiterführende Themen .....	89
7.3.1	Modellierung von Ressourcen .....	89
7.3.2	Autorisierungskonzepte in Workflow-Management-Systemen .....	89
7.3.3	Schemaevolution in Workflow-Management-Systemen .....	90
<b>8</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>92</b>
	<b>Quellenverzeichnis .....</b>	<b>94</b>
	<b>Abbildungsverzeichnis .....</b>	<b>97</b>
	<b>Tabellenverzeichnis .....</b>	<b>98</b>
	<b>Abkürzungsverzeichnis .....</b>	<b>99</b>
	<b>Anhang .....</b>	<b>100</b>

## Zusammenfassung

Workflow-Management-Systeme unterstützen die Modellierung, Analyse und Steuerung von Geschäftsprozessen in Organisationen. Dabei stand bisher eine flexible Gestaltung von Prozeßabläufen im Mittelpunkt der Forschung. Wenig Beachtung wurde dagegen den Änderungen der zugrundeliegenden Organisation und den dadurch notwendigen Anpassungen des Organisationsmodells geschenkt.

Da zwischen dem Organisationsmodell und anderen Komponenten eines Workflow-Management-Systems zahlreiche Cross-Referenzen bestehen, können bei Änderungen des Organisationsmodells diese Referenzen oder die daraus abgeleiteten Datenstrukturen verwaist bzw. nicht mehr aktuell sein. Von solchen Änderungen kann insbesondere die Zuordnung von Workflow-Aktivitäten zu Bearbeitern betroffen sein. In der Folge werden diese Aktivitäten möglicherweise falsch oder gar nicht mehr zugewiesen. Dadurch können Sicherheitsbestimmungen verletzt werden oder sogar der gesamte Workflow ins Stocken geraten.

In dieser Arbeit wird ein umfangreiches Konzept zur Modellierung von Organisationen vorgestellt. Es wird eine vollständige Menge von elementaren und komplexen Änderungsoperationen beschrieben, die sich durch eine präzise Semantik auszeichnen und sämtliche Änderungen des Organisationsmodells unter Beachtung von Konsistenz- und Korrektheitseigenschaften erlauben.

Die Problematik der Cross-Referenzen zwischen dem (geänderten) Organisationsmodell und anderen Komponenten des Workflow-Management-Systems wird sowohl für den statischen als auch für den dynamischen Fall diskutiert.

Im statischen Fall, in dem noch keine Workflow-Instanzen berücksichtigt werden, stehen Bearbeiterformeln, die das Organisationsmodell referenzieren, im Mittelpunkt. Je nach Semantik der Änderung werden Lösungsansätze zur automatischen, semiautomatischen oder manuellen Anpassung der veralteten Bearbeiterformeln angeboten. Es wird diskutiert, welche Bearbeiterformeln angepaßt werden müssen, wann und durch wen dies geschehen kann.

Im dynamischen Fall wird betrachtet, welche Auswirkungen organisatorische Änderungen auf laufende Workflow-Instanzen haben. Die Möglichkeiten zur Aktualisierung von Datenstrukturen der Arbeitslistenverwaltung des Workflow-Servers und von klientenseitigen Arbeitslisten werden gegenübergestellt und bewertet.

Abschließend werden verwandte Ansätze und Themen der Forschung diskutiert. Es erfolgt eine Abgrenzung der vorliegenden Arbeit sowie ein Ausblick auf weiterführende Themen.

# 1 Einleitung

Workflow-Management-Systeme werden zur Modellierung, Analyse und Steuerung von Geschäftsprozessen eingesetzt (Leymann & Roller, 2000; Jablonski, Böhm & Schulze, 1999).

In den letzten Jahren hat die Workflow-Technologie stark an Popularität gewonnen. Gerade in der heutigen Zeit, die von Firmenfusionen und Globalisierung geprägt ist, sind komplexe Arbeitsabläufe mit sehr vielen Beteiligten an verschiedenen Standorten keine Seltenheit. Der Einzelne hat kaum eine Chance, den vollständigen Produktions- oder Geschäftsprozeß zu überblicken. Der Einsatz von Workflow-Management-Systemen kann den Unternehmen zu gut strukturierten, effizienten und korrekten Arbeitsabläufen verhelfen und ihnen damit einen echten Wettbewerbsvorteil verschaffen.

Auf dem kommerziellen Sektor sind inzwischen zahlreiche Systeme verschiedener Anbieter verfügbar, wie MQSeries Workflow (IBM), Staffware 2000 (Staffware plc.), Ultimus (Ultimus GmbH) oder Oracle 11i (Oracle Corporation).

Auch in der Grundlagenforschung ist das Thema Workflow-Management sehr aktuell (Bertino, Ferrari & Atluri, 1999; Martschat, 2001; Sparr, 2001; van der Aalst & Jablonski, 2000). Ein besonderer Schwerpunkt liegt hier in der Entwicklung flexibler Workflow-Management-Systeme, die dynamische Ablaufänderungen im laufenden Betrieb gestatten (Reichert, 2000; Rinderle, Reichert & Dadam, 2002).

Wenig Beachtung wurde dagegen bisher Änderungen der zugrundeliegenden Organisationsstrukturen des Workflows geschenkt. Gerade aber in Organisationen finden häufige Veränderungen statt, etwa durch Personalwechsel, Umstrukturierungen oder Fusionen.

Heutige Workflow-Management-Systeme fangen dieses Problem häufig mit Hilfe von *rollenbasierten* Ansätzen ab. Dabei wird eine auszuführende Aktivität einer Rolle und nicht einem festen Bearbeiter zugewiesen (NIST<sup>1</sup>; Ferraiolo & Kuhn, 1992). Das schränkt einerseits den Spielraum zur Formulierung der Bearbeiterzuordnung stark ein, andererseits können Organisationsänderungen erheblich komplexer sein, als daß nur einzelne Mitarbeiter wechseln. Beispielsweise gehören umfangreiche Personalrotationen zum Alltag von Universitäts-Kliniken, es werden Forschungsgruppen geschaffen oder aufgelöst, oder auch ganze Kliniken zusammengelegt. Dafür ist der bisherige, rollenbasierte Ansatz nicht ausreichend.

In dieser Arbeit wird ein generischer Ansatz zum Umgang mit Organisationsmodell-Änderungen entwickelt, der Methoden zur Modifikation von Organisationsmodellen sowie zur Anpassung der davon abhängenden Bearbeiterzuordnungen und Arbeitslisteneinträge anbietet.

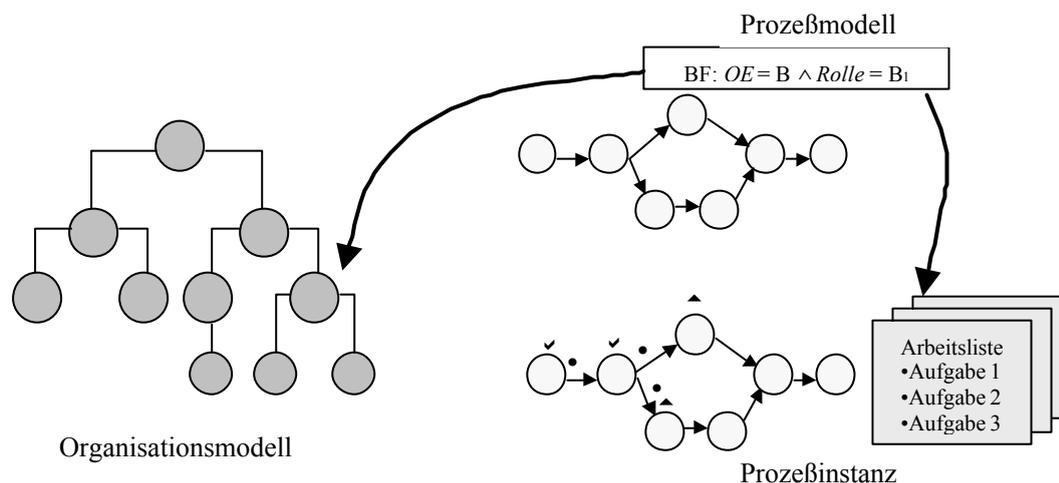
---

<sup>1</sup> National Institute of Standards and Technology

## 1.1 Problemstellung

Workflow-Management-Systeme benötigen zur Steuerung von Prozessen Informationen über die Organisations- und Personalstruktur des Unternehmens. Nur diese Informationen machen es möglich, einzelnen Aktivitäten des Prozesses entsprechende Bearbeitermengen zuzuweisen.

Die Organisations- und Personalstruktur wird normalerweise in einem separaten Organisationsmodell abgebildet. Für das resultierende Organisationsmodell kommen - neben der Definition von Zugriffs- und Ausführungsrechten in Workflow-Management-Systemen - noch weitere Anwendungsbereiche, wie Personalplanung, Zugriffsrechtverwaltung in Dokumenten-Management-Systemen oder Ressourcenverwaltung in Frage. Idealerweise sollte deshalb ein Organisationsmodell vorliegen, auf das verschiedene Werkzeuge und Anwendungen Bezug nehmen können. In der Praxis verfügen viele Systeme jedoch über ihr eigenes Organisationsmodell, so daß die entsprechenden Daten separat, häufig auch redundant gepflegt werden müssen (Jablonski, Schlundt & Wedekind, 2001).



**Abbildung 1** Zusammenhang zwischen Organisationsmodell, Prozeßmodell und Prozeßinstanzen

Im Workflow-Management-System existieren diverse Referenzen zwischen dem Organisationsmodell, dem Prozeßmodell sowie den Arbeitslisten der Benutzer (Abbildung 1). Bei der Modellierung des Prozeßmodells kann für jede Aktivität über eine Bearbeiterformel angegeben werden, wie die Menge der potentiellen Bearbeiter dieser Aktivität zur Laufzeit berechnet werden soll. Bei der Auflösung dieser Formeln zur Laufzeit werden die (logischen) Ausdrücke über das Organisationsmodell aufgelöst und die Bearbeitermenge berechnet. Die entsprechenden potentiellen Bearbeiter bekommen dann den Auftrag für diese Aktivität in ihre Arbeitsliste geschrieben.

Was passiert nun, wenn die reale Organisation sich ändert? Es ergeben sich folgende wichtige Fragestellungen:

1. Zunächst muß das Organisationsmodell aktualisiert werden. Dafür werden entsprechende Änderungsoperationen benötigt. Aber ist das Organisationsmodell nach einer Änderung noch konsistent und semantisch korrekt?
2. Was geschieht mit den Bearbeiterformeln der Prozeßaktivitäten, wenn das referenzierte Organisationsmodell geändert wurde? Welche Formeln sind davon betroffen und was referenzieren sie jetzt? Wenn beispielsweise eine Stelle in der Organisation eingespart wurde, kann es vorkommen, daß diese noch von diversen Bearbeiterformeln referenziert wird. Dann ist eventuell nicht mehr sichergestellt, wem die Aktivität im laufenden Betrieb zugeteilt wird oder auch was mit den Zugriffsrechten passiert.
3. Zur Laufzeit verschärft sich dieses Problem noch. Jetzt sind nicht nur die jeweiligen Prozeßmodelle sondern auch sämtliche Instanzen von ihnen betroffen. Bei komplexen Änderungen können das sehr viele Instanzen sein. Es können veraltete Arbeitslisten auftreten, weil - anders als beim klassischen rollenbasierten Zugriff - die Arbeitslisten schon bei Aktivierung des Prozeßschrittes geschrieben werden und nicht erst beim Zugriff. Dennoch muß eine Änderung am Organisationsmodell auch zur Laufzeit möglich sein, da aus wirtschaftlichen und sonstigen Gründen der Betrieb des laufenden Workflows meistens nicht unterbrochen werden kann.

Aus diesen Problemen, die bei der Anpassung von Organisationsänderungen in den Modellen von Workflow-Management-Systemen entstehen, leiten sich die Ziele der Diplomarbeit ab.

## **1.2 Ziele und Gliederung der Arbeit**

Es sollen in dieser Arbeit Strategien zum Umgang mit Organisationsänderungen in Workflow-Management-Systemen entwickelt werden. Dabei müssen sowohl der statische Fall (Entwicklungszeit) als auch der dynamische Fall (Laufzeit) betrachtet werden. Eine wichtige Rolle spielen hierbei die Referenzen zwischen dem Organisationsmodell und Bearbeiterformeln, die zur Zuweisung von Zugriffs- und Ausführungsrechten genutzt werden.

Zunächst werden in Abschnitt 2 die wesentlichen Grundlagen zur Modellierung von Organisationen gegeben. Es wird gezeigt, wie sich Organisationsmodelle und Organisations-Metamodelle mit Hilfe von Mengen sinnvoll darstellen lassen. Dieses Mengenkonzept erleichtert später die korrekte Formulierung von Änderungsoperationen und ihren formalen Vor- und Nachbedingungen.

Eine Annäherung an konkrete Organisationsänderungen erfolgt in Abschnitt 3. Es werden beispielhafte Szenarien von Organisationsänderungen im Kontext von

Krankenhäusern und Arztpraxen beschrieben und Kategorien von Änderungen aufgestellt. Auf diese Beispielszenarien wird im weiteren Verlauf der Arbeit Bezug genommen.

In den Abschnitten 4 und 5 wird für Organisationsänderungen zunächst der statische Fall betrachtet.

In Abschnitt 4 werden semantisch sinnvolle und formal korrekte Änderungsoperationen zur Manipulation des Organisationsmodells mit Vor- und Nachbedingungen formuliert. Diese Operationen besitzen unterschiedliche Komplexität von einfachen Änderungsprimitiven, darauf aufsetzenden elementaren Änderungsoperationen bis hin zu komplexen, vordefinierten Änderungsoperationen mit einer höheren Semantik. In jedem Fall ist die Menge angebotener Änderungsoperationen vollständig. Sie erlaubt also jede semantisch sinnvolle und korrekte Transformation von Organisationsmodellen.

Abschnitt 5 geht auf die Problematik von Cross-Referenzen zwischen Organisationsmodell und Mitarbeiterzuordnungen ein. Es werden Entscheidungsstrategien vorgestellt, um zwischen kritischen und unkritischen Organisationsmodell-Änderungen in bezug auf solche Mitarbeiterzuordnungen zu unterscheiden. Schließlich werden Lösungsansätze zur Pflege dieser Referenzen angeboten, wobei eine größtmögliche Unterstützung des Modellierers angestrebt wird.

In Abschnitt 6 wird der dynamischen Fall für Organisationsänderungen zur Laufzeit betrachtet. Hier werden Lösungsansätze zur Pflege der Arbeitslisten von laufenden Instanzen nach Organisationsmodell-Änderungen vorgestellt und bewertet.

In Abschnitt 7 werden verwandte Themen und Ansätze der Forschung sowie die Realisierung von Organisationsmodell-Änderungen in ausgewählten, existierenden Workflow-Management-Systemen diskutiert. Es erfolgt dabei eine Einordnung und Abgrenzung der eigenen Arbeit.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf weiterführende Themen.

## 2 Grundlagen der Organisations-Modellierung

In diesem Abschnitt werden die Grundlagen zur Modellierung von Organisationen gegeben. Zunächst wird ein Metamodell vorgestellt, auf dessen Basis anschließend Organisationsmodelle beschrieben werden. Für beide Modelle wurde eine Mengendarstellung gewählt, die es später erleichtert, präzise Änderungsoperationen auf dem Organisationsmodell mit ihren formalen Vor- und Nachbedingungen zu formulieren (Abschnitt 4).

### 2.1 Organisations-Metamodell

Das Organisations-Metamodell gibt explizit die Strukturen vor, auf deren Basis konkrete Organisationen modelliert werden kann (2. Sprachstufe; Jablonski, Schlundt & Wedekind, 2001). Es wird zum besseren Verständnis zunächst als ER-Diagramm eingeführt. Grundlage ist das Organisations-Metamodell des Workflow-Management-Systems ADEPT<sup>2</sup>, welches in der Abteilung Datenbanken und Informationssysteme der Universität Ulm entwickelt wurde (Reichert & Dadam, 1998).

Anschließend wird ein allgemeingültiges Organisations-Metamodell in Mengendarstellung beschrieben, das sich an den individuellen Bedarf in Form eines *konfiguriertes* Organisations-Metamodell anpassen läßt.

#### 2.1.1 Allgemeingültiges Organisations-Metamodell

Das Organisations-Metamodell besteht aus einer Anzahl von Entitäten und Relationen, welche die zulässigen Entitäts- und Relationstypen für konkrete Organisationsmodelle repräsentieren. Dieser Arbeit wird das in Abbildung 2 dargestellte Organisations-Metamodell zugrunde gelegt.

Im folgenden werden die wichtigsten Entitäts- und Relationstypen kurz beschrieben.

Das zentrale Konstrukt des Metamodell ist Stelle, die bestimmte Aufgaben bündelt (Rosemann & zur Mühlen, 1997). Sie wird normalerweise durch *einen* Mitarbeiter besetzt. Um die Modellierung flexibel zu halten (Beispiel Teilzeit), sind zwischen Stelle und Mitarbeiter (n:m)-Beziehungen zugelassen. Hat ein Mitarbeiter mehrere Teilzeit-Stellen im Unternehmen, muß er bei Anmeldung am System angeben, in welcher Stelle er gerade tätig wird, damit Stellen stets eindeutig zuordenbar bleiben (Bauer, Reichert & Fries, 2001). Eine Stelle wird genau einer Organisationseinheit zugeordnet, sie kann aber beliebig viele Organisationseinheiten leiten. Stellen können in beliebig vielen Arbeitsgruppen zusammengefaßt werden. Damit können beispielsweise temporäre Projektgruppen modelliert werden. Jede Arbeitsgruppe kann von maximal einer Stelle geleitet werden.

---

<sup>2</sup> Application Development Based on Encapsulated Premodeled Process Templates  
(siehe: <http://www.informatik.uni-ulm.de/dbis> - Stand 18.04.2002)

Die hierarchischen Beziehungen zwischen Stellen (Vorgesetztenbeziehungen) werden über die Beziehungstypen *ist fachlich übergeordnet* bzw. *ist disziplinarisch übergeordnet* beschrieben. Dabei ist jeweils maximal eine übergeordnete Stelle erlaubt.

Hierarchische Beziehungen können auch für Organisationseinheiten modelliert werden. Organisationseinheiten können darüber hinaus in beliebig vielen Organisationsgruppen zusammengefaßt werden.

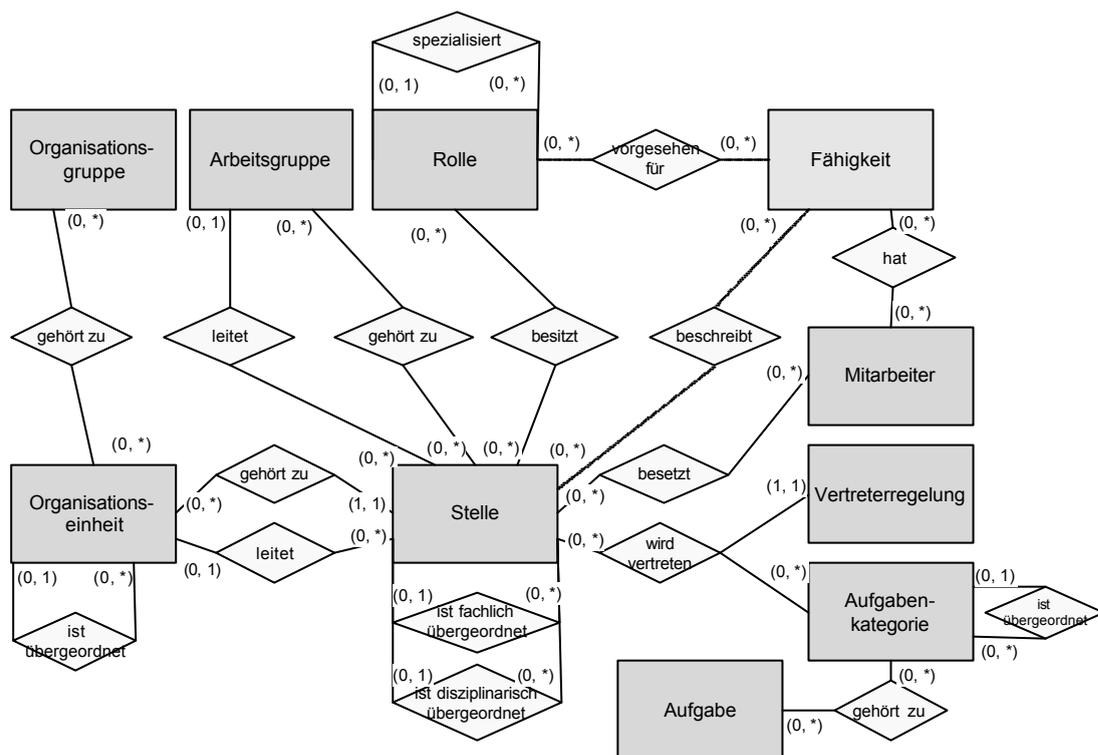


Abbildung 2 Organisations-Metamodell von ADEPT

Einer Stelle können beliebig viele Rollen zugeordnet werden. Rollen beschreiben minimale Qualifikationen und Kompetenzen des Rollenträgers (Rosemann & zur Mühlen, 1997). Rollen können weiter spezialisiert werden. Dabei "erben" die stark spezialisierten Rollen von den weniger spezialisierten (vgl. Objektorientierung).

Mitarbeiter können beliebig viele Fähigkeiten<sup>3</sup> haben. Daneben steht Mitarbeiter nur mit Stelle in Beziehung. Das hat den Vorteil, daß die übrigen Zuordnungen im Organisationsmodell erhalten bleiben, wenn ein Mitarbeiter zum Beispiel gelöscht wird.

Das Metamodell gestattet die Modellierung von Vertreterregelungen. Diese haben die Form von Bearbeiterzuordnungen (vgl. Abschnitt 5), so daß sie sich auf feste Mitarbeiter aber auch auf Rollen, Stellen usw. beziehen können. Außerdem können

<sup>3</sup> Die gestrichelte Linie zwischen Fähigkeit und Rolle bzw. Stelle (Abbildung 2) deutet an, daß die Fähigkeit zu der Stelle bzw. Rolle paßt. Hier ist jedoch keine eigenständige Beziehung vorgesehen.

Aufgabenkategorien für die Vertreterregelung festgelegt werden (Bauer, Reichert & Fries, 2001).

Ein einfaches Beispiel für ein Organisationsmodell findet sich am Ende dieses Abschnittes. Weitere komplexe Beispiele sind in (Konyen & Reichert, 1996; Schultheiß, Meyer, Mangold, Zemmler & Reichert, 1996) dokumentiert.

Den folgenden Betrachtungen dieser Arbeit liegt ein mengenbasiertes Konzept zugrunde. Dafür werden einige grundlegende Definitionen benötigt.

Die Menge *Entitytype* umfaßt alle Entitätstypen des oben beschriebenen Organisations-Metamodells, die Menge *Relationtype* entsprechend alle Relationstypen:

**Entitytype** = {*Arbeitsgruppe, Aufgabe, Aufgabenkategorie, Fähigkeit, Mitarbeiter, Organisationseinheit, Organisationsgruppe, Rolle, Stelle*}

**Relationtype** = {*gehört\_zu, besteht\_aus, leitet, wird\_geleitet, hat, ist\_fachlich\_übergeordnet, ist\_fachlich\_untergeordnet, ist\_hierarchisch\_übergeordnet, ist\_hierarchisch\_untergeordnet, ist\_disziplinarisch\_übergeordnet, ist\_disziplinarisch\_untergeordnet*}

Entitäts- und Relationstypen besitzen bestimmte, allgemeingültige Attribute, wie zum Beispiel *Familienstand* für Entitäten des Typs *Mitarbeiter* oder *Gültigkeitszeitraum* für bestimmte Relationen. Auf die exakte Darstellung dieser Attribute wird an dieser Stelle der Einfachheit halber verzichtet.

Wie aus dem eingangs vorgestellten Organisations-Metamodell leicht ersichtlich ist, werden zwischen zwei Entitäten nur Beziehungen ganz bestimmter Relationstypen zugelassen. Durch die Menge *ValidRelation* ist festgelegt, welche Relationstypen zwischen welchen Entitätstypen im Organisationsmodell gültig sind. Sie beinhaltet Tupel aus jeweils zwei Entitätstypen und einem Relationstyp:

**ValidRelation** = {(*Organisationseinheit, Stelle, besteht\_aus*), (*Arbeitsgruppe, Stelle, besteht\_aus*), (*Rolle, Rolle, spezialisiert*), (*Organisationsgruppe, Organisationseinheit, besteht\_aus*), (*Aufgabenkategorie, Aufgabe, besteht\_aus*), (*Stelle, Organisationseinheit, leitet*), (*Stelle, Arbeitsgruppe, leitet*), (*Stelle, Rolle, hat*), (*Stelle, Mitarbeiter, hat*), (*Stelle, Aufgabenkategorie, hat*), (*Mitarbeiter, Fähigkeit, hat*), (*Organisationseinheit, Organisationseinheit, ist\_hierarchisch\_übergeordnet*), (*Stelle, Stelle, ist\_fachlich\_übergeordnet*), (*Stelle, Stelle, ist\_disziplinarisch\_übergeordnet*), (*Aufgabe, Aufgabe, ist\_hierarchisch\_übergeordnet*)}

Jedem zulässigen Beziehungstyp sind Kardinalitäten für beide Richtungen der Beziehung zugeordnet. Sie geben die geforderte minimale und maximale Anzahl der jeweils an der Beziehung beteiligten Entitäten an. *Cardinality* bezeichne die

Funktion, welche jedem Beziehungstyp ein 4-Tupel ( $min_1, max_1, min_2, max_2$ ) zuordnet. Dabei definieren  $min_1$  und  $max_1$  ( $min_2$  und  $max_2$ ) für Entitäten der linken (rechten) Position im Relationstupel die minimale bzw. maximale Anzahl der Beziehungen auf Ausprägungsebene.

**cardinality:**  $ValidRelation \rightarrow \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$

**Beispiel 1: (Cardinality)**

Für den Beziehungstyp  $r = (Organisationseinheit, Stelle, besteht\_aus)$  gilt  $cardinality(r) = (0, \infty, 1, 1)$ .

Eine Organisationseinheit besteht aus *beliebig vielen* Stellen (ggf. auch Null); eine Stelle gehört zu *genau einer* Organisationseinheit.

Tabelle 1 verschafft einen Überblick über die Kardinalitäten aller gültiger Beziehungstypen aus *ValidRelation* (vgl. Abbildung 2).

Beziehungstyp	$min_1$	$max_1$	$min_2$	$max_2$
(Organisationseinheit, Stelle, besteht_aus)	0	$\infty$	1	1
(Arbeitsgruppe, Stelle, besteht_aus)	0	$\infty$	0	$\infty$
(Organisationsgruppe, Organisationseinheit, besteht_aus)	0	$\infty$	0	$\infty$
(Aufgabenkategorie, Aufgabe, besteht_aus)	1	1	0	$\infty$
(Stelle, Organisationseinheit, leitet)	0	$\infty$	0	1
(Stelle, Arbeitsgruppe, leitet)	0	$\infty$	1	1
(Stelle, Rolle, hat)	0	$\infty$	0	$\infty$
(Stelle, Mitarbeiter, hat)	0	$\infty$	0	$\infty$
(Stelle, Aufgabenkategorie, hat)	0	$\infty$	0	$\infty$
(Mitarbeiter, Fähigkeit, hat)	0	$\infty$	0	$\infty$
(Organisationseinheit, Organisationseinheit, ist_hierarchisch_übergeordnet)	0	1	0	$\infty$
(Stelle, Stelle, ist_fachlich_übergeordnet)	0	1	0	$\infty$
(Stelle, Stelle, ist_disziplinarisch_übergeordnet)	0	1	0	$\infty$
(Rolle, Rolle, spezialisiert)	0	1	0	$\infty$
(Aufgabe, Aufgabe, ist_hierarchisch_übergeordnet)	0	1	0	$\infty$

**Tabelle 1** Beziehungstypen und ihre Kardinalitäten

Daraus ergeben sich für konkrete Organisationsmodell-Ausprägungen Restriktionen, die bei der Erstellung und Änderung des jeweiligen Modells einzuhalten sind (vgl. Abschnitt 2.2.1).

## 2.1.2 Konfiguriertes Organisations-Metamodell

Für bestimmte Workflow-Anwendungen kann es sinnvoll sein, das Organisations-Metamodell an den individuellen Bedarf anzupassen. So sind Branchen denkbar, in denen bestimmte Entitätstypen, wie zum Beispiel *Arbeitsgruppe*, nicht benötigt werden. In anderen Bereichen kann wiederum Bedarf an weiteren Typattributen

bestehen, zum Beispiel an einem zusätzlichen Attribut „Umsatz“ für Mitarbeiter einer Vertriebsfirma.

Werden derartige individuelle Modifikationen des Organisations-Metamodells zugelassen, so können die von ihm abgeleiteten Organisationsmodelle weitgehend auf den Bedarf der Workflow-Anwendung zugeschnitten werden.

Auf dem kommerziellen Sektor für Workflow-Management-Systeme basiert beispielsweise MQ Series (IBM) auf einem in diesem Sinne eingeschränkten Organisations-Metamodell. Es umfaßt die Entitätstypen *Mitarbeiter*, *Rolle* und *Organisationseinheit* sowie einfache Beziehungstypen (Martschat, 2001).

Für die folgenden Betrachtung wird deshalb zulassen, daß das in Abschnitt 2.1.1 beschriebene Metamodell in zweierlei Hinsicht modifiziert werden kann: Zum einen können die Mengen *Entitytype* und *Relationtype* eingeschränkt werden, zum anderen können weitere Attribute für Entitäts- und Relationstypen definiert werden.

Da das vorgestellte Metamodell genügend ausdrucksmächtig ist, um Organisationen vollständig und adäquat abzubilden, wird auf das Hinzufügen von Entitäts- und Relationstypen verzichtet.

Ein solches angepaßtes Metamodell wird im folgenden als *konfiguriertes Organisations-Metamodell* bezeichnet.

Ein konfiguriertes Organisations-Metamodell *OMM* eines Workflow-Management-Systems ist darstellbar als Tupel von Organisations-Entitätstypen (OET), Organisations-Relationstypen (ORT), Basis-Attributen (BAttributes) und Organisations-Typattributen (OTA) (Definition 1).

**Definition 1 (konfiguriertes Organisations-Metamodell)**

$OMM = (OET, ORT, BAttributes, OTA)$

$OET \subseteq Entitytype$

$ORT \subseteq Relationtype$

$BAttributes \subseteq Label \times Domain$

$OTA \subseteq (OET \cup ORT) \times BAttributes$

Label = Menge gültiger Bezeichner aus einem Namensraum

Domain = Menge gültiger Wertebereiche

OET bezeichnet die Menge der Organisations-Entitätstypen des konfigurierten Organisationsmodells. Sie ist Teilmenge der Menge Entitytype des (allgemeinen) Organisations-Metamodells.

ORT bezeichnet die Menge von Organisations-Relationstypen eines konfigurierten Organisationsmodells. Sie ist Teilmenge der Menge Relationtype des (allgemeinen) Organisations-Metamodells.

BAttributes bezeichnet eine fixe Menge von Attributen, welche das konfigurierte Organisations-Metamodell als Basis zur Verfügung stellt. Die Attribute setzen sich aus einem Bezeichner (Label) und einem Wertebereich (Domain) zusammen<sup>4</sup>.

OTA bezeichnet die Menge von Organisations-Typattributen eines konfigurierten Organisations-Metamodells. Sie enthält Tupel aus Entitäts- oder Relationstypen und deren (zugeordneten) Basis-Attributen.

Im Umgang mit der Menge OTA ist die Hilfsfunktion  $f_{OTA}$  nützlich. Sie entspricht der Abbildung von OET oder ORT auf die Potenzmenge von BAttributes (Definition 2). Sie liefert für jeden Entitätstyp (bzw. Relationstyp) die zugehörige Attributmenge aus BAttributes.

**Definition 2 (Hilfsfunktion  $f_{OTA}$ )**

Sei  $OMM = (OET, ORT, BAttributes, OTA)$  ein konfiguriertes Organisations-Metamodell. Dann sei  $f_{OTA}$  wie folgt definiert:

$$f_{OTA}: (OET \cup ORT) \rightarrow P(BAttributes)$$

mit  $f_{OTA}(x) := \{attr \in BAttributes \mid (x, attr) \in OTA\}$

## 2.2 Organisationsmodell

Das Organisationsmodell bildet die Strukturen der realen Organisation auf Grundlage des (konfigurierten) Organisations-Metamodells ab (1. Sprachebene oder Objektebene; Jablonski, Schlundt & Wedekind, 2001).

Zunächst folgt in diesem Abschnitt die formale Darstellung des Organisationsmodells in Form von Mengen. Es werden notwendige Einschränkungen des Modells zur Konsistenzsicherung formuliert. Abschließend werden die wesentlichen Grundlagen an einem einfachen Beispiel verdeutlicht.

### 2.2.1 Formale Darstellung

Um Operationen auf einem Organisationsmodell und damit verknüpfte Einschränkungen (*Constraints*) präzise definieren zu können, wird in dieser Arbeit ein mengenbasierter Ansatz verfolgt. Dabei werden Organisations-Entitäten und die Beziehungen zwischen ihnen auf Mengen abgebildet. Auf diese Weise können Änderungen von Organisationsmodellen durch einfache Mengenoperationen mit präziser Semantik beschrieben werden.

Sei  $OM$  die Menge aller Ausprägungen eines konfigurierten Organisations-Metamodells  $OMM = (OET, ORT, BAttributes, OTA)$ . Dann ist ein Organisationsmodell  $OM \in OM$  definiert als Tupel von Organisations-Entitäten

<sup>4</sup> Um auf die Felder Label bzw. Domain zuzugreifen, wird folgende Notation verwendet:  $attr.Label$  bzw.  $attr.Domain$  für ein  $attr \in BAttributes$ .

(OE), Organisations-Relationen (OR), erweiterten Attributen (EAttributes), Organisations-Attributen (OA) und deren Wertezuweisung (OAV) (Definition 3).

**Definition 3 (Organisationsmodell)**

$OM = (OE, OR, EAttributes, OA, OAV)$

$OE \subseteq \text{Label} \times \text{OET}$

$OR \subseteq OE \times OE \times \text{ORT}$

$EAttributes \supseteq \text{BAttributes}$

$OA \subseteq (OE \cup OR) \times EAttributes$

$OAV \subseteq OA \times \text{Values}$

Values = Menge gültiger Werte

*OE* bezeichnet die Menge von Organisations-Entitäten in einem Organisationsmodell. Eine solche Entität wird durch ein Tupel aus eindeutigen Bezeichner und dem Entitätstyp repräsentiert.

*OR* bezeichnet die Menge von Organisations-Relationen, die in einem Organisationsmodell auftreten. Ein Element dieser Menge ist ein Tupel aus jeweils zwei Organisations-Entitäten und einem Relationstypen, daß die Beziehung zwischen diesen beiden Entitäten festlegt.

Die Menge *EAttributes* kann neben den Elementen der Basis-Attributmeng *BAttributes* weitere Attribute enthalten. Sie stellt damit eine Erweiterung der Basis-Attributmeng des konfigurierten Organisations-Metamodells auf Organisationsmodell-Ebene dar.

*OA* bezeichnet die Menge aller Attributzuweisungen für Entitäten und Relationen eines Organisationsmodells. *OAV* bezeichnet die Menge aller Wertzuweisungen für Entitäts- und Relationsattribute.

Definition 4 gibt zunächst Hilfsfunktionen zum Umgang mit Entitäts- oder Relationsattributen an.

**Definition 4 (Hilfsfunktionen *fOA*, *fOAV*)**

Sei  $OM = (OE, OR, EAttributes, OA, OAV)$  ein Organisationsmodell. Dann sei definiert:

$fOA: (OE \cup OR) \rightarrow P(EAttributes)$

mit  $fOA(x) = \{attr \in EAttributes \mid (x, attr) \in OA\}$

$fOAV: (OE \cup OR) \times EAttributes \rightarrow \text{Values}$

mit  $fOAV((x, attr)) = value$ , wobei gilt:  $((x, attr), value) \in OAV$

Die Funktion *fOA* liefert für jede Entität (bzw. Relation) die zugehörige Attributmeng. Die Funktion *fOAV* liefert den Wert *value* eines Entitätsattributs (bzw. Relationsattributs).

Damit das Organisationsmodell  $OM = (OE, OR, EAttributes, OA, OAV)$  korrekt und konsistent ist, müssen für die Mengen  $OA$  und  $OAV$  folgende Einschränkungen eingehalten werden:

**Einschränkungen für  $OA$ ,  $OAV$ :**

- (1)  $((x, attr), value) \in OAV \Rightarrow value \in attr.Domain$
- (2)  $x = (entitylabel, entitytype) \in OE: f_{OA}(x) \supseteq f_{OTA}(entitytype)$
- (3)  $x = (oe1, oe2, relationtype) \in OR: f_{OA}(x) \supseteq f_{OTA}(relationtype)$
- (4)  $\forall (x1, value1), (x2, value2) \in OAV \text{ mit } x1 = x2 \Rightarrow value1 = value2$

Wird einem Organisations-Attribut ein Wert zugeordnet, so muß dieser aus dem Wertebereich des Attributes stammen (1). Die Menge von Organisations-Attributen einer Entität (bzw. Relation) ist Obermenge der zum Entitätstyp (bzw. Relationstyp) gehörenden Organisations-Typattributmengemenge (2) (3). Jedem Organisations-Attribut kann nur genau ein Wert zugeordnet werden (4).

Eine weitere Einschränkung besteht für Relationen aus der Menge  $OR$  eines Organisationsmodells  $OM = (OE, OR, EAttributes, OA, OAV)$ :

**Einschränkung für  $OR$ :**

- (5)  $oe1 = (entitylabel1, entitytype1)$   
 $oe2 = (entitylabel2, entitytype2)$   
 $(oe1, oe2, relationtype) \in OR \Rightarrow$   
 $(entitytype1, entitytype2, relationtype) \in ValidRelation$

Eine Relation ist dann gültig, wenn die Tupel aus ihren Entitätstypen und dem Relationstyp in der Menge *ValidRelation* (vgl. Abschnitt 2.1.1.) enthalten sind (5).

Für gültige Beziehungstypen bestehen schließlich Beschränkungen, die durch ihre Kardinalitäten gegeben sind (6):

**Kardinalitätsbeschränkung:**

- (6) Seien  $oe = (entitylabel, entitytype) \in OE$  und  
 $vr = (entitytype1, entitytype2, relationtype) \in ValidRelation^{OMM}$   
 mit  $entitytype \in \{entitytype1, entitytype2\}$  beliebig.  
 Sei ferner  $cardinality(vr) = (min_1, max_1, min_2, max_2)$

Dann muß gelten:

Falls  $entitytype1 = entitytype$

$$min_1 \leq |\{oe' \mid (oe, oe', relationtype) \in OR\}| \geq max_1$$

Falls  $entitytype2 = entitytype$

$$min_2 \leq |\{oe' \mid (oe', oe, relationtype) \in OR\}| \geq max_2$$

Die Kardinalitätsbeschränkung für einen Beziehungstyp ist dann eingehalten, wenn die Anzahl aller Relationen des Types mit einer bestimmten Entität *links* im Tupel das Minimum  $\min_1$  nicht unterschreitet und das Maximum  $\max_1$  nicht überschreitet. Analog darf die Anzahl aller Relationen des Types mit dieser Entität *rechts* im Tupel das Minimum  $\min_2$  nicht unterschreiten und das Maximum  $\max_2$  nicht überschreiten.

**Beispiel 2: (Kardinalitätsbeschränkung)**

Gegeben sei ein Organisationsmodell  $OM = (OE, OR, EAttributes, OA, OAV)$  mit der Relation (ärztlicher Leiter, Klinik, leitet). Das zugehörige Element aus ValidRelation ist (Stelle, Organisationseinheit, leitet). Die Kardinalitäten werden ermittelt durch:  
 $\text{cardinality}(\text{Stelle, Organisationseinheit, leitet}) = (0, \infty, 0, 1)$ .

Um die Kardinalitätsbeschränkungen einzuhalten, dürfen in dem Organisationsmodell beliebig viele Relationen vom Typ „leitet“ existieren, bei denen „ärztlicher Leiter“ links im Tupel steht, aber nur null oder eine Relation vom Typ „leitet“, bei der „Klinik“ rechts im Tupel steht. Das heißt der ärztliche Leiter kann auch mehrere Organisationseinheiten leiten, die Klinik kann dagegen nur durch maximal eine Stelle geleitet werden.

Ein korrektes Organisationsmodell  $OM \in OM$  muß all diesen Einschränkungen genügen.

## 2.2.2 Beispiel für ein Organisationsmodell

An dieser Stelle sollen die vorangegangenen Überlegungen zu den Grundlagen von Organisationsmodellen an einem einfachen Beispiel veranschaulicht werden:

**Beispiel 3: (Organisationsmodell)**

Zu einer ärztlichen Praxis gehören zwei Bereiche: eine Verwaltung und ein Behandlungsbereich.

Der Verwaltung ist die Stelle einer Arzthelferin zugeordnet, die Frau Lehmann inne hat. Zu der Stelle gehören Aufgaben beim Empfang, als Arzthelfer und bei der Abrechnung. (Diese Aufgabenkomplexe werden als Rollen modelliert.) Frau Lehmann kann darüber hinaus Blut abnehmen und mit dem PC umgehen.

Dem Behandlungsbereich sind eine Arztstelle und ein weiterer Arzthelfer zugeordnet. Der Praxis-Arzt ist Dr. Meier, der besondere Fähigkeiten mit Ultraschall hat. Er leitet die Praxis und ist fachlich und disziplinarisch den beiden Arzthelfern übergeordnet. Neben der Rolle als Arzt gehört auch die Rolle als Ausbilder zu der Stelle. Die zugehörige Arzthelferstelle ist von Herrn Brandt besetzt, der neben Blut abnehmen auch Türkisch kann.

Abbildung 3 zeigt ein stark vereinfachtes Organisationsmodell dieser ärztlichen Praxis.

Dem Organisationsmodell  $OM^{Praxis}$  liegt das *konfigurierte* Organisations-Metamodell  $OMM^{Praxis}$  zugrunde:

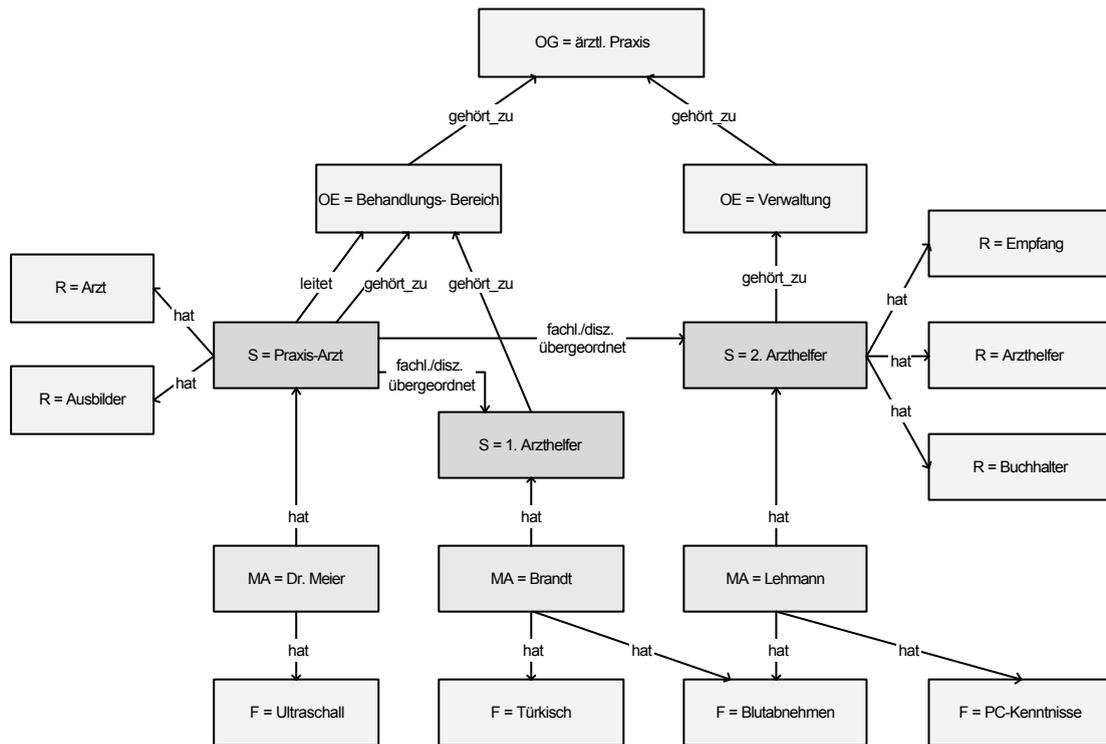
$OMM^{Praxis} = (OET, ORT, BAttributes, OTA)$  mit

$OET = \{Organisationsgruppe, Organisationseinheit, Stelle, Mitarbeiter, Fähigkeit, Rolle\}$

$ORT = \{gehört\_zu, hat, leitet, fachlich\_übergeordnet, disziplinarisch\_übergeordnet\}$

$BAttributes = \{(Name, character), (Alter, integer), (Gehalt, decimal), (Anteil, percent), \dots\}$

$OTA = \{(Mitarbeiter, (Name, character)), (Mitarbeiter, (Alter, integer)), (hat, (Anteil, percent)), \dots\}$



(Pfeile stehen für die Leserichtung der Beziehungen;  
 F = Fähigkeit; MA = Mitarbeiter; OE = Organisationseinheit; OG = Organisationsgruppe; R = Rolle; S = Stelle)

**Abbildung 3 Beispiel für ein einfaches Organisationsmodell einer ärztlichen Praxis**

Dieses konfigurierte Organisations-Metamodell enthält also nicht sämtliche vom allgemeingültigen Organisations-Metamodell angebotenen Entitäts- und Relationstypen, beispielsweise sind keine Arbeitsgruppen definiert und sollen später auch nicht hinzu genommen werden.

Formal läßt sich nun das Organisationsmodell der Arztpraxis darstellen als Tupel von Organisations-Entitäten (OE), Organisations-Relationen (OR), erweiterten Attributen (EAttributes), Organisations-Attributen (OA) und deren Wertezuweisung (OAV):<sup>5</sup>

<sup>5</sup> Entitätstypen abgekürzt wie folgt: F = Fähigkeit; MA = Mitarbeiter; OE = Organisationseinheit; OG = Organisationsgruppe; R = Rolle; S = Stelle

$OM^{Praxis} = (OE, OR, EAttributes, OA, OAV)$  mit

$OE = \{(ärztliche\ Praxis, OG), (Behandlungsbereich, OE), (Verwaltung, OE), (Praxis-Arzt, S), (1.Arzhelfer, S), (2.Arzhelfer, S), (Dr. Meier, MA), (Brandt, MA), (Lehmann, MA), (Arzt, R), (Ausbilder, R), (Empfang, R), (Arzhelfer, R), (Buchhalter, R), (Ultraschall, F), (Türkisch, F), (Blutabnehmen, F), (PC-Kenntnisse, F)\}$

$OR = \{((Verwaltung, OE), (ärztliche\ Praxis, OG), gehört\_zu), ((Dr. Meier, MA), (Praxis-Arzt, S), hat), ((Praxis-Arzt, S), (Behandlungsbereich, OE), leitet), ((Praxis-Arzt, S), (1.Arzhelfer, S), fachlich\_übergeordnet), ((Praxis-Arzt, S), (1.Arzhelfer, S), disziplinarisch\_übergeordnet), \dots\}$

$EAttributes = \{(Gesundheitszeugnis, boolean), (Überstunden, integer), \dots, (Name, character), (Alter, integer), (Gehalt, decimal), (Anteil, percent), \dots\}$

$OA = \{((Brandt, MA), (Gesundheitszeugnis, boolean)), ((Brandt, MA), (Alter, integer)), (((Dr. Meier, MA), (Praxis-Arzt, S), hat), (Anteil, percent)), \dots\}$

$OAV = \{(((Brandt, MA), (Gesundheitszeugnis, boolean)), true), (((Brandt, MA), (Alter, integer)), 37), (((Dr. Meier, MA), (Praxis-Arzt, S), hat), (Anteil, percent)), 100), \dots\}$

Anmerkung: Bei einer konkreten Implementierung des Modells würde man die Instanzen eines bestimmten Typs, zum Beispiel (Brandt, MA), durch eindeutige *Object-Identifizier* kennzeichnen und das hier verwendete *Label* "Brandt" als Attributwert von *Name* führen.

Auf das Modell sind die oben definierten Funktionen  $fOTA(x)$ ,  $fOA(x)$ ,  $fOAV(x)$  anwendbar, beispielsweise:

$fOTA(MA) = \{(Name, character), (Alter, integer), \dots\}$

$fOA((Brandt, MA)) = \{(Gesundheitszeugnis, boolean), (Alter, integer), \dots\}$

$fOAV(((Brandt, MA), (Alter, integer))) = \{37\}$

Wie leicht zu überprüfen ist, erfüllt das Organisationsmodell sämtliche Einschränkungen, die in Abschnitt 2.2.1 definiert wurden.

## 2.3 Zusammenfassung

In diesem Abschnitt wurde ein mengenbasiertes Ansatz zur Modellierung von Organisationen vorgestellt.

Zunächst wurde ein ausdrucks mächtiges Organisations-Metamodell definiert, von dem sich beliebige Organisationsmodelle ableiten lassen. Dieses Metamodell kann darüber hinaus an den konkreten Bedarf des Anwenders in Form eines konfigurierten Organisations-Metamodells angepaßt werden.

Zur Wahrung der Konsistenz und semantischen Korrektheit der abgeleiteten Organisationsmodelle wurden gewisse Einschränkungen formuliert, die bei der Modellierung von Organisationen eingehalten werden müssen.

### 3 Organisationsmodell-Änderungen

Nachdem im vorigen Abschnitt das grundlegende Konzept zur Modellierung von Organisationen erläutert wurde, soll nun betrachtet werden, auf welche Weise Organisationen sich in der Realität ändern und wie es sich auf das zugehörige Organisationsmodell auswirkt.

Zunächst werden einige typische Änderungsszenarien beschrieben. Anschließend werden die wichtigsten Organisationsänderungen in Kategorien zusammengefaßt.

#### 3.1 Anwendungsszenarien

Organisationen wie Unternehmen oder öffentlichen Einrichtungen (zum Beispiel Universitäten, Kliniken, Behörden) unterliegen einem ständigen Wandel.

Besonders häufig sind personelle Veränderungen, etwa infolge von Neueinstellungen oder Entlassungen von Mitarbeiter, durch Rotationen, usw.

Strukturelle Änderungen der Organisation sind eher seltener. Meist sind wirtschaftliche oder administrative Gründe die Ursache für Umstrukturierungen bzw. Reorganisationen.

Änderungen können temporär oder dauerhaft sein. Temporäre Änderungen können ggf. über Vertreterregelungen abgefangen werden, so daß sie an dieser Stelle nicht weiter vertieft werden sollen.

Die folgenden Szenarien bewegen sich im Kontext von ärztlichen Praxen (vgl. Beispiel 3, Abschnitt 2.2.2) bzw. medizinischen Kliniken. Eine ausführliche Dokumentation einer Kliniksstruktur am Beispiel der Ulmer Universitäts-Frauenklinik findet sich bei Konyen, Reichert und Schultheiß (1996).

##### a) Personelle Änderungen

Personelle Änderungen beschränken sich auf Entitäten vom Typ *Mitarbeiter* und ihre Beziehungen. Die übrigen Strukturen der Organisation sind davon gewöhnlich nicht betroffen. Diese Änderungen sind in den meisten Fällen wenig komplex, kommen dafür aber häufig vor. Daher ist es sinnvoll, daß die wesentliche Funktionalität sowohl im Organisationsmodell (zum Beispiel Zuordnung zu Organisationseinheiten oder Rollen) als auch bei den Bearbeiterzuordnungen der Aktivitäten nicht an feste Mitarbeiter sondern an Stellen, Rollen, usw. gekoppelt wird. Auf diese Weise können zur Laufzeit trotz Personalwechsels Bearbeiterformel ohne weiteres aufgelöst und Arbeitslisten geschrieben werden.

Folgende Szenarien zeigen einfache personelle Änderungen. Sie beziehen sich auf die Arzt-Praxis aus Beispiel 3 (Abschnitt 2.2.2):

	<b>Beschreibung</b>	<b>Art der Änderung</b>
Szenario 1	Frau Lehmann scheidet aus der Praxis aus	Mitarbeiter löschen
Szenario 2	Frau Schumann übernimmt die Stelle der 2. Arzthelferin	Mitarbeiter einfügen
Szenario 3	Frau Schumann kann Steno schreiben	Fähigkeit eines Mitarbeiter einfügen
Szenario 4	Herr Brandt kann nicht mehr genügend Türkisch	Fähigkeit eines Mitarbeiter löschen
Szenario 5	Herr Brandt wechselt auf die 2. Arzthelfer-Stelle in der Verwaltung	Mitarbeiter verschieben
Szenario 6	Frau Schumann hat geheiratet und damit ihre Steuerklasse geändert	Attribut ändern

**Tabelle 2 Szenarien personeller Änderungen von Organisationen**

### b) Einfache strukturelle Änderungen

Strukturelle Änderungen sind nicht an konkrete Mitarbeiter gebunden, sondern betreffen grundlegende, relativ stabile Strukturen der Organisation. Selbst verhältnismäßig einfache, strukturelle Änderungen, wie in Tabelle 3 dargestellt, können weiteren Änderungsbedarf nach sich ziehen. Wenn wie in Szenario 10 eine Organisationseinheit gelöscht wird, muß auf der semantischen Ebene geklärt werden, was mit den zu ihr gehörenden Stellen geschieht. Werden diese ebenfalls gelöscht oder werden diese einem anderen Bereich zugeordnet? Diese Informationen sind später auch für die begleitende Anpassung von Bearbeiterformeln wichtig.

	<b>Beschreibung</b>	<b>Art der Änderung</b>
Szenario 7	Stelle eines Auszubildenden wird im Behandlungsbereich eingerichtet	Stelle einfügen
Szenario 8	Stelle des 1. Arzthelfers im Behandlungsbereich wird eingespart	Stelle löschen
Szenario 9	Bereich Krankengymnastik wird geschaffen	Organisationseinheit einfügen
Szenario 10	Bereich Verwaltung wird aufgelöst	Organisationseinheit löschen
Szenario 11	Verwaltung und Behandlungsbereich werden zusammengelegt	Organisationseinheiten vereinen
Szenario 12	Behandlungsbereich wird in Arztbereich und Pflegebereich geteilt	Organisationseinheit teilen
Szenario 13	Stelle des Auszubildenden wird vom Behandlungsbereich der Verwaltung zugeordnet	Stelle verschieben
Szenario 14	1. Arzthelfer wird Ausbildungsverantwortlicher	Rollenzuordnung einfügen

**Tabelle 3 Szenarien einfacher struktureller Änderungen von Organisationen**

### c) Komplexe strukturelle Änderungen

Neben einfachen strukturellen Änderungen kommen in der Praxis auch (beliebig) komplexe Organisationsänderungen vor, die an die Anpassung des Organisationsmodells sehr hohe Anforderungen zur Wahrung der Semantik, Korrektheit und Konsistenz stellen.

Tabelle 4 zeigt einige Beispiele für komplexe Änderungen.

Werden etwa wie in Szenario 16 zwei Organisationsmodelle vereint, müssen ein einheitliches konfiguriertes Organisations-Metamodell bestimmt sowie Namens- und Strukturkonflikte ermittelt und aufgelöst werden, etwa vergleichbar mit einer Schema-Integration bei Datenbanken (Dadam, 1996; S. 100 ff.). In den Szenarien 17 bis 19 werden gemeinsame Sichten oder Schnittstellen der beteiligten Organisationsmodelle benötigt.

	Beschreibung	Art der Änderung
Szenario 15	In der Ausbildung von Ärzten kommen häufig Rotationen vor. Dabei werden die Ausbildungsstellen regelmäßig anderen Bereichen zugeordnet	regelmäßig wechselnde Zuordnung von Stellen zu Organisationseinheiten
Szenario 16	Fusion zweier Kliniken zu Klinikverbund	Vereinigen der gesamten Organisationsmodelle
Szenario 17	Bereich Krankengymnastik einer Klinik wird ausgelagert als eigenständige Organisation	Trennen des Organisationsmodells
Szenario 18	Zusammenarbeit zwischen zwei Kliniken bei klinik-übergreifenden Aufgaben	Schnittstellen zwischen den Organisationen nötig
Szenario 19	Bereich Nierentransplantation von Gefäß-Klinik zur Chirurgischen Klinik zugeteilt	Organisationseinheit aus Organisation in andere Organisation verschieben

**Tabelle 4** Szenarien komplexer struktureller Änderungen von Organisationen

### 3.2 Klassifizierung von Organisationsmodell-Änderungen

Aus den im vorigen Abschnitt skizzierten Szenarien von Organisationsänderungen lassen sich für das Organisationsmodell bestimmte Kategorien typischer Änderungen ableiten, die in ähnlicher Weise immer wieder vorkommen. Die wichtigsten Änderungskategorien sind mit dem Hinweis auf die jeweiligen (einfachen) Beispiel-Szenarien (vgl. Tabellen 2, 3) in Tabelle 5 aufgeführt.

Komplexe strukturelle Änderungen (vgl. Tabelle 4) sind in dem Maße seltener nötig. Sie setzen sich im wesentlichen aus den einfacheren Änderungen zusammen und sind hier deshalb nicht gesondert aufgeführt.

Kategorie	Szenarien
Entität oder Relation einfügen	2; 3; 7; 9; 14
Entität oder Relation löschen	1; 4; 8; 10
Attribut einer Entität (oder Relation) ändern	6
Entität teilen	12
Entität vereinigen	11
Entität verschieben	5; 13

**Tabelle 5** Änderungskategorien mit Verweis auf Beispiel-Szenarien

Einige Änderungen sind nicht bei allen Entitätstypen semantisch sinnvoll. Beispielsweise können Mitarbeiter weder zusammengefaßt noch geteilt werden. Tabelle 6 gibt einen Überblick über semantisch sinnvolle Änderungen bezogen auf die Entitätstypen.

Entitätstyp	Einfügen	Löschen	Attribut Verändern	Teilen	Vereinigen	Verschieben
Arbeitsgruppe	x	x	x	x	x	-
Aufgabe	x	x	x	x	x	-
Aufgabenkategorie	x	x	x	x	x	-
Fähigkeit	x	x	x	x	x	-
Mitarbeiter	x	x	x	-	-	x
Organisationseinheit	x	x	x	x	x	x
Organisationsgruppe	x	x	x	x	x	-
Rolle	x	x	x	x	x	-
Stelle	x	x	x	x	x	x

**Tabelle 6** Entitätstypen mit semantisch sinnvolle Änderungskategorien (x sinnvoll)

Bestimmte Arten von Änderungen sind bei sämtlichen Entitäts- und Relationstypen möglich. Diese elementaren Änderungskategorien sind *Einfügen*, *Löschen* und *Attribut Ändern*.

Komplexe Änderungen sind *Vereinigen*, *Teilen* und *Verschieben* von Entitäten. Sie lassen sich prinzipiell auf elementare Änderungen zurückführen. Beispielsweise entspricht *Verschieben* dem *Einfügen* und *Löschen* von Relationen. Eine speziellere Behandlung dieser komplexen Änderungen kann dennoch vorteilhaft sein: So läßt sich ihre höhere Semantik später bei der Anpassung von Bearbeiterformeln nutzen.

Mit diesen sechs Änderungskategorien *Einfügen*, *Löschen*, *Attribut Ändern*, *Vereinigen*, *Teilen* und *Verschieben* lassen sich nun alle Organisationsänderungen abbilden.

Im nächsten Abschnitt werden die entsprechenden Operationen definiert, mit denen sich diese Änderungen an dem dieser Arbeit zugrundeliegenden Organisationsmodell beschreiben lassen.

### **3.3 Zusammenfassung**

In diesem Abschnitt wurden anhand von einfachen Szenarien die wichtigsten Kategorien von Organisationsänderungen abgeleitet. Dies sind die elementaren Änderungen *Einfügen*, *Löschen* und *Attribut Ändern* von Entitäten oder Relationen, sowie die komplexen Änderungen *Vereinigen*, *Teilen* und *Verschieben* von Entitäten. Die semantisch höheren, komplexen Änderungen lassen sich ggf. nur auf bestimmte Entitätstypen anwenden.

Mit den elementaren und komplexen Änderungen lassen sich alle organisatorischen Änderungen beschreiben.

## 4 Operationen für Organisationsmodell-Änderungen

Nachdem im vorigen Abschnitt die wichtigsten Kategorien von elementaren und semantisch komplexeren Änderungen in Organisationen erarbeitet wurden, sollen nun die Operationen zur Verfügung gestellt werden, mit denen sich solche Änderungen am Organisationsmodell durchführen lassen.

### 4.1 Anforderungen an Änderungsoperationen

Für die Modellierung von Organisationen wurde in dieser Arbeit ein mengenbasierter Ansatz gewählt. Auf diese Weise können - wie zuvor verdeutlicht - die notwendigen Änderungsoperationen auf Mengenoperationen abgebildet werden. Die präzise Semantik dieser Änderungen wird dabei über eine exakte Definition von Vor- und Nachbedingungen der Mengenoperationen ausgedrückt.

Nach diesem Ansatz entspricht eine Änderung des Organisationsmodells OM der Überfunktionsfunktion  $OM \rightarrow OM^*$  mit dem resultierenden Organisationsmodell  $OM^*$ .

Sowohl das originale als auch das resultierende Organisationsmodell müssen insbesondere ausdrucks mächtig, konsistent und korrekt sein. Für die zugehörigen Änderungsoperationen werden deshalb folgende Eigenschaften gefordert:

- *Vollständigkeit*: Mit Hilfe der Änderungsoperationen sollen alle (sinnvollen) organisatorischen Änderungen realisierbar sein.
- *Minimalität*: Eine Änderungsoperation soll nicht durch eine oder mehrere andere Änderungsoperationen ersetzbar sein.
- *Korrektheit und Konsistenz*: Die Änderungsoperationen sollen von einem korrekten, konsistenten Organisationsmodell wiederum zu einem korrekten und konsistenten Organisationsmodell führen.
- *Eindeutige Semantik*: Die Änderungsoperationen sollen die Semantik der organisatorischen Änderung einhalten.

Die Forderung nach Minimalität von Änderungsoperationen ist nicht in jedem Fall sinnvoll aufrecht zu erhalten. Wie bereits diskutiert, kann es vorteilhaft sein, auch für komplexe Änderungen wie Vereinigen oder Teilen entsprechende Änderungsoperationen zu formulieren, obwohl diese durch elementare Änderungsoperationen auszudrücken sind.

Neben diesen grundlegenden Eigenschaften sollten auch die folgenden zusätzlichen Anforderungen an die Änderungsoperationen erfüllt werden:

- *Benutzerfreundlichkeit*: Aufwendige Interaktionen sollen vermieden werden. Der Modellierer sollte zum Beispiel nur (unbedingt) notwendige Parameter angeben müssen. Das System sollte ihn dabei unterstützen.
- *Implementierbarkeit*: Die Änderungsoperationen sind in einem realen Informationssystem implementierbar.

Neben den hier genannten Anforderungen gibt es noch viele weitere Entwurfsziele, wie effiziente Anwendbarkeit, die Einhaltung von Sicherheitsaspekten usw. (Reichert, 2000).

## 4.2 Herleitung der Änderungsoperationen

In Abschnitt 3 wurden Kategorien von organisatorischen Änderungen gebildet.

Elementaren Änderungen sind demnach *Einfügen* und *Löschen* von organisatorische Entitäten und Relationen sowie deren *Attribut Ändern*. Zu den komplexen Änderungen gehören das *Vereinigen*, *Teilen* und *Verschieben* von bestimmten Entitäten.

Für diese organisatorischen Änderungen müssen nun die entsprechenden Änderungsoperationen entwickelt werden.

Elementare Änderungen lassen sich auf *jeden* Entitäts- und Relationstyp anwenden. Mit ihrer einfachen Semantik sind sie damit auf der Ebene der 3. Sprachstufe (Metametastufe; Jablonski, Schlundt & Wedekind, 2001) angesiedelt, auf der lediglich zwischen Organisations-Entitäten und -Relationen unterschieden wird (Abbildung 4).

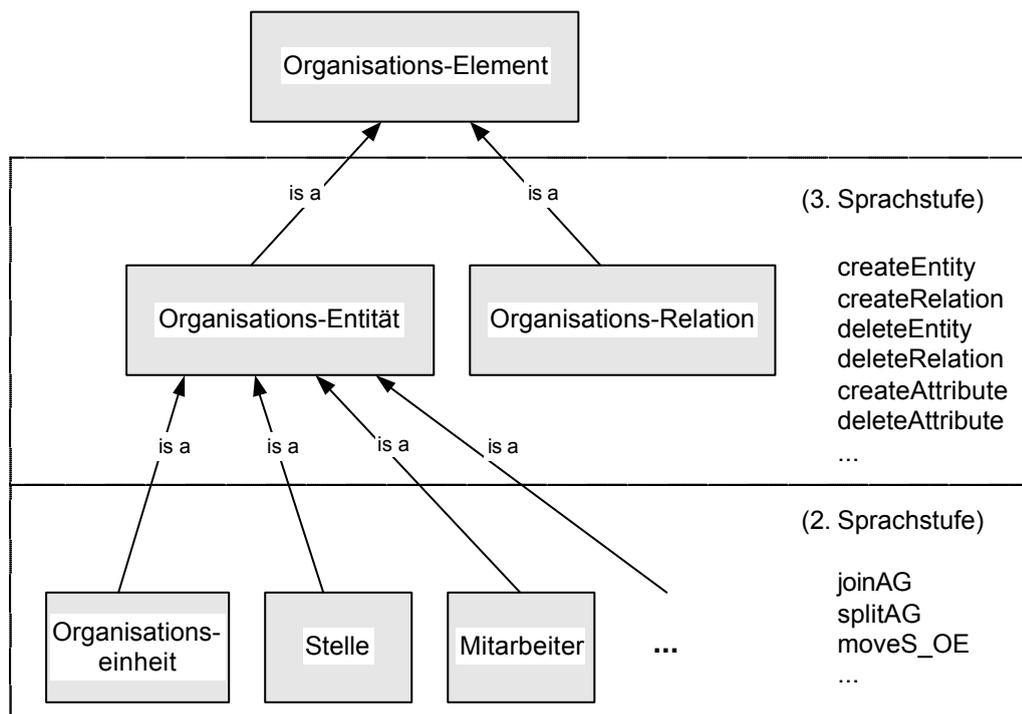


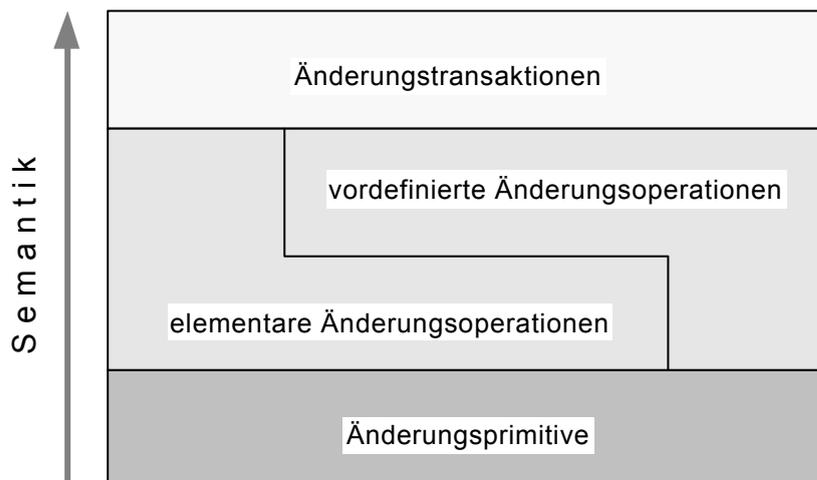
Abbildung 4 Einordnung der Änderungsoperationen in Bezug auf die Modellierungsebene

Komplexe Änderungen weisen dagegen eine höhere Semantik auf und lassen sich nur auf *bestimmte* Entitätstypen anwenden (vgl. Tabelle 6; Abschnitt 3.2). Damit bewegen sie sich auf der Ebene der 2. Sprachstufe (Metastufe; Jablonski, Schlundt & Wedekind, 2001), auf der zwischen den unterschiedlichen Organisations-Entitätstypen unterschieden wird (Abbildung 4).

Diese hier besprochenen Änderungen beziehen sich stets auf ein konkretes Organisationsmodell. Falls ein Organisations-Metamodell konfiguriert werden soll (vgl. Abschnitt 2.1.2), müßten Änderungsoperationen in bezug auf *Typen* angewandt werden, also Löschen<sup>6</sup> von Entitäts- und Relationstypen, sowie Hinzufügen oder Löschen von Typattributen. Diese Änderungen am Organisations-Metamodell werden hier nicht näher behandelt.

Elementare und komplexe Änderungsoperationen finden normalerweise gekapselt im Rahmen von Änderungstransaktionen statt. Auf diese Weise wird erreicht, daß bei einer Organisationsmodell-Änderung sämtliche in Abschnitt 2.2.1 formulierten Einschränkungen eingehalten werden können. Bei Anwendung einer einzelnen Änderungsoperation kann diese Konsistenzsicherung nicht gegeben werden. Auf diese Problematik wird in Abschnitt 4.5 näher eingegangen.

Elementare und vordefinierte Änderungsoperationen setzen sich aus einzelnen Änderungsprimitiven zusammen, die selbst nicht weiter zerlegbar sind. Abbildung 5 zeigt den strukturellen Zusammenhang zwischen Änderungsprimitiven, elementaren und vordefinierten (komplexen) Änderungsoperationen sowie Änderungstransaktionen.



**Abbildung 5** Zusammenhang zwischen Änderungsprimitiven, elementaren und vordefinierten Änderungsoperationen und Änderungstransaktionen

In den folgenden Abschnitten werden diese Änderungsprimitive, die elementaren und vordefinierten (komplexen) Änderungsoperationen sowie die Änderungstransaktionen dargestellt.

<sup>6</sup> Das Hinzufügen von Entitäts- und Relationstypen ist nicht gestattet (vgl. Abschnitt 2.1.2).

### 4.3 Änderungsprimitive

Änderungsprimitive sind elementare Operationen, welche die Mengen des Organisationsmodells  $OM \in OM$  manipulieren. Sie führen noch nicht zwingend zur Korrektheit des resultierenden Organisationsmodells  $OM^*$ , sondern legen zunächst eine eindeutige Semantik der Änderungsoperationen fest. Die Korrektheit des veränderten Organisationsmodells wird im Rahmen der übergeordneten elementaren bzw. komplexen Änderungsoperationen sichergestellt.

Relevant sind Änderungsprimitive zum Hinzufügen und Löschen von Organisations-Entitäten, Organisations-Relationen und Organisations-Attributen sowie zum Setzen von Wertzuweisungen von Attributen. Außerdem ist es sinnvoll zuzulassen, daß in die erweiterte Attributmenge *EAttributes* weitere Attribute aufgenommen bzw. aus ihr entfernt werden können.

Einen Überblick über diese Änderungsprimitive gibt Tabelle 7.

Menge	add	delete	andere
OE	addOrgEntity	delOrgEntity	
OR	addOrgRelation	delOrgRelation	
OA	addOrgAttr	delOrgAttr	
OAV			setOrgAttrValue
EAttributes	addAttribute	delAttribute	

Tabelle 7 Überblick über relevante Änderungsprimitive

Bei Anwendung dieser Änderungsprimitive auf ein Organisationsmodell  $OM$  wird dieses in ein Organisationsmodell  $OM^*$  überführt. In den folgenden Übersichten werden diese Primitive und ihre Semantik vorgestellt. Dafür seien jeweils ein Organisationsmodell  $OM = (OE, OR, EAttributes, OA, OAV)$  und das zugehörige Organisations-Metamodell  $OMM = (OET, ORT, BAttributes, OTA)$  gegeben.

#### a) Organisations-Entität

<b>addOrgEntity(OM, entitylabel, entitytype)</b>	
Erklärung:	fügt dem Organisationsmodell $OM$ eine neue Entität mit der Bezeichnung <i>entitylabel</i> und dem Typ <i>entitytype</i> zu.
Vorbedingung <sup>7</sup> :	entitylabel $\in$ Label entitytype $\in$ OET
Nachbedingung:	$OE^* = OE \cup \{(entitylabel, entitytype)\}$
<b>delOrgEntity(OM, oe)</b>	
Erklärung:	löscht aus dem Organisationsmodell $OM$ die Entität <i>oe</i> .
Vorbedingung:	(keine)
Nachbedingung:	$OE^* = OE \setminus \{oe\}$

<sup>7</sup> Die Vorbedingungen sind jeweils minimal gehalten, insbesondere wurde auf Bedingungen verzichtet, die aufgrund der Idempotenz von Operationen nicht zwingend sind, wie z.B. bei addOrgEntity die Forderung  $(entitylabel, entitytype) \notin OE$ .

## b) Organisations-Relation

<b>addOrgRelation(OM, oe1, oe2, relationtype)</b>	
Erklärung:	fügt dem Organisationsmodell <i>OM</i> eine neue Relation vom Typ <i>relationtype</i> zwischen den Entitäten <i>oe1</i> und <i>oe2</i> zu.
Vorbedingung:	$oe1, oe2 \in OE$ mit $oe1 = (\text{entitylabel1}, \text{entitytype1})$ , $oe2 = (\text{entitylabel2}, \text{entitytype2})$ $relationtype \in ORT$ $(\text{entitytype1}, \text{entitytype2}, relationtype) \in \text{ValidRelation}$
Nachbedingung:	$OR^* = OR \cup \{(oe1, oe2, relationtype)\}$
<b>delOrgRelation(OM, or)</b>	
Erklärung:	löscht aus dem Organisationsmodell <i>OM</i> die Relation <i>or</i> .
Vorbedingung:	(keine)
Nachbedingung:	$OR^* = OR \setminus \{or\}$

## c) Organisations-Attribut und -Wert

<b>addOrgAttr(OM, x, attr)</b>	
Erklärung:	ordnet der Entität oder Relation <i>x</i> des Organisationsmodells <i>OM</i> das Attribut <i>attr</i> zu.
Vorbedingung:	$x \in OE \cup OR$ $attr \in E\text{Attributes}$
Nachbedingung:	$OA^* = OA \cup \{(x, attr)\}$
<b>delOrgAttr(OM, oa)</b>	
Erklärung:	löscht das einer Entität oder Relation zugeordnete Attribut <i>oa</i> aus dem Organisationsmodell <i>OM</i> . Falls eine zu <i>oa</i> gehörende Wertzuordnung in <i>OAV</i> existiert, wird auch diese gelöscht.
Vorbedingung:	(keine)
Nachbedingung:	$OA^* = OA \setminus \{oa\}$ $OAV^* = OAV \setminus \{oav\}$ , falls $\exists oav = (x, attr, val)^8$ mit $(x, attr) = oa$
<b>setOrgAttrValue(OM, oa, value)</b>	
Erklärung:	ordnet dem Organisations-Attribut <i>oa</i> des Organisationsmodells <i>OM</i> den Wert <i>value</i> <sup>9</sup> zu. Dabei wird beachtet, daß ggf. ein früherer Eintrag eines Wertes zu diesem Organisations-Attribut entfernt wird.
Vorbedingung:	$oa \in OA$ $oa = (x, attr) \Rightarrow value \in attr.\text{Domain}$
Nachbedingung:	$OAV^* = \begin{cases} (OAV \setminus \{(oattr, ovalue)\}) \cup \{(oa, value)\} & \text{falls} \\ & \exists (oattr, ovalue) \in OAV \text{ mit } oattr = oa \\ OAV \cup \{(oa, value)\} & \text{sonst} \end{cases}$

<sup>8</sup> Wegen Einschränkung (4) in Abschnitt 2.2.1. gibt es maximal ein solches Tupel.

<sup>9</sup> Es sei zugelassen, *value* = undefined zu setzen. Auf diese Weise kann das Löschen eines Attributwertes erreicht werden.

**d) erweiterte Attributmenge**

<b>addAttribute(OM, label, domain)</b>	
Erklärung:	fügt der Attributmenge $EAttributes$ des Organisationsmodells $OM$ ein Attribut mit dem Namen $label$ und dem Wertebereich $domain$ zu.
Vorbedingung:	$label \in Label$ $domain \in Domain$
Nachbedingung:	$EAttributes^* = EAttributes \cup \{(label, domain)\}$
<b>delAttribute(OM, attr)</b>	
Erklärung:	löscht aus dem Organisationsmodell $OM$ das Attribut $attr$ , sofern es nicht zu den Basis-Attributen gehört.
Vorbedingung:	$attr \notin BAttributes$
Nachbedingung:	$EAttributes^* = EAttributes - \{attr\}$

**4.4 Elementare Änderungsoperationen**

Elementare Änderungsoperationen werden benötigt, um organisatorische Entitäten oder Relationen eines Organisationsmodells  $OM \in OM$  zu erzeugen, zu ändern oder zu löschen. Dabei kommen die oben beschriebenen Änderungsprimitive zur Anwendung, so daß auch die Semantik der Elementaroperationen präzise festliegt. Für ihre Anwendung wird die Einhaltung gewisser Vor- und Nachbedingungen gefordert, die sicherstellen sollen, daß im Anschluß wieder ein korrektes Organisationsmodell resultiert. Allerdings lassen sich bestimmte Modelleigenschaften, etwa die in Abschnitt 2.2.1 erklärten Kardinalitätsrestriktionen nur bei kombinierter Anwendung von Elementaroperationen aufrechterhalten. Dieser Aspekt wird an dieser Stelle zunächst ausgeklammert, aber in Abschnitt 4.5 weiterverfolgt.

Vorausgesetzt werden das Organisationsmodell  $OM = (OE, OR, EAttributes, OA, OAV)$  und das zugehörige Organisations-Metamodell  $OMM = (OET, ORT, BAttributes, OTA)$ . Durch die Anwendung der elementaren Änderungsoperationen entsteht ein *weitgehend korrektes* Organisationsmodell  $OM^* \in OM$ .

Die Einhaltung der definierten Restriktionen (1) bis (5) (vgl. Abschnitt 2.2.1.) sowie die semantische Korrektheit der bestehenden Referenzen werden durch die Änderungsoperation sichergestellt. So treten auch keine Typ- oder Namenskonflikte mehr auf. Jedoch kann noch nicht gewährleistet werden, daß bei der Anwendung dieser elementaren Änderungsoperationen auch die Kardinalitätsbeschränkungen (6) für das gegebene Organisationsmodell erfüllt sind. Dies gilt zum Beispiel für das Erzeugen einer neuen Entität, aber auch für die veränderte Zuordnung von Relationen. Diese Beschränkungen werden erst durch die Anwendung von komplexen Operationen bzw. Änderungstransaktionen berücksichtigt.

### 4.4.1 Entitäten

#### a) Entität erzeugen

Um eine Organisations-Entität in einem Organisationsmodell zu erzeugen, wird die elementare Änderungsoperation **createEntity**(OM, entitylabel, entitytype, attr\_value\_set) angewendet. Sie fügt in dem Organisationsmodell *OM* eine Entität mit der Bezeichnung *entitylabel* und dem Typ *entitytype* ein, verknüpft sie mit den für ihren Entitätstyp vorgesehenen Attributen und weist diesen die in der Attributmenge *attrValueSet* enthaltenen Werte zu.

#### **createEntity(OM, entitylabel, entitytype, attrValueSet)**

<b>Input:</b>	OM:	gültiges Organisationsmodell
	entitylabel:	Bezeichner für die zu erzeugende Entität
	entitytype:	Typ der zu erzeugenden Entität
	attrValueSet:	Menge von Attributwertzuordnungen {(attr <sub>0</sub> , value <sub>0</sub> ), ..., (attr <sub>k</sub> , value <sub>k</sub> )} für die zu erzeugende Entität

**begin**

addOrgEntity(OM, entitylabel, entitytype)

**for** i : = 0 to k **do**

addOrgAttr(OM, (entitylabel, entitytype), attr<sub>i</sub>)

setOrgAttrValue(OM, (entitylabel, entitytype), attr<sub>i</sub>, value<sub>i</sub>)

**end**

**end**

Damit diese Elementaroperation korrekt auf das Organisationsmodell OM anwendbar ist, müssen folgende Vorbedingungen erfüllt sein:

- entitylabel ∈ Label
- entitytype ∈ OET
- (entitylabel, entitytype) ∉ OE
- attrValueSet = {(attr<sub>0</sub>, value<sub>0</sub>), ..., (attr<sub>k</sub>, value<sub>k</sub>)}    k ∈ ℕ<sub>0</sub> ⇒  
  {attr<sub>0</sub>, ..., attr<sub>k</sub>} = fOTA(entitytype)
- ∀ (attr<sub>i</sub>, value<sub>i</sub>) ∈ attrValueSet: value<sub>i</sub> ∈ attr<sub>i</sub>.Domain

Wie man leicht zeigen kann, sind bei ihrer Einhaltung auch die Vorbedingungen der verwendeten Änderungsprimitive erfüllt.

Die Nachbedingungen für die Elementaroperation ergeben sich unmittelbar aus der mengenwertigen Semantik der angewendeten Änderungsprimitive:

Sei oe = (entitylabel, entitytype), dann gelte

- OE\* = OE ∪ {(oe)}
- OA\* = OA ∪ {(oe, attr<sub>0</sub>), ..., (oe, attr<sub>k</sub>)}    k ∈ ℕ<sub>0</sub>
- OAV\* = OAV ∪ {(oe, attr<sub>0</sub>, value<sub>0</sub>), ..., (oe, attr<sub>k</sub>, value<sub>k</sub>)}    k ∈ ℕ<sub>0</sub>

Die Kardinalitätsbeschränkungen werden hier noch nicht berücksichtigt.

## b) Entität löschen

Um eine Organisations-Entität aus einem Organisationsmodell zu entfernen, wird die elementare Änderungsoperation **deleteEntity**(OM, oe) angewendet. Sie entfernt aus dem Organisationsmodell *OM* alle Relationen und Entitätsattribute sowie deren Wertzuweisungen, an denen die Entität *oe* beteiligt ist, und löscht anschließend die Entität selbst.

```

deleteEntity(OM, oe)
Input:    OM:          gültiges Organisationsmodell
           oe:          zu löschende Organisations-Entität

begin
  attrdelSet = {oa = (entity, attr) ∈ OA | entity = oe}
  reldelSet = {r = (oe1, oe2, relationtype) ∈ OR | (oe1 = oe) ∨ (oe2 = oe)}
  forall r ∈ reldelSet do
    attrdelSet = attrdelSet ∪ {oa = (relation, attr) ∈ OA | relation = r}
    delOrgRelation(OM, r)
  end
  forall oa ∈ attrdelSet do
    delOrgAttr(OM, oa)10
  end
  delOrgEntity(OM, oe)
end

```

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM, müssen folgende Vorbedingungen erfüllt sein:

- $oe \in OE$

Die Nachbedingungen für die Elementaroperation ergeben sich wiederum aus der Semantik der angewendeten Änderungsprimitive. Die Kardinalitätsbeschränkungen werden noch nicht berücksichtigt.

## 4.4.2 Relationen

### a) Relation erzeugen

Um eine Organisations-Relation zwischen zwei Entitäten in einem Organisationsmodell zu erzeugen, wird die elementare Änderungsoperation **createRelation**(OM, oe1, oe2, relationtype, attrValueSet) angewendet. Sie fügt in dem Organisationsmodell *OM* eine Relation vom Typ *relationtype* zwischen den Entitäten *oe1* und *oe2* ein, verknüpft die Relation mit den für ihren Typ vorgesehenen Attributen und weist diesen die in der Attributmenge *attrValueSet* enthaltenen Werte zu.

<sup>10</sup> Die Änderungsprimitive *delOrgAttr* entfernt auch den Eintrag der Wertzuweisung in OAV (vgl. Abschnitt 4.3.)

**createRelation(OM, oe1, oe2, relationtype, attrValueSet)**

**Input:** OM: gültiges Organisationsmodell  
 oe1, oe2: Organisations-Entitäten  
 relationtype: Typ der zu erzeugenden Relation  
 attrValueSet: Menge von Attributwertzuordnungen  
 $\{(attr_0, value_0), \dots, (attr_k, value_k)\}$  für die zu erzeugende Relation

**begin**

addOrgRelation(OM, oe1, oe2, relationtype)

**for** i := 0 to k **do**addOrgAttr(OM, (oe1, oe2, relationtype), attr<sub>i</sub>)setOrgAttrValue(OM, (oe1, oe2, relationtype), attr<sub>i</sub>, value<sub>i</sub>)**end****end**

Damit diese Elementaroperation korrekt auf das Organisationsmodell OM anwendbar ist, müssen folgende Vorbedingungen erfüllt sein:

- relationtype  $\in$  ORT
- oe1, oe2  $\in$  OE
- (oe1, oe2, relationtype)  $\notin$  OR
- oe1 = (entitylabel1, entitytype1), oe2 = (entitylabel2, entitytype2)  $\Rightarrow$   
 (entitytype1, entitytype2, relationtype)  $\in$  ValidRelation
- attrValueSet =  $\{(attr_0, value_0), \dots, (attr_k, value_k)\}$   $k \in \mathbb{N}_0 \Rightarrow$   
 $\{attr_0, \dots, attr_k\} = f_{OTA}(\text{relationtype})$
- $\forall (attr_i, value_i) \in \text{attrValueSet}: value_i \in attr_i.\text{Domain}$

Die Nachbedingungen für diese Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive. Die Kardinalitätsbeschränkungen werden hier nicht berücksichtigt.

## b) Relation löschen

Um eine Organisations-Relation aus dem Organisationsmodell zu entfernen, wird die elementare Änderungsoperation **deleteRelation**(OM, or) angewendet. Sie entfernt aus dem Organisationsmodell *OM* die Relation *or* sowie deren Relationsattribute und ihre Wertzuweisungen.

### **deleteRelation(OM, or)**

**Input:** OM: gültiges Organisationsmodell  
or: zu löschende Organisations-Relation

#### **begin**

attrdelSet = {oa = (relation, attr) ∈ OA | relation = r}

**forall** oa ∈ attrdelSet **do**

delOrgAttr(OM, oa)

**end**

delOrgRelation(OM, or)

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM, müssen folgende Vorbedingungen erfüllt sein:

- or ∈ OR

Die Nachbedingungen für die Elementaroperation ergeben sich aus der mengenwertigen Semantik der angewendeten Änderungsprimitive. Die Kardinalitätsbeschränkungen werden noch nicht berücksichtigt.

## 4.4.3 Attribute

### a) Attribut definieren

Für bestimmte Anwendungen kann es sinnvoll sein, das konkrete Organisationsmodell beispielsweise um branchenbezogene Attribute zu erweitern.

Um ein neues Attribut in der erweiterten Attributmenge *EAttributes* des Organisationsmodells zu definieren, wird die elementare Änderungsoperation **createAttribute**(OM, label, domain) angewendet. Sie fügt in die Attributmenge *EAttributes* des Organisationsmodells *OM* ein Attribut mit der Bezeichnung *label* und dem Wertebereich *domain* ein.

### **createAttribute(OM, label, domain)**

**Input:** OM: gültiges Organisationsmodell  
label: Bezeichner des zu erzeugenden Attributes  
domain: Wertebereich des zu erzeugenden Attributes

#### **begin**

addAttribute(OM, label, domain)

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM müssen folgende Vorbedingungen erfüllt sein:

- $\text{label} \in \text{Label}$
- $\text{domain} \in \text{Domain}$
- $(\text{label}, \text{domain}) \notin \text{EAttributes}$

Die Nachbedingungen für die Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive.

### b) Attribut entfernen

Es kann auch sinnvoll sein, ein Attribut aus einem speziellen Organisationsmodell zu entfernen, sofern es nicht standardmäßig in der Basis-Attributmenge  $\text{BAttributes}$  des konfigurierten Organisations-Metamodells vorgesehen ist.

Um ein Attribut aus der erweiterten Attributmenge  $\text{EAttributes}$  des Organisationsmodells zu löschen, wird die elementare Änderungsoperation **deleteAttribute**(OM, attr) angewendet. Sie entfernt aus der Attributmenge  $\text{EAttributes}$  des Organisationsmodells  $\text{OM}$  das Attribut  $\text{attr}$ , sofern es weder einer Entität noch einer Relation zugeordnet oder ein Basisattribut aus  $\text{BAttributes}$  ist.

#### **deleteAttribute(OM, attr)**

**Input:** OM: gültiges Organisationsmodell  
 attr: zu löschendes Attribut

**begin**

delAttribute(OM, attr)

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM müssen folgende Vorbedingungen erfüllt sein:

- $\text{attr} \in \text{EAttributes} \neg \text{BAttributes}$
- $\text{oa} = (x, \text{attribute}) \in \text{OA}: \text{attribute} \neq \text{attr}$

Das Attribut darf nicht in der Basis-Attributmenge  $\text{BAttributes}$  enthalten sein. Außerdem darf keinerlei Zuordnung einer Entität oder Relation zu diesem Attribut vorliegen.

Die Nachbedingungen für die Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive.

### c) Attribut zuordnen

Um ein Attribut einer Entität oder Relation des Organisationsmodells zuzuordnen, wird die elementare Änderungsoperation **createAttributeAssignment**(OM, x, attr, value) angewendet. Sie verknüpft das Attribut  $\text{attr}$  mit der Entität oder Relation  $x$  des Organisationsmodells  $\text{OM}$ , indem sie einen Eintrag in  $\text{OA}$  macht, und weist diesem Organisationsattribut den Wert  $\text{value}$  zu.

**createAttributeAssignment(OM, x, attr, value)**

**Input:** OM: gültiges Organisationsmodell  
 x: Organisations-Entität oder Relation  
 attr: Attribut  
 value: Wert

**begin**

addOrgAttr(OM, x, attr)  
 setOrgAttrValue(OM, (x, attr), value)

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM müssen folgende Vorbedingungen erfüllt sein:

- $attr \in EAttributes$
- $x \in OE \cup OR$
- $(x, attr) \notin OA$
- $value \in attr.Domain$

Die Nachbedingungen für die Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive.

**d) Attributzuordnung löschen**

Um eine Attributzuordnung zu einer Entität oder Relation des Organisationsmodells zu löschen, wird die elementare Änderungsoperation **deleteAttributeAssignment(OM, oa)** angewendet. Sie entfernt das Organisations-Attribut *oa* und ggf. seine Wertzuordnungen.

**deleteAttributeAssignment(OM, oa)**

**Input:** OM: gültiges Organisationsmodell  
 oa: Organisations-Attribut

**begin**

delOrgAttr(OM, oa)<sup>11</sup>

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM, muß folgende Vorbedingung erfüllt sein:

- $oa \in OA$

Die Nachbedingungen für die Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive.

<sup>11</sup> Die Änderungsprimitive *delOrgAttr* entfernt auch den Eintrag der Wertzuweisung in OAV (vgl. Abschnitt 4.3.)

**e) changeAttributeValue(OM, oa, value)**

Um den Wert eines Organisations-Attributes einer Entität oder Relation des Organisationsmodells zu ändern, wird die elementare Änderungsoperation **changeAttributeValue(OM, oa, value)** angewendet. Sie ordnet dem Organisations-Attribut *oa* den Wert *value* zu.

**changeAttributeValue(OM, oa, value)**

**Input:** OM: gültiges Organisationsmodell  
 oa: Organisations-Attribut  
 value: Wert

**begin**

setOrgAttrValue(OM, oa, value)

**end**

Für eine korrekte Anwendung dieser Elementaroperation auf das Organisationsmodell OM müssen folgende Vorbedingungen erfüllt sein:

- $oa = (x, attr) \in OA$
- $value \in attr.Domain$

Die Nachbedingungen für die Elementaroperation ergeben sich aus der Semantik der angewendeten Änderungsprimitive.

**4.5 Komplexe Änderungen und Änderungstransaktionen**

*Komplexe Änderungen* eines Organisationsmodells  $OM \in OM$  setzen auf den beschriebenen elementaren Änderungsoperationen auf.

Normalerweise kann eine gewünschte Änderung des Organisationsmodells nicht durch eine einzige Änderungsoperation ausgedrückt werden, sondern erfordert die kombinierte Anwendung mehrerer solcher Elementaroperationen. Häufig benötigte komplexe Änderungen, etwa die Neuordnung einer Stelle zu einer Organisationseinheit (d.h. Entfernen der alten und Festlegen der neuen Zuordnung), sollten dabei durch *vordefinierte Änderungsoperationen* beschreibbar sein.

Unabhängig davon, ob eine komplexe Änderung vordefiniert ist oder nicht, muß ihre Anwendung gewissen Anforderungen genügen: Darauf wird im folgenden Abschnitt Bezug genommen. Abschnitt 4.5.2 geht darauf ein, wie diese Anforderungen möglichst effizient erfüllt werden können. Ausgewählte vordefinierte Änderungsoperationen werden beispielhaft in Abschnitt 4.5.3 behandelt.

**4.5.1 Änderungstransaktionen**

Eine komplexe Änderung umfaßt also eine oder mehrere elementare Änderungsoperationen  $op_1, \dots, op_n$  ( $n \in \mathbb{N}$ ), die ausgehend vom Organisationsmodell OM nacheinander ausgeführt werden. Dabei müssen ähnlich wie bei Transaktionen (Dadam, 1996; S. 185 f) folgende ACID-Eigenschaften gelten (Tabelle 8):

Atomarität	Die Operationen $op_1, \dots, op_n$ ( $n \in \mathbb{N}$ ) werden entweder vollständig oder gar nicht auf OM angewendet.
Konsistenz	Ausgehend von dem konsistenten Organisationsmodell OM muß die Anwendung der Operationen $op_1, \dots, op_n$ ( $n \in \mathbb{N}$ ) wieder zu einem konsistenten Organisationsmodell $OM^*$ führen. Konsistenz ist dabei im Sinne der Einschränkungen (1) bis (6) (vgl. Abschnitt 2.2.1) zu verstehen.
Isolation	Für andere Änderungstransaktionen sind Zwischenzustände, die sich aus der Anwendung einzelner Operationen $op_i$ ergeben, nicht sichtbar.
Dauerhaftigkeit	Die durch die Operationen $op_1, \dots, op_n$ ( $n \in \mathbb{N}$ ) definierten Änderungen sollen nach Beendigung der Änderungstransaktion dauerhaft für das Organisationsmodell OM gelten.

**Tabelle 8**      **Eigenschaften von Änderungstransaktionen**

Solche Änderungstransaktionen werden wie folgt definiert:

**Definition 5 (Änderungstransaktion auf Organisationsmodelle)**

Gegeben sei ein Organisationsmodell  $OM = (OE, OR, EAttributes, OA, OAV)$ .

Eine Änderungstransaktion ist die serielle Anwendung von elementaren Änderungsoperationen  $op_1, \dots, op_n$  ( $n \in \mathbb{N}$ ) auf OM, für die Eigenschaften wie Atomarität, Konsistenz, Isolation und Dauerhaftigkeit gelten.

Im folgenden Abschnitt wird gezeigt, wie sich für diese Änderungstransaktionen Konsistenz sichern läßt.

#### 4.5.2 Konsistenzsicherung einer Änderungstransaktion

Für jede komplexe Änderung eines konsistenten Organisationsmodells OM wird gefordert, daß anschließend wieder ein konsistentes Organisationsmodell  $OM^*$  resultiert. Wie wir später sehen werden, ist die Einhaltung dieser Konsistenzeigenschaften auch für weitergehende Betrachtungen (z.B. Pflege der Cross-Referenzen zwischen Organisationsmodell und Prozeßmodellen) essentiell.

Ohne Beschränkung der Allgemeinheit wird im Folgenden angenommen, daß bei Ausführung einer Änderungstransaktion  $T$  die elementaren Änderungsoperationen  $op_1, \dots, op_n$  ( $n \in \mathbb{N}$ ) in der angegebenen Reihenfolge auf das Ausgangsmodell OM angewendet werden. Dabei ergeben sich für das Organisationsmodell Zwischenzustände, die aufgrund der Isolationseigenschaft von  $T$  für andere Transaktionen nicht sichtbar sind, für die aber gewisse Zusicherungen gemacht werden können. Formal:

$$OM = : OM_1 [op_1 > OM_2 [op_2 > \dots OM_n [op_n > OM_{n+1} := OM^*$$

$OM_i [op_i > OM_{i+1}$  bedeutet dabei, daß  $op_i$  auf  $OM_i$  anwendbar ist und daraus das (temporäre, für andere Transaktionen nicht sichtbare) Modell  $OM_{i+1}$  resultiert. Für

jede Operation  $op_i$  sind dabei die formalen Vorbedingungen für ihre Anwendung auf  $OM_i$  erfüllt, so daß anschließend für  $OM_{i+1}$  jeweils gewisse Nachbedingungen bzw. Korrektheitseigenschaften gelten.

Wie oben diskutiert, lassen sich dadurch bereits die Einschränkungen (1) bis (5) (vgl. Abschnitt 2.2.1) bei Ausführung einzelner Elementaroperationen sicherstellen. Dies kann für die Kardinalitätsbeschränkung (6) des Organisationsmodells noch nicht garantiert werden. Die Kardinalitätsrestriktionen können im allgemeinen erst durch die zusammenhängender Anwendung einer *Menge* von Elementaroperationen aufrechterhalten werden.

Bezogen auf eine Änderungstransaktion  $T$  bedeutet das, daß für einen Zwischenzustand  $OM_i$  die Einschränkung (6) nicht gewährleistet ist. Aufgrund der Isolationseigenschaft der Transaktion ist das auch nicht erforderlich. In jedem Falle aber muß das resultierende Organisationsmodell  $OM^*$  die Kardinalitätsbeschränkungen erfüllen, um für die gesamte Transaktion Konsistenz zu gewährleisten.

Da die Einschränkungen (1) bis (5) (vgl. Abschnitt 2.2.1) bereits bei Anwendung der Elementaroperationen  $op_1, \dots, op_n$  ( $n \in \mathbb{N}$ ) eingehalten werden, muß anschließend nur noch überprüft werden, ob die Kardinalitätsbeschränkungen für das neu erzeugte (nach außen noch nicht sichtbare) Organisationsmodell  $OM^*$  erfüllt werden.

Falls das zutrifft, kann die Änderungstransaktion ausgeführt werden (*Commit*), so daß die Effekte auch für andere Transaktionen sichtbar gemacht werden. Im anderen Fall wird die Transaktion mit entsprechenden Fehlermeldungen abgebrochen.

Es gibt mehrere Möglichkeiten zur Konsistenzsicherung bei Anwendung einer Änderungstransaktion.

Ein naiver Ansatz wäre es, vor Commit der Änderungstransaktion für alle Entitäten des Organisationsmodells  $OM^*$  zu überprüfen, ob ihre Kardinalitätsbeschränkungen erfüllt sind. Da eine Änderungstransaktion in der Regel aber nur eine kleine Teilmenge der Entitäten und Beziehungen des Gesamtmodells verändert, ist dieser Ansatz zu aufwendig und scheidet deshalb aus.

Ein effizienteres Verfahren überprüft die Kardinalitätsbeschränkungen nur für diejenigen Entitäten bzw. Relationen, die durch die einzelnen Änderungsoperationen tatsächlich modifiziert wurden. Konkret betrifft das Entitäten, die neu hinzugefügt bzw. für die Relationen neu eingefügt oder gelöscht worden sind. Für gelöschte Entitäten müssen trivialerweise keine Kardinalitätsbeschränkungen überprüft werden.

Im folgenden wird ein weiterführender Ansatz vorgestellt. Bei diesem optimierten Verfahren werden bei Anwendung einzelner Elementaroperationen Informationen über die vorgenommenen Änderungen "gesammelt". Dabei wird auch berücksichtigt, daß bestimmte Änderungsoperationen sich neutralisieren, so daß temporäre Verletzungen der Kardinalitätsbeschränkungen wieder rückgängig gemacht werden. Demzufolge müssen nur noch die Kardinalitätsbedingungen für die Entitäten

überprüft werden, welche durch Anwendung der Transaktion entweder neu hinzugekommen sind oder für die sich die Anzahl der Beziehungen zu anderen Entitäten geändert hat.

Für diesen Ansatz werden nun folgende Mengen und Funktionen definiert (Tabelle 9):

CreatedEntities	Menge aller im Verlauf der Änderungstransaktion erzeugten Entitäten
DeletedEntities	Menge aller im Verlauf der Änderungstransaktion gelöschten Entitäten
ChangedRelations	Enthält für jede Entität, für die eine Relation eingefügt oder gelöscht wurde, ein Tupel bestehend aus dem Namen der Entität, ihrer Position in der Relation ( <i>left</i> oder <i>right</i> ) und dem Relationstyp. Es gilt: $\text{ChangedRelations} \subseteq (\text{OE}^* \cup \text{DeletedEntities}) \times \{\textit{left}, \textit{right}\} \times \text{Relationtype}$
counter	Gibt für jedes Element aus der Menge ChangedRelations an, wieviel Relationen dieser Art insgesamt hinzugekommen oder gelöscht worden sind. Es gilt: $\text{counter}: \text{ChangedRelations} \rightarrow \mathbb{Z}$

**Tabelle 9 Mengen und Funktionen für Algorithmus zur Konsistenzsicherung**

Zu Beginn einer Änderungstransaktion sind diese Mengen leer, der Zähler *counter* wird für jedes neue Element aus ChangedRelation zunächst mit Null initialisiert. Jede in das Modell neu eingefügte Entität wird in *CreatedEntities* hinzugefügt, jede gelöschte Entität in *DeletedEntities*. Wenn eine Relation zwischen zwei Entitäten eingefügt wird, erhöhen sich die Zähler der entsprechenden Einträge in ChangedRelations um Eins. Existiert noch kein solcher Eintrag, wird er zuvor erzeugt. Wird eine Relation gelöscht, werden die Zähler der entsprechende Einträge um Eins erniedrigt. Der Zähler zeigt also an, wie sich die Anzahl von Relationen eines bestimmten Relationstypes mit einer bestimmten Entität in der linken bzw. rechten Position der Relation vom Ausgangsmodell unterscheidet. Bei einem positiven Zähler sind demnach bei Anwendung der Änderungstransaktion mehr Relationen hinzugekommen als gelöscht worden, bei einem negativen Zähler verhält es sich umgekehrt.

#### **Beispiel 4: (Einhaltung der Kardinalitätsbeschränkung)**

Szenario: Stationsarzt Dr. Meier verläßt die Klinik. Frau Dr. Müller wird eingestellt und übernimmt diese Stelle.

Erklärung: Mitarbeiter Dr. Meier wird gelöscht, es erfolgt ein Eintrag in DeletedEntities. Die Relation ((*Dr. Meier, Mitarbeiter*), (*Stationsarzt, Stelle*), *hat*) wird ebenfalls gelöscht, also erfolgen in ChangedRelations die Einträge ((*Dr. Meier, Mitarbeiter*), *left, hat*) und ((*Stationsarzt, Stelle*), *right, hat*) mit den Zählern  $\text{counter}((\textit{Dr. Meier, Mitarbeiter}), \textit{left, hat}) = -1$  und  $\text{counter}((\textit{Stationsarzt, Stelle}), \textit{right, hat}) = -1$ .

Die neue Mitarbeiterin Frau Dr. Müller wird in das Modell eingefügt, es erfolgt ein Eintrag in CreatedEntities. Außerdem wird die Relation

((Dr. Müller, Mitarbeiter), (Stationsarzt, Stelle), hat) eingefügt und damit die Einträge ((Dr. Müller, Mitarbeiter), left, hat) und ((Stationsarzt, Stelle), right, hat) mit den Zählern  $\text{counter}((\text{Dr. Müller, Mitarbeiter}), \text{left}, \text{hat}) = 1$  und  $\text{counter}((\text{Stationsarzt, Stelle}), \text{right}, \text{hat}) = (-1 + 1) = 0$ .

Der mitgeführte Zähler dient letztlich als Kriterium, um effizient entscheiden zu können, ob die geforderten Kardinalitätsbeschränkungen eingehalten werden. Der genaue Algorithmus wird im Folgenden dargestellt.

Wie sich die Mengen CreatedEntities, DeletedEntities, ChangedRelations und  $\text{counter}(\text{ChangedRelations})$  bei der Anwendung einer Änderungstransaktion bzw. einer Elementaroperation bestimmen lassen, wird in Anhang A gezeigt.

Die Prüffunktion *checkCardinality* an (Algorithmus 1) überprüft nun effizient, ob die geforderten Kardinalitätsbeschränkungen auch für das aus der Transaktion hervorgehende Organisationsmodell  $\text{OM}^*$  erfüllt sind. Wie erwähnt, basiert die Überprüfung auf dem bisherigen Organisationsmodell  $\text{OM}$ , sowie auf den durchgeführten Änderungen, die durch die Mengen CreatedEntities, CreatedRelations und ChangedRelations sowie die Funktion counter beschrieben sind. Das Organisationsmodell  $\text{OM}$  selbst ist nach Voraussetzung korrekt und erfüllt damit alle Kardinalitätsbeschränkungen. Das Ergebnis der Prüffunktion *checkCardinality* ist die Aussage, ob alle Restriktionen erfüllt sind oder nicht (ggf. erfolgen genauere Diagnoseangaben für den Modellierer).

#### Algorithmus 1 (Überprüfung der Kardinalitätsbeschränkung)

```

boolean checkCardinality(OM, CreatedEntities, CreatedRelations, ChangedRelations,
    counter(ChangedRelations))
begin
    correct := TRUE

    /* Teil 1: Überprüfen der Kardinalitätsbeschränkungen für neu eingeführte Entitäten */
    /* Ermitteln aller relevanten Beziehungstypen für oe */
    forall oe  $\in$  CreatedEntities  $\neg$  DeletedEntities do
        RelevantRelation = {rt = (entitytype1, entitytype2, relationtype)  $\in$  ValidRelations |
            oe.entitytype  $\in$  {entitytype1, entitytype2} }
        forall rt = (entitytype1, entitytype2, relationtype)  $\in$  RelevantRelation do
            (min1, max1, min2, max2) := cardinality(rt)      /* vgl. Abschnitt 2.1.1 */

    /* Überprüfen der Relationen mit oe in linker Position; relevant falls mini > 0 oder maxi < ∞ */
    if oe.entitytype = entitytype1 and (min1, max1)  $\neq$  (0, ∞) then
        if r = (oe, left, relationtype)  $\in$  ChangedRelations then
            #relations := counter(r)
        else
            #relations := 0
        endif
        if #relations < min1 or #relations > max1 then
            correct := FALSE      /* zzgl. Benutzerausgabe */
        endif
    endif
endif

```

```

/* Überprüfen der Relationen mit oe in rechter Position */
if oe.entitytype = entitytype2 and (min2, max2) ≠ (0, ∞) then
  if r = (oe, right, relationtype) ∈ ChangedRelations then
    #relations := counter(r)
  else
    #relations := 0
  endif
  if #relations < min2 or #relations > max2 then
    correct := FALSE          /* zzgl. Benutzerausgabe */
  endif
endif
done
done

/* Teil 2: Überprüfung der Kardinalitätsbeschränkungen für Entitäten, für die Relationen
hinzugekommen oder gelöscht worden sind, und die in Teil 1 noch nicht untersucht wurden. Sie erfolgt
auf Grundlage der Menge ChangedRelations und der Funktion counter. Wegen der Konstruktion
dieser beiden Größen müssen nur solche Eintragungen in ChangedRelations betrachtet werden, für die
counter1 0 gilt. */

forall cr = (oe, x, relationtype) ∈ ChangedRelations with
  (oe ∉ CreatedEntities ∪ DeletedEntities) and counter(r) ≠ 0 do
  rt := zu cr korrespondierender Beziehungstyp aus ValidRelation
  (min1, max1, min2, max2) := cardinality(rt)

  /* Entität in der linken Position der Beziehung; relevant falls min1 > 0 oder max1 < ∞ */
  if x = left and (min1, max1) ≠ (0, ∞) then
    /* Ermittelt die Anzahl der Beziehungen, die bisher für das Ausgangsmodell
    OM= (OE, OR, EAttributes, OA, OAV) gültig waren */
    current_card := |{(oe1, oe2, reltype) ∈ OR | oe1 = oe ∧ reltype = relationtype }|
    current_card := current_card + counter(cr) /* aktualisiert Anzahl der Beziehungen */
    if (counter(cr) > 0 and current_card > max1) or
      (counter(cr) < 0 and current_card < min1) then
      correct := FALSE          /* zzgl. Benutzerausgabe */
    endif

    /* Entität in der rechten Position der Beziehung; Überprüfung wie oben mit min2 und max2 */
    else
      if x = right and (min2, max2) ≠ (0, ∞) then
        ...
      endif
    endif
  done
return correct          /* ggf. Benutzerausgabe des Diagnosesystems */

```

### 4.5.3 Vordefinierte komplexe Änderungen

Bestimmte (komplexe) Änderungen von Organisationen kommen in der Praxis häufiger vor. Beispielsweise können einzelne Abteilungen zusammengelegt oder geteilt werden, Stellen anderen Abteilungen zugeordnet werden, usw.

Um diese Änderungen im Organisationsmodell durchzuführen, kommen wiederholt Änderungstransaktionen zur Anwendung, die ganz bestimmte Sequenzen von Elementaroperationen enthalten. Deshalb ist es sinnvoll, diese häufig verwendeten Änderungstransaktionen vorzudefinieren. Relevant sind dafür die *komplexen* Änderungen *Vereinigen* (*Join*), *Teilen* (*Split*) und *Verschieben* (*Move*) von Entitäten.

Zunächst muß geprüft werden, ob sie für den jeweiligen Entitätstyp semantisch sinnvoll sind. So lassen sich beispielsweise Mitarbeiter weder teilen noch zusammenlegen, sehr wohl aber auf eine andere Stelle verschieben (versetzen). Dagegen ist es nicht sinnvoll, eine Arbeitsgruppe (zum Beispiel ein Projekt) im Organisationsmodell zu verschieben (vgl. Organisations-Metamodell in Abschnitt 2.1.1). Tabelle 10 zeigt eine Übersicht dieser komplexen Änderungen in bezug auf den betroffenen Entitätstyp.

Entitätstyp	Vereinigen (Join)	Teilen (Split)	Verschieben (Move) <sup>12</sup>
Arbeitsgruppe (AG)	x	x	
Aufgabe (A)	x	x	
Aufgabenkategorie (AK)	x	x	
Fähigkeiten (F)	x	x	
Mitarbeiter (MA)	-	-	x
Organisationseinheit (OE)	x	x	x
Organisationsgruppe (OG)	x	x	
Rolle (R)	x	x	
Stelle (S)	x	x	x

**Tabelle 10** Entitätstypen und ihre komplexen Änderungen  
(x semantisch sinnvoll)

Wird eine Entität in dieser Weise verändert, müssen alle zugehörigen Relationen<sup>13</sup> angepaßt werden. Bestimmte Relationsanpassungen können *automatisch*, andere müssen *manuell* durch den Modellierer durchgeführt werden.

<sup>12</sup> Verschieben der Entität bedeutet, daß ihre Beziehung vom Typ (OE, OG, gehört\_zu), (S, OE, gehört\_zu), (S, AG, gehört\_zu) bzw. (MA, S, hat) geändert wird. (**fett** geänderte Entität)

<sup>13</sup> Neben den Relationen müssen auch alle Attribtzuordnungen der Entität und deren Werte angepaßt werden.

**Beispiel 5a (Arbeitsgruppen vereinigen)**

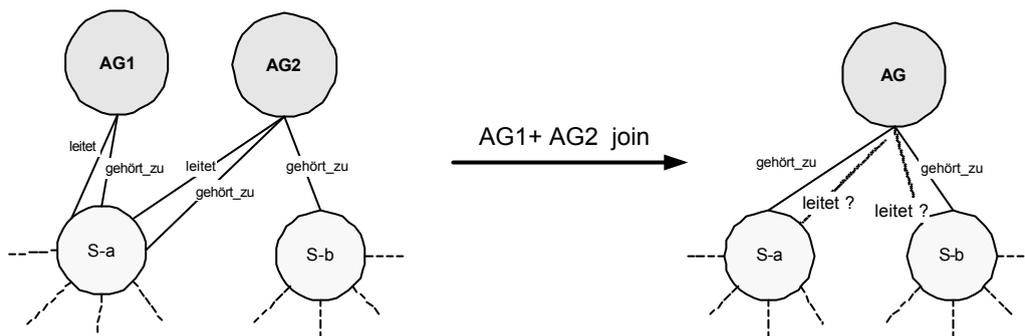
Die Arbeitsgruppen *AG1* und *AG2* sollen zusammengelegt werden.

Zu *AG1* gehört die Stelle *S-a*, sie leitet auch *AG1*.

Zu *AG2* gehören sowohl *S-a* als auch *S-b*, sie wird von *S-a* geleitet.

Die resultierende Arbeitsgruppe *AG* beinhaltet sinnvollerweise alle beide Stellen *S-a* und *S-b*. Diese Anpassung kann automatisch erfolgen.

Problem: Welche Stelle soll nun die resultierende Arbeitsgruppe *AG* leiten? Dafür ist nur eine Stelle zuge lassen. Diese Frage kann also nicht automatisch entschieden werden, sondern muß vom Benutzer geklärt werden. (Abbildung 6)



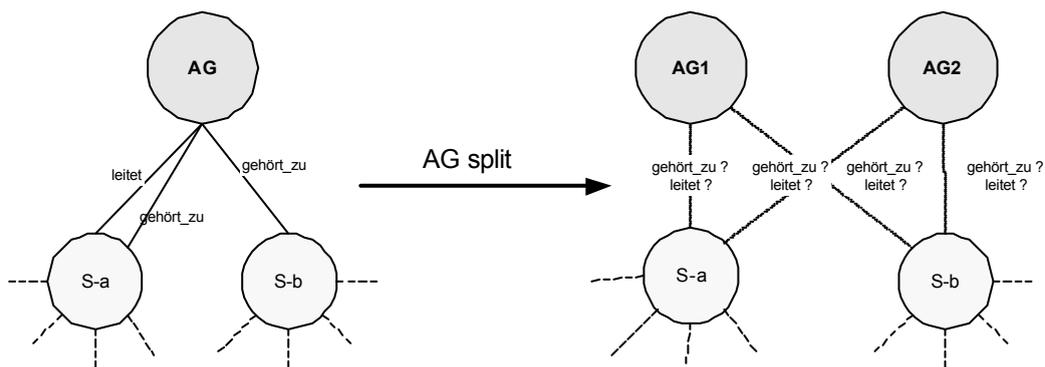
**Abbildung 6 Beispiel 5a: Arbeitsgruppen vereinigen**  
**Wer leitet die neue Arbeitsgruppe?**

**Beispiel 5b (Arbeitsgruppe teilen)**

Die Arbeitsgruppe *AG* soll in *AG1* und *AG2* geteilt werden.

*AG* beinhaltet die zwei Stellen *S-a* und *S-b* und wird von *S-a* geleitet.

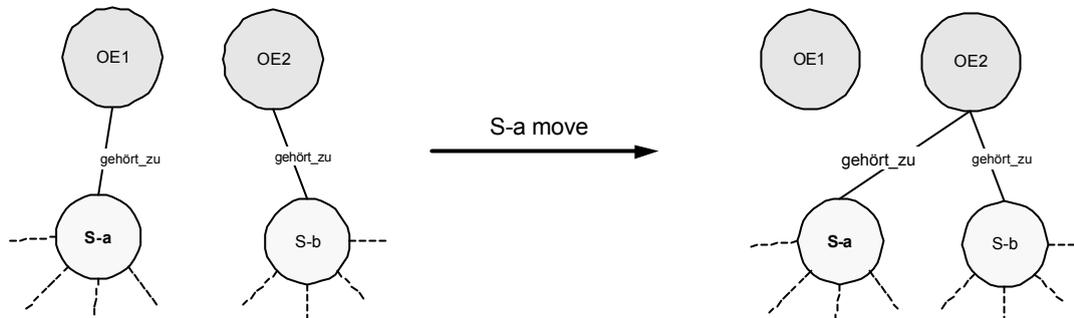
Problem: Welche Stellen gehören in Zukunft zu *AG1*, welche zu *AG2*? Wer leitet *AG1*, wer *AG2*? (Abbildung 7)



**Abbildung 7 Beispiel 5b: Arbeitsgruppe teilen**  
**Welche Stelle gehört zu welcher neuen Arbeitsgruppe? Wer leitet diese?**

**Beispiel 5c (Stelle verschieben)**

Die Stelle *S* soll von der Organisationseinheit *OE1* zu der Organisationseinheit *OE2* verschoben werden. Dafür muß lediglich die Beziehung (*OE1*, *S*, besteht\_aus) in (*OE2*, *S*, besteht\_aus) geändert werden (Abbildung 8).



**Abbildung 8** Beispiel 5c: Stelle anderer Organisationseinheit zuordnen

Die Anpassung der Relationen ist unter Beachtung der Semantik vom Typ der Änderung (Join, Split) sowie vom Beziehungstyp abhängig (vgl. Übersicht in Anhang B). Nach Verschiebungen können die Relationen dagegen immer *automatisch* angepaßt werden.

Aus dem Anpassungsbedarf der konkreten Beziehungstypen nach Split- und Join-Operationen lassen sich bestimmte Muster ableiten, anhand derer schnell entschieden werden kann, ob eine Relation eines bestimmten Typs automatisch oder manuell durch den Modellierer angepaßt werden kann bzw. muß (Tabelle 11).

<b>SPLIT</b>			
Beziehungstyp	standardmäßige Anpassung		Ausnahmen
	Entität LINKS*	Entität RECHTS	
gehört_zu	manuell	manuell	(S, OE, gehört_zu) wegen Kardinalität automatisch anpaßbar
leitet	manuell	manuell	
ist_übergeordnet / generalisiert	manuell	automatisch	
besitzt / hat	manuell	automatisch	
<b>JOIN</b>			
Beziehungstyp	standardmäßige Anpassung		Ausnahmen
	Entität LINKS	Entität RECHTS	
gehört_zu	automatisch	automatisch	(OE, OG, gehört_zu) wegen Semantik manuell anzupassen
leitet	automatisch	manuell	
ist_übergeordnet / generalisiert	automatisch	automatisch	
besitzt / hat	automatisch	automatisch	
<b>fett:</b> geänderte Entität * Entität in linker bzw. rechter Position der Relation			

**Tabelle 11** Muster der Anpassung von Relationen nach Split- oder Join-Operationen

Bei der *manuellen Anpassung* muß der Modellierer explizit angeben, wie die Relationen behandelt werden sollen. Unter bestimmten Umständen kann bei Join-Operationen die eigentlich erforderliche manuelle Anpassung auch automatisch durchgeführt werden:

Beispielsweise kann für Relationen vom Beziehungstyp (**OE**, OG, gehört\_zu) die Anpassung nach einer *Join-Operation* automatisch erfolgen, falls die zu vereinenden Organisationseinheiten ursprünglich zur gleichen Organisationsgruppe gehörten.

Analog: (OE', **OE**, übergeordnet), (S, AG, gehört\_zu).<sup>14</sup>

Im folgenden werden exemplarisch die vordefinierten komplexen Änderungen *Split* und *Join* von Entitäten des Typs Arbeitsgruppe sowie die Veränderung der Zuordnung (*Move*) einer Stelle zu einer Organisationseinheit dargestellt.

### a) Arbeitsgruppen vereinigen

Um zwei Arbeitsgruppen in einem Organisationsmodell zu vereinigen, wird die vordefinierte Änderungsoperation `joinAG(OM, ag1, ag2, newlabel, leiter, attrValueSetAG, attrValueSetrelAG, attrValueSetleitAG)` angewendet. Dabei werden die Beziehungen des Typs (Stelle, Arbeitsgruppe, gehört\_zu) automatisch angepaßt, während für Beziehungen des Typs (Stelle, Arbeitsgruppe, leitet) explizit vom Modellierer angegeben werden muß, welche Stelle *leiter* die Arbeitsgruppe leiten soll, da dafür maximal eine Stelle zugelassen ist. (Vgl. Beispiel 5a)

**joinAG(OM, ag1, ag2, newlabel, leiter, attrValueSet<sup>AG</sup>, attrValueSet<sup>relAG</sup>, attrValueSet<sup>leitAG</sup>)**

**Input:**

OM:	gültiges Organisationsmodell
ag1, ag2:	zu vereinigende Arbeitsgruppen
newlabel:	neuer Bezeichner für vereinigte Arbeitsgruppe
leiter:	Stelle, die vereinigte Arbeitsgruppe leiten wird
attrValueSet <sup>AG</sup> :	Attributmenge für resultierende Arbeitsgruppe
attrValueSet <sup>relAG</sup> :	Attributmenge für gehört_zu-Beziehungen
attrValueSet <sup>leitAG</sup> :	Attributmenge für leitet-Beziehung

#### **begin**

```

createEntity(OM, newlabel, Arbeitsgruppe, attrValueSetAG)
ag := (newlabel, Arbeitsgruppe)
relSetAG1 := {r = (s1, ag1, gehört_zu) ∈ OR}
  forall r ∈ relSetAG1 do
    createRelation(OM, s1, ag, gehört_zu, attrValueSetrelAG)
    deleteRelation(OM, r)
  done
relSetAG2 = {r = (s2, ag2, gehört_zu) ∈ OR}
  forall r ∈ relSetAG2 do
    createRelation(OM, s2, ag, gehört_zu, attrValueSetrelAG)
    deleteRelation(OM, r)

```

<sup>14</sup> **fett** die veränderte Entität

```

done
deleteRelation(OM, (s, ag1, leitet))
deleteRelation(OM, (s, ag2, leitet))
if leiter  $\neq$  Null then                                     /** Kardinalität: max. ein Leiter */
    createRelation(OM, leiter, ag, leitet, attrValueSetleitAG)
endif
deleteEntity(OM, ag1)
deleteEntity(OM, ag2)
done

```

Damit diese vordefinierte Änderungstransaktion korrekt auf das Organisationsmodell OM anwendbar ist, müssen folgende Vorbedingungen erfüllt sein:

- newlabel  $\in$  Label
- leiter = (label, Stelle), ag1, ag2  $\in$  OE
- (newlabel, Arbeitsgruppe)  $\notin$  OE
- attrValueSet<sup>AG</sup> = {(attr<sub>0</sub>, value<sub>0</sub>), ..., (attr<sub>k</sub>, value<sub>k</sub>)}      k  $\in$   $\mathbb{N}_0 \Rightarrow$   
     {attr<sub>0</sub>, ..., attr<sub>k</sub>} = fOTA(Arbeitsgruppe)
- $\forall$  (attr<sub>i</sub>, value<sub>i</sub>)  $\in$  attrValueSet<sup>AG</sup>: value<sub>i</sub>  $\in$  attr<sub>i</sub>.Domain
- attrValueSet<sup>relAG</sup> = {(attr<sub>0</sub>, value<sub>0</sub>), ..., (attr<sub>m</sub>, value<sub>m</sub>)}      m  $\in$   $\mathbb{N}_0 \Rightarrow$   
     {attr<sub>0</sub>, ..., attr<sub>m</sub>} = fOTA(gehört\_zu)
- $\forall$  (attr<sub>j</sub>, value<sub>j</sub>)  $\in$  attrValueSet<sup>relAG</sup>: value<sub>j</sub>  $\in$  attr<sub>j</sub>.Domain
- attrValueSet<sup>leitAG</sup> = {(attr<sub>0</sub>, value<sub>0</sub>), ..., (attr<sub>n</sub>, value<sub>n</sub>)}      n  $\in$   $\mathbb{N}_0 \Rightarrow$   
     {attr<sub>0</sub>, ..., attr<sub>n</sub>} = fOTA(leitet)
- $\forall$  (attr<sub>h</sub>, value<sub>h</sub>)  $\in$  attrValueSet<sup>leitAG</sup>: value<sub>h</sub>  $\in$  attr<sub>h</sub>.Domain

Die Nachbedingungen ergeben sich wiederum aus der Semantik der angewendeten Elementaroperationen:

- OE\* = OE  $\cup$  {ag}  $\setminus$  {ag1, ag2}
- OR\* = OR  $\cup$  {(leiter, ag, leitet)}  $\cup$  relSetAG  $\setminus$  relSetAG1  $\setminus$  relSetAG2  $\setminus$   
     {(s<sub>1</sub>, ag1, leitet)}  $\setminus$  {(s<sub>2</sub>, ag2, leitet)}  
     mit relSetAG = {r = (s, ag, gehört\_zu)  $\in$  OR}
- OA\* = OA  $\cup$  {(ag, attr<sub>0</sub>), ..., (ag, attr<sub>k</sub>)}  $\setminus$  {(ag1, attr<sub>0</sub>), ..., (ag1, attr<sub>l</sub>)}  $\setminus$   
     {(ag2, attr<sub>0</sub>), ..., (ag2, attr<sub>m</sub>)}  $\cup$  {(r, attr<sub>0</sub>), ..., (r, attr<sub>n</sub>)}  $\setminus$  {(r1, attr<sub>0</sub>), ...,
- (r1, attr<sub>o</sub>)}  $\setminus$  {(r2, attr<sub>0</sub>), ..., (r2, attr<sub>p</sub>)}  $\cup$  {(lt, attr<sub>0</sub>), ..., (lt, attr<sub>q</sub>)}  $\setminus$  {(lt1,
- attr<sub>0</sub>), ..., (lt1, attr<sub>r</sub>)}  $\setminus$  {(lt2, attr<sub>0</sub>), ..., (lt2, attr<sub>s</sub>)}      k, l, m, n, o, p, q,  
     r, s  $\in$   $\mathbb{N}_0$   
     mit r1  $\in$  relSetAG1, r2  $\in$  relSetAG2,  
     lt = (s, ag, leitet), lt1 = (s, ag1, leitet), lt2 = (s, ag1, leitet)
- OAV\* = OAV  $\cup$  {(ag, attr<sub>0</sub>, value<sub>0</sub>), ... analog OA\*}

**b) Arbeitsgruppe teilen**

Um eine Arbeitsgruppe eines Organisationsmodells in zwei Arbeitsgruppen zu teilen, wird die vordefinierte Änderungsoperation  $\text{splitAG}(\text{OM}, \text{ag}, \text{newlabel1}, \text{newlabel2}, \text{setSAG1}, \text{setSAG2}, \text{leiter1}, \text{leiter2}, \text{attrValueSet}^{\text{AG1}}, \text{attrValueSet}^{\text{AG2}}, \text{attrValueSet}^{\text{relAG1}}, \text{attrValueSet}^{\text{relAG2}}, \text{attrValueSet}^{\text{leitAG1}}, \text{attrValueSet}^{\text{leitAG2}})$  angewendet.

Hier muß explizit als Parameter angegeben werden, welche Stellen jeweils zu den gesplitteten Arbeitsgruppen gehören, wer diese leitet sowie alle zugehörigen Attribute. (Vgl. Beispiel 5b)

**splitAG(OM, ag, newlabel1, newlabel2, setSAG1, setSAG2, leiter1, leiter2, attrValueSet<sup>AG1</sup>, attrValueSet<sup>AG2</sup>, attrValueSet<sup>relAG1</sup>, attrValueSet<sup>relAG2</sup>, attrValueSet<sup>leitAG1</sup>, attrValueSet<sup>leitAG2</sup>)**

**Input:** OM: gültiges Organisationsmodell  
 ag: zu teilende Arbeitsgruppe  
 newlabel1, newlabel2: Bezeichner für gesplittete Arbeitsgruppen  
 setSAG1, setSAG2: Stellen, die zu den Arbeitsgruppen gehören  
 leiter1, leiter2: Stellen, die gesplittete Arbeitsgruppen leiten werden  
 attrValueSet<sup>AG1</sup>, attrValueSet<sup>AG2</sup>: Attributmenge für resultierende Arbeitsgruppen  
 attrValueSet<sup>relAG1</sup>, attrValueSet<sup>relAG2</sup>, attrValueSet<sup>leitAG1</sup>, attrValueSet<sup>leitAG2</sup>:  
 Attributmenge für angepaßte Beziehungen

**begin**

```

createEntity(OM, newlabel1, Arbeitsgruppe, attrValueSetAG1)
ag1 := (newlabel1, Arbeitsgruppe)
createEntity(OM, newlabel2, Arbeitsgruppe, attrValueSetAG2)
ag2 := (newlabel2, Arbeitsgruppe)
relSetAG := {r = (s, ag, gehört_zu) ∈ OR}
forall r ∈ relSetAG do
  if s ∈ setSAG1 then
    createRelation(OM, s, ag1, gehört_zu, attrValueSetrelAG1)
  endif
  if s ∈ setSAG2 then
    createRelation(OM, s, ag2, gehört_zu, attrValueSetrelAG2)
  endif
  deleteRelation(OM, r)
done
deleteRelation(OM, (s, ag, leitet))
if leiter1 ≠ Null then
  createRelation(OM, leiter1, ag1, leitet, attrValueSetleitAG1)
endif
if leiter2 ≠ Null then
  createRelation(OM, leiter2, ag2, leitet, attrValueSetleitAG2)
endif
deleteEntity(OM, ag)

```

**done**

Damit diese vordefinierte Änderungstransaktion korrekt auf das Organisationsmodell OM anwendbar ist, müssen folgende Vorbedingungen erfüllt sein:

- $\text{newlabel1}, \text{newlabel2} \in \text{Label}$
- $(\text{newlabel1}, \text{Arbeitsgruppe}), (\text{newlabel2}, \text{Arbeitsgruppe}) \notin \text{OE}$
- $\text{leiter1} = (\text{label1}, \text{Stelle}), \text{leiter2} = (\text{label2}, \text{Stelle}), \text{ag} \in \text{OE}$
- $\text{setSAG1}, \text{setSAG2} \subseteq \text{OE}$  mit  $\text{entitytype} = \text{Stelle}$
- $\text{attrValueSet}^x = \{(\text{attr}_0, \text{value}_0), \dots, (\text{attr}_k, \text{value}_k)\} \quad k \in \mathbb{N}_0 \Rightarrow$   
 $\{\text{attr}_0, \dots, \text{attr}_k\} = f\text{OTA}(\text{Arbeitsgruppe}) \quad \text{mit } x \in \{\text{AG1}, \text{AG2}\}$
- $\forall (\text{attr}_i, \text{value}_i) \in \text{attrValueSet}^x: \text{value}_i \in \text{attr}_i.\text{Domain} \quad \text{mit } x \in \{\text{AG1}, \text{AG2}\}$
- $\text{attrValueSet}^y = \{(\text{attr}_0, \text{value}_0), \dots, (\text{attr}_m, \text{value}_m)\} \quad m \in \mathbb{N}_0 \Rightarrow$   
 $\{\text{attr}_0, \dots, \text{attr}_m\} = f\text{OTA}(\text{gehört\_zu}) \quad \text{mit } y \in \{\text{relAG1}, \text{relAG2}\}$
- $\forall (\text{attr}_j, \text{value}_j) \in \text{attrValueSet}^y: \text{value}_j \in \text{attr}_j.\text{Domain}$   
 $\text{mit } y \in \{\text{relAG1}, \text{relAG2}\}$
- $\text{attrValueSet}^z = \{(\text{attr}_0, \text{value}_0), \dots, (\text{attr}_n, \text{value}_n)\} \quad n \in \mathbb{N}_0 \Rightarrow$   
 $\{\text{attr}_0, \dots, \text{attr}_n\} = f\text{OTA}(\text{leitet}) \quad \text{mit } z \in \{\text{leitAG1}, \text{leitAG2}\}$
- $\forall (\text{attr}_h, \text{value}_h) \in \text{attrValueSet}^z: \text{value}_h \in \text{attr}_h.\text{Domain}$   
 $\text{mit } z \in \{\text{leitAG1}, \text{leitAG2}\}$

Die Nachbedingungen für die Änderungstransaktion ergeben sich wiederum aus der Semantik der angewendeten Elementaroperationen.

- $\text{OE}^* = \text{OE} \cup \{\text{ag1}, \text{ag2}\} \setminus \{\text{ag}\}$
- $\text{OR}^* = \text{OR} \cup \text{relSetAG1} \cup \text{relSetAG2} \cup \{(s_m, \text{ag1}, \text{leitet})\} \cup \{(s_n, \text{ag2}, \text{leitet})\}$   
 $\setminus \text{relSetAG} \setminus \{(\text{leiter}, \text{ag}, \text{leitet})\} \quad (0 \leq i, j \leq k; i, j, k, m, n \in \mathbb{N}_0)$   
 $\text{mit } \text{relSetAG1} = \{r1 = (s, \text{ag1}, \text{gehört\_zu}) \in \text{OR}\},$   
 $\text{relSetAG2} = \{r2 = (s, \text{ag2}, \text{gehört\_zu}) \in \text{OR}\}$
- $\text{OA}^* = \text{OA} \cup \{(\text{ag1}, \text{attr}_0), \dots, (\text{ag1}, \text{attr}_k)\} \cup \{(\text{ag2}, \text{attr}_0), \dots, (\text{ag2}, \text{attr}_l)\} \setminus$   
 $\{(\text{ag}, \text{attr}_0), \dots, (\text{ag}, \text{attr}_m)\} \cup \{(r1, \text{attr}_0), \dots, (r1, \text{attr}_n)\} \cup \{(r2, \text{attr}_0), \dots,$   
 $(r2, \text{attr}_o)\} \setminus \{(r, \text{attr}_0), \dots, (r, \text{attr}_p)\} \cup \{(lt1, \text{attr}_0), \dots, (lt1, \text{attr}_q)\} \cup \{(lt2,$   
 $\text{attr}_0), \dots, (lt2, \text{attr}_r)\} \setminus \{(lt, \text{attr}_0), \dots, (lt, \text{attr}_s)\} \quad k, l, m, n, o, p, q, r, s \in \mathbb{N}_0$   
 $\text{mit } r1 \in \text{relSetAG1}, r2 \in \text{relSetAG2},$   
 $lt = (s, \text{ag}, \text{leitet}), lt1 = (s, \text{ag1}, \text{leitet}), lt2 = (s, \text{ag1}, \text{leitet})$
- $\text{OAV}^* = \text{OAV} \cup \{(\text{ag1}, \text{attr}_0, \text{value}_0), \dots \quad \text{analog } \text{OA}^*$

### c) Stelle zu anderer Organisationseinheit verschieben

Um eine Stelle eines Organisationsmodells einer anderen Organisationseinheit zuzuordnen, wird die vordefinierte Änderungsoperation  $\text{moveS\_OE}(\text{OM}, s, \text{oe}, \text{attrValueSet})$  angewendet. Die Anpassung der Relationen erfolgt automatisch. (Vgl. Beispiel 5c)

**moveS\_OE(OM, s, oe, attrValueSet)**

**Input:** OM: gültiges Organisationsmodell  
 s: zu verschiebende Stelle  
 oe: neuzugeordnete Organisationseinheit  
 attrValueSet: Attributmenge für resultierende Relation

**begin**

$r := (s, oe', gehört\_zu) \in OR$

**if**  $oe' \neq oe$  **then**

  deleteRelation(OM, r)

  createRelation(OM, s, oe, gehört\_zu, attrValueSet)

**endif**

**done**

Es müssen folgende Vorbedingungen erfüllt sein:

- $s = (\text{label}, \text{Stelle}), oe \in OE$
- $(s, oe, gehört\_zu) \notin OR$
- $attrValueSet = \{(attr_0, value_0), \dots, (attr_k, value_k)\} \quad k \in \mathbb{N}_0 \Rightarrow$   
 $\{attr_0, \dots, attr_k\} = f_{OTA}(gehört\_zu)$
- $\forall (attr_i, value_i) \in attrValueSet: value_i \in attr_i.Domain$

Die Nachbedingungen für diese Änderungstransaktion ergeben sich wiederum aus der Semantik der angewendeten Elementaroperationen.

- $OR^* = OR \cup \{(s, oe, gehört\_zu)\} \setminus \{(s, oe', gehört\_zu)\}$
- $OA^* = OA \cup \{(r, attr_0), \dots, (r, attr_k)\} \setminus \{(r', attr_0), \dots, (r', attr_l)\} \quad k, l \in \mathbb{N}_0$   
 mit  $r = (s, oe, gehört\_zu), r' = (s, oe', gehört\_zu)$
- $OAV^* = OAV \cup \{(r, attr_0, value_0), \dots, (r, attr_k, value_k)\} \setminus \{(r', attr_0, value_0), \dots, (r', attr_l, value_l)\} \quad k, l \in \mathbb{N}_0$   
 mit  $r = (s, oe, gehört\_zu), r' = (s, oe', gehört\_zu)$

## 4.6 Zusammenfassung

In diesem Abschnitt wurden die notwendigen Operationen, mit denen sich Änderungen des Organisationsmodells durchführen lassen, beschrieben.

Zunächst wurden Änderungsprimitive mit präzise formulierten Vor- und Nachbedingungen definiert, auf denen elementare Änderungsoperationen aufsetzen. Zur Einhaltung der in Abschnitt 2.2.1 geforderten Einschränkungen wurde ein Änderungstransaktionskonzept entwickelt. Darüber hinaus wurden ausgewählte, vordefinierte komplexe Änderungsoperationen am Beispiel demonstriert.

## 5 Pflege der Bearbeiterformeln (statischer Fall)

Nachdem bisher Änderungen des Organisationsmodells noch isoliert betrachtet worden sind, stehen in diesem Abschnitt die Auswirkungen dieser Änderungen auf Bearbeiterformeln, welche das Organisationsmodell referenzieren, im Vordergrund.

Bearbeiterformeln werden im allgemeinen nicht nur zur Zuweisung von Workflow-Aktivitäten an Bearbeiter verwendet, sondern kommen auch bei der Vergabe von Zugriffs-, Änderungs- und Ausführungsrechten in Informationssystemen zum Einsatz (Sparr, 2001).

Wenn nun das referenzierte Organisationsmodell verändert wird, kann es vorkommen, daß (Teil-)Ausdrücke der Bearbeiterformeln nicht mehr korrekt definiert sind und somit verwaiste Referenzen auftreten. Die daraus entstehenden Probleme und ihre Behandlung sollen im folgenden besprochen werden.

In diesem Abschnitt wird zunächst der statische Fall betrachtet, bei dem noch keine von den Bearbeiterformeln abgeleiteten Datenstrukturen (zum Beispiel Arbeitslisten) existieren, wie später zur Laufzeit (Abschnitt 6).

### 5.1 Grundprinzipien der Bearbeiterformeln

Mittels Bearbeiterformeln lassen sich Zugriffs- und Ausführungsrechte definieren. Eine solche Formel referenziert immer Entitäten eines konkreten Organisationsmodells. Sie beschreibt eine Menge von Personen, für die die jeweiligen Berechtigungen gelten sollen.

Normalerweise werden von Bearbeiterformeln relativ statische, organisatorische Entitäten wie Stellen, Organisationseinheiten oder Rollen referenziert. Nach ihrer Auswertung liefern die Bearbeiterformeln Mengen von Stellen zurück, aus denen indirekt die Mitarbeiter ermittelt werden, die diese Stellen innehaben. Natürlich können Mitarbeiter in Bearbeiterformeln auch direkt referenziert werden. Allerdings finden in Organisationen relativ oft Wechsel unter Mitarbeitern statt, so daß diese Formeln häufig angepaßt werden müßten.

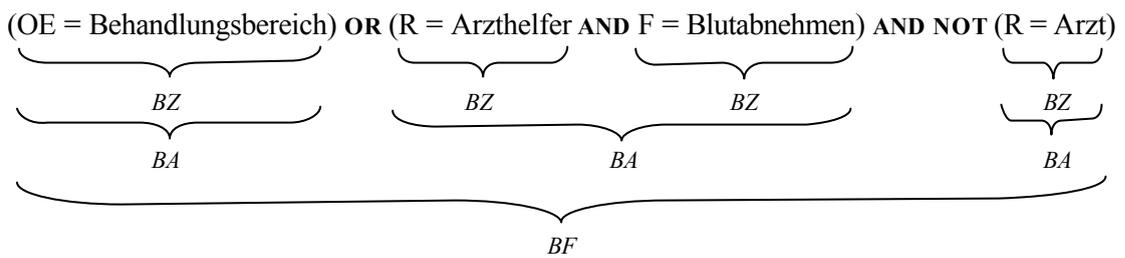
Bearbeiterformeln (BF) bestehen aus Bearbeiterausdrücken (BA), die mit den logischen Operatoren OR und AND NOT verknüpft werden. Die Bearbeiterausdrücke selbst enthalten Bearbeiterzuordnungen (BZ), die mit den logischen Operatoren AND verknüpft werden können.

Das geschieht also nach folgendem Schema:

Bearbeiterformel:	BA OR BA ...OR BA AND NOT BA ...AND NOT BA
Bearbeiterausdruck:	BZ AND BZ ...AND BZ

Die Bearbeiterzuordnung ist demnach die kleinste Einheit in der Konstruktion von Bearbeiterformeln. Beispiel 6 verdeutlicht den Aufbau von Bearbeiterformeln:

**Beispiel 6** (bezieht sich auf das Beispiel einer ärztlichen Praxis, Abschnitt 2.2.2)



Die Auswertung dieser Bearbeiterformel wird anschließend in Beispiel 7 demonstriert.

Tabelle 12 zeigt eine Aufstellung der wichtigsten Typen von Bearbeiterzuordnungen.

Bearbeiterzuordnung	Bedeutung
AG = X	Bearbeiter ist Mitglied der <i>Arbeitsgruppe X</i>
AGl = X	Bearbeiter ist <i>Leiter</i> der <i>Arbeitsgruppe X</i>
AGu = X	Bearbeiter ist Mitglied der <i>Arbeitsgruppe</i> , die von der Stelle X <i>geleitet</i> wird
F = X	Bearbeiter hat die <i>Fähigkeit X</i>
MA = X	Bearbeiter ist <i>Mitarbeiter X</i>
OE = X	Bearbeiter gehört der <i>Organisationseinheit X</i> an
OE+ = X	Bearbeiter gehört einer <i>Organisationseinheit</i> an, die der <i>Organisationseinheit X übergeordnet</i> ist
OE+n = X	Bearbeiter gehört einer <i>Organisationseinheit</i> an, die der <i>Organisationseinheit X n-fach höher übergeordnet</i> ist
OE- = X	Bearbeiter gehört einer <i>Organisationseinheit</i> an, die der <i>Organisationseinheit X beliebig untergeordnet</i> ist
OE-n = X	Bearbeiter gehört einer <i>Organisationseinheit</i> an, die der <i>Organisationseinheit X n-fach niedriger untergeordnet</i> ist
OEl = X	Bearbeiter ist <i>Leiter</i> der <i>Organisationseinheit X</i>
OEu = X	Bearbeiter ist Mitglied der <i>Organisationseinheit</i> , die von der Stelle X <i>geleitet</i> wird
OG = X	Bearbeiter ist Mitglied einer <i>Organisationseinheit</i> , die der <i>Organisationsgruppe X</i> angehört
R = X	Bearbeiter hat <i>Rolle X</i>
R+ = X	Bearbeiter hat angegebene oder <i>spezialisiertere Rolle</i>
S = X	Bearbeiter besetzt die <i>Stelle X</i>
S+ = X	Bearbeiter ist <i>Vorgesetzter</i> der <i>Stelle X</i> (beliebige Ebene)
S+n = X	Bearbeiter ist ( <i>n-facher</i> ) <i>Vorgesetzter</i> der <i>Stelle X</i>
S- = X	Bearbeiter ist <i>Untergebener</i> der <i>Stelle X</i> (beliebige Ebene)
S-n = X	Bearbeiter ist ( <i>n-facher</i> ) <i>Untergebener</i> der <i>Stelle X</i>

**Tabelle 12** Typen von Bearbeiterzuordnungen

Bearbeiterzuordnungen können *unabhängig* oder *abhängig* sein. Aus unabhängigen Bearbeiterzuordnungen lassen sich die Stellen potentieller Bearbeiter direkt aus einem einfachen Ausdruck ableiten. Abhängige Bearbeiterzuordnungen beziehen sich dagegen auf die Stelle des Bearbeiters einer referenzierten Aktivität, von der ausgehend potentielle Bearbeiter abgeleitet werden, zum Beispiel "*gleiche Stelle wie bei Aktivität Y*" oder "*Vorgesetzter der Stelle von Aktivität Z*". Die entsprechende

Stelle kann erst zur Laufzeit ermittelt werden, wenn feststeht, wer die referenzierte Aktivität ausgeführt hat.

Bei der Auswertung einer Bearbeiterformel werden zunächst die Bearbeiterausdrücke mit Hilfe des Organisationsmodells aufgelöst und eine Vereinigungsmenge über die ermittelten Bearbeiter gebildet, solange bis die erste AND NOT-Verknüpfung erscheint. Dann wird für alle bisher ermittelten Bearbeiter überprüft, ob sie das Ausschlußkriterium erfüllen. Falls das zutrifft, werden diese Bearbeiter aus der Menge wieder entfernt.

Ein Beispiel soll die Formulierung und Auswertung von Bearbeiterzuordnungen verdeutlichen.

**Beispiel 7:** (vgl. Bearbeiterformel aus Beispiel 6)

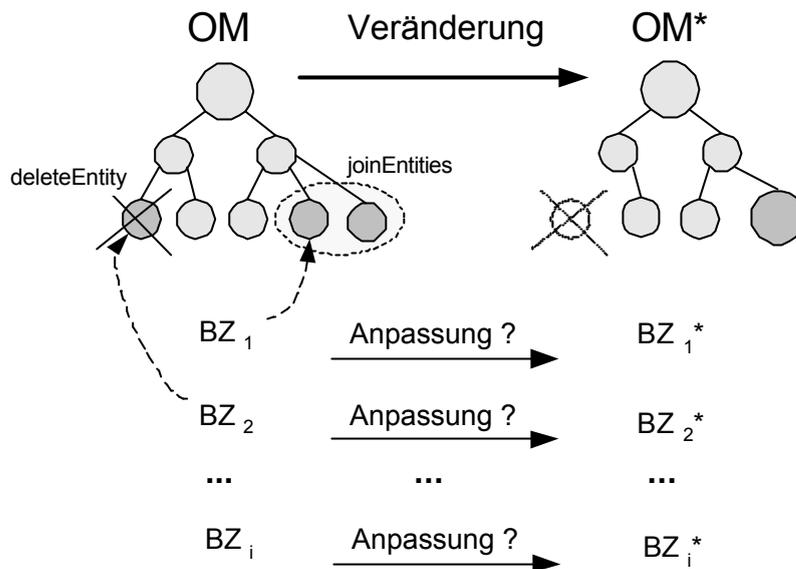
(OE = Behandlungsbereich) **OR** (R = Arzthelfer **AND** F = Blutabnehmen) **AND NOT** (R = Arzt)

Aus (OE = Behandlungsbereich) werden die Stellen *Praxis-Arzt* und *1. Arzthelfer* ermittelt. Durch **OR** (R = Arzthelfer **AND** F = Blutabnehmen) wird dieser Menge die Stelle *2. Arzthelfer* hinzugefügt. Bei der Auswertung von **AND NOT** (R = Arzt) wird die Stelle *Praxis-Arzt* wieder aus der Menge entfernt. Potentielle Bearbeiter dieser Aktivität sind also die Inhaber der Stellen *1. Arzthelfer* und *2. Arzthelfer*.

## 5.2 Problemstellung

### 5.2.1 Bearbeiterformeln

Wie in Abschnitt 5.1 verdeutlicht, referenzieren Bearbeiterformeln bzw. -zuordnungen immer ein konkretes Organisationsmodell. Wenn dieses Modell nun verändert wird, muß gewährleistet sein, daß die Bearbeiterformeln weiterhin konsistent und semantisch korrekt sind. Das heißt, die Veränderung des Organisationsmodells muß ggf. an die Bearbeiterzuordnungen propagiert werden, damit diese entsprechend angepaßt werden können (siehe Abbildung 9). Ansonsten kann es vorkommen, daß eine Bearbeiterformel nicht mehr auflösbar ist, weil in ihr eine organisatorische Entität referenziert wird, die zum Beispiel aus dem Organisationsmodell gelöscht wurde und damit nicht mehr existiert (verwaiste Referenz). Andererseits gibt es auch Modelländerungen, die keine Bearbeiterformeln betreffen. Wenn beispielsweise eine Stelle neu geschaffen wurde, kann sie bisher noch nicht durch eine existierende Bearbeiterzuordnung referenziert worden sein. Dem entsprechend ist auf der Ebene der Bearbeiterformeln keine Anpassung erforderlich. Aus der Semantik der angewandten Änderung können also ggf. Rückschlüsse auf erforderliche Anpassungen der Bearbeiterformeln getroffen werden.



**Abbildung 9** Problem der Anpassung von Bearbeiterzuordnungen (BZ) nach Änderungen des Organisationsmodells (OM)

Zunächst muß also geklärt werden, bei welchen Organisationsmodell-Änderungen es überhaupt nötig wird, die Bearbeiterzuordnungen anzupassen und wie dies geschehen kann. Darüber hinaus wird es kaum notwendig sein, *sämtliche* Bearbeiterzuordnungen zu verändern, da meist nur ein Teil von ihnen betroffen ist.

Bezogen auf konkrete Bearbeiterzuordnungen ( $BZ_i$ ) können sich aus Organisationsmodell-Änderungen zwei Fälle ergeben:

1. Fall: Die Bearbeiterzuordnung muß nicht angepaßt werden, da sie von der Organisationsmodell-Änderung nicht betroffen ist. Sie ist weiterhin definiert. Es gilt:  $BZ_i^* = BZ_i$
2. Fall: Die Bearbeiterzuordnung muß angepaßt werden, da sie nach der Organisationsmodell-Änderung nicht mehr definiert ist. Es gilt  $BZ_i^* \neq BZ_i$ .

Dabei sei  $BZ_i$  diejenige Bearbeiterzuordnung, die das ursprüngliche Organisationsmodell OM referenziert,  $BZ_i^*$  sei die Bearbeiterzuordnung, die veränderte Organisationsmodell OM\* referenziert.

Im ersten Fall kann die Bearbeiterzuordnung unverändert bestehen bleiben. Im zweiten Fall jedoch ist eine Anpassung der Bearbeiterzuordnung nötig. Hier muß weiter geklärt werden,

- bei *welchen* Organisationsmodell-Änderungen eine Anpassung notwendig ist (Abschnitt 5.3.1),
- *wie* die Anpassung der Bearbeiterformel konkret erfolgen kann (Abschnitt 5.3.2),

- *wer* die Anpassung durchführen darf (Abschnitt 5.3.3),
- und *wann* die Anpassung erfolgen kann (Abschnitt 5.3.4).

Diese Fragen, die bisher in der relevanten Literatur kaum aufgegriffen wurden, werden in den folgenden Abschnitten behandelt.

## 5.2.2 Bearbeitermengen

Ein weiterer Aspekt von Organisationsmodell-Änderungen betrifft die Bearbeitermengen, die aus den Bearbeiterformeln ermittelt werden.

Änderungen am Organisationsmodell können unterschiedliche Auswirkungen auf diese Mengen haben. So bleiben die Bearbeitermengen nach einer Attributänderung<sup>15</sup> im Organisationsmodell unverändert. Die entsprechende Bearbeitermenge kann sich auch verkleinern, wenn beispielsweise die Zuordnung einer Stelle zu einer referenzierten Organisationseinheit gelöscht wird. Umgekehrt kann sich die Menge auch vergrößern (vgl. Beispiel 8). Die resultierende Bearbeitermenge kann sich auch völlig von der ursprünglichen unterscheiden.

Abbildung 10 zeigt die möglichen Fälle von Auswirkungen der Organisationsmodell-Änderungen auf die Bearbeitermengen der angepaßten Bearbeiterzuordnungen. Dabei sei  $BM$  die Bearbeitermenge zur Bearbeiterzuordnung  $BZ$  auf dem Organisationsmodell  $OM$  und  $BM^*$  die resultierende Bearbeitermenge zur (ggf. angepaßten) Bearbeiterzuordnung  $BZ^*$  auf dem veränderten Organisationsmodell  $OM^*$ .

Organisationsmodell-Änderungen können trotz gleichbleibender Bearbeiterzuordnungen zu unterschiedlichen Bearbeitermengen führen.

Aus  $BZ_i^* = BZ_i$  folgt also keinesfalls  $BM_i^* = BM_i$  (Beispiel 8). Diese Beobachtung ist vor allem für die Betrachtungen über das Verhalten des Systems zur Laufzeit (vgl. Abschnitt 6) relevant.

**Beispiel 8:** Ausgangssituation sei die ärztliche Praxis aus Beispiel 3 (Abschnitt 2.2.2). Dem Behandlungsbereich wird eine neue Stelle *3.Arzhelfer* hinzugefügt. Eine Bearbeiterzuordnung laute  $BZ_1$ : (OE = Behandlungsbereich). Sie bleibt nach der Organisationsmodell-Änderung bestehen, es gilt:  $BZ_1^* = BZ_1$ . Die ursprüngliche zugehörige Bearbeitermenge ist  $BM_1 = \{Praxis-Arzt, 1.Arzhelfer\}$ . Nach der Modelländerung ist  $BM_1^* = \{Praxis-Arzt, 1.Arzhelfer, 3.Arzhelfer\}$ . Es gilt also:  $BM_1^* \supset BM_1$ .

<sup>15</sup> Attribute werden von Bearbeiterzuordnungen nicht referenziert. Wenn dem so wäre, müßten Attributänderungen entsprechend in die weiteren Betrachtungen einbezogen werden.

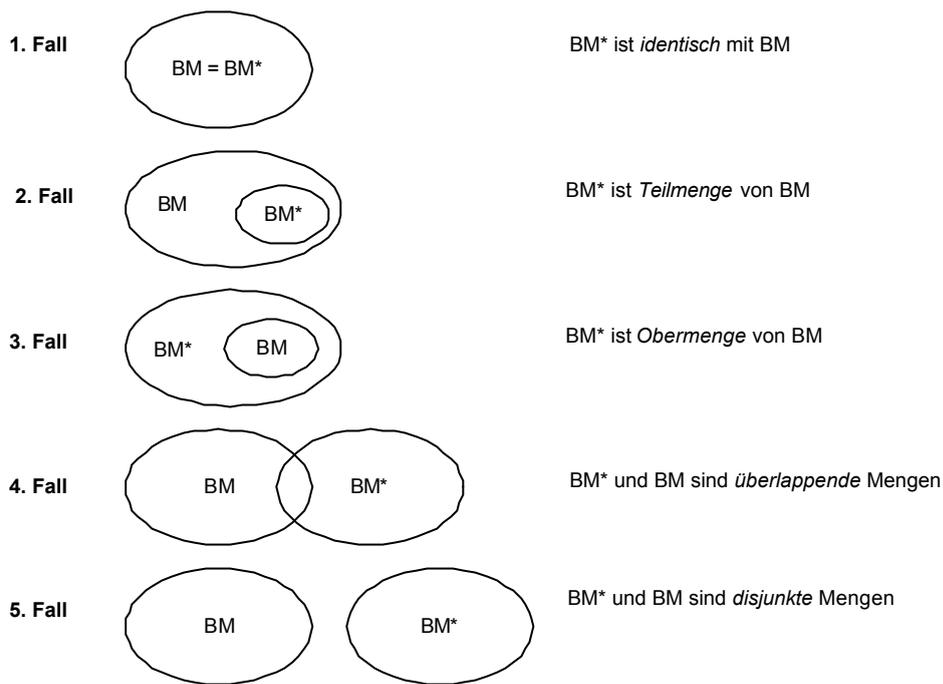


Abbildung 10 Verhältnis der Bearbeitermengen vor (BM) und nach (BM\*) der Organisationsmodell-Änderung

### 5.3 Lösungsansätze zur Behandlung von Referenzierungs-Problemen

In diesem Abschnitt werden Strategien und Lösungsansätze zur Anpassung von Bearbeiterformeln nach Organisationsmodell-Änderungen behandelt. Angestrebt wird eine weitgehende Entlastung des Modellierers<sup>16</sup> durch systemseitige Automatisierung. Allerdings ist die automatische Anpassung durch das System nicht immer möglich und auch nicht in jedem Fall erwünscht. Der Modellierer muß jederzeit die Möglichkeit haben, einzugreifen und Änderungen manuell vorzunehmen. Eine automatische Anpassung der Bearbeiterformeln kann also als Default-Strategie angegeben werden, muß aber immer auch deaktivierbar sein.

Abbildung 11 zeigt die verschiedenen Varianten der Anpassung von Bearbeiterformeln. Neben der *automatischen* Anpassung, die das System selbständig durchführen kann, und zwar ohne daß der Modellierer eingreifen muß, gibt es die *semi-automatische* Anpassung. In diesem Fall gibt der Modellierer dem System die Informationen, die es anschließend für eine automatische Anpassung der Bearbeiterformeln braucht.

Eine *manuelle* Anpassung führt der Modellierer bzw. der Prozeßverantwortliche selbst aus. Sie kann allerdings sehr aufwendig werden, da von einer Änderung des Organisationsmodells sehr viele Bearbeiterformeln betroffen sein können.

<sup>16</sup> Welche Person konkret die Änderungen vornehmen darf, wird später geklärt. Zunächst gehen wir von einem berechtigten Modellierer aus.

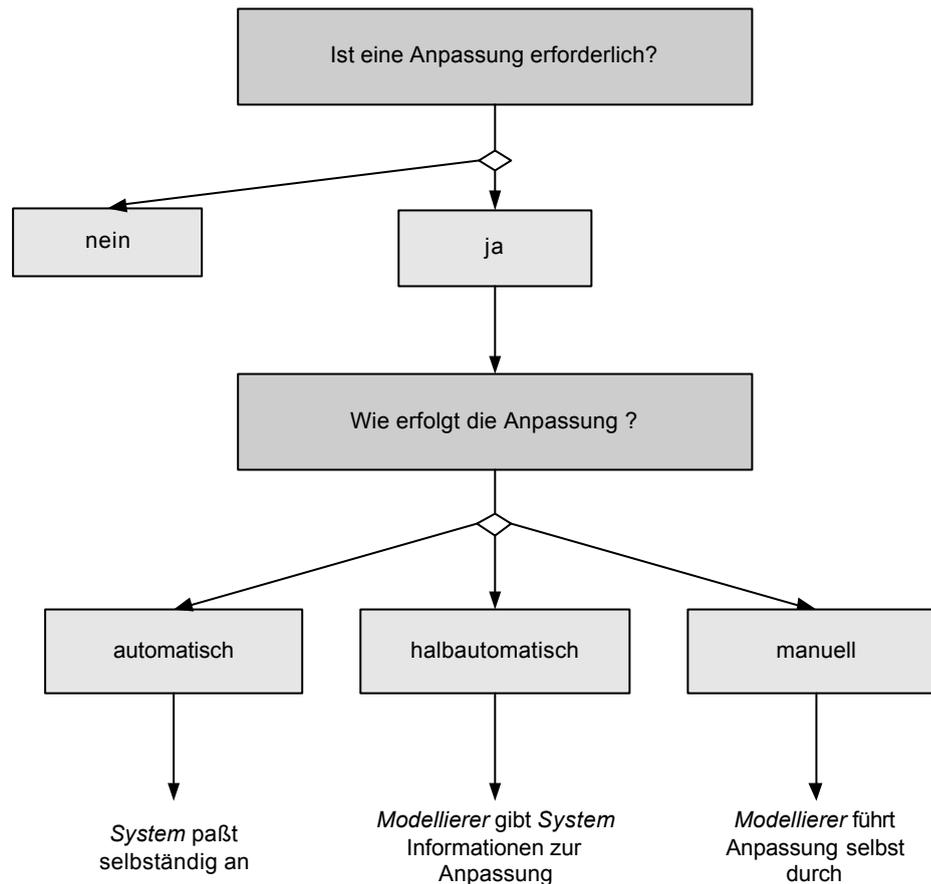


Abbildung 11 Varianten der Anpassung von Bearbeiterformeln

Im folgenden werden die in Abschnitt 5.2 entwickelten Fragestellungen weiterverfolgt.

Zunächst soll geklärt werden, bei welchen Änderungsoperationen des Organisationsmodells Bearbeiterformeln ggf. angepaßt werden müssen.

### 5.3.1 Welche Bearbeiterzuordnungen müssen bei welchen Änderungsoperationen angepaßt werden?

Ein naiver Ansatz wäre es, wenn man nach jeder Organisationsmodell-Änderung jede Bearbeiterzuordnung explizit daraufhin überprüfen würde, ob sie angepaßt werden muß oder nicht. Diese Vorgehensweise ist nicht nur mit einem enormen Aufwand verbunden, sondern ist - wie zuvor dargelegt - auch nicht notwendig. Deshalb wird eine Strategie benötigt, mit der über die Anpassung von Bearbeiterzuordnungen entschieden werden kann.

Ausgangspunkt der Betrachtungen sind zunächst die Bearbeiterzuordnungen als *kleinste Einheit* der Bearbeiterformeln (vgl. Abschnitt 5.1).

Tabelle 13 zeigt den Anpassungsbedarf von Bearbeiterzuordnungen in bezug auf die elementaren Änderungsoperationen des Organisationsmodells sowie die vordefinierten komplexen Änderungsoperationen zum Vereinigen, Teilen und Verschieben von organisatorischen Entitäten. Operationen zum Manipulieren von

Attributen sind an dieser Stelle nicht von Interesse, da Attribute in den Bearbeiterzuordnungen nicht referenziert werden.

Änderungsoperation	BZ anpassen	BZ <i>nicht</i> anpassen
createEntity	-	gilt immer
deleteEntity	falls von BZ referenziertes <i>X</i> gelöscht	sonst
createRelation	-	gilt immer
deleteRelation	-	gilt immer
join	falls von BZ referenziertes <i>X</i> beteiligt	sonst
split	falls von BZ referenziertes <i>X</i> geteilt	sonst
move	falls von BZ direkt referenzierter Mitarbeiter <i>X</i> verschoben	sonst

**Tabelle 13 Anpassungsbedarf von Bearbeiterzuordnungen (BZ) nach bestimmten Organisationsmodell-Änderungen**

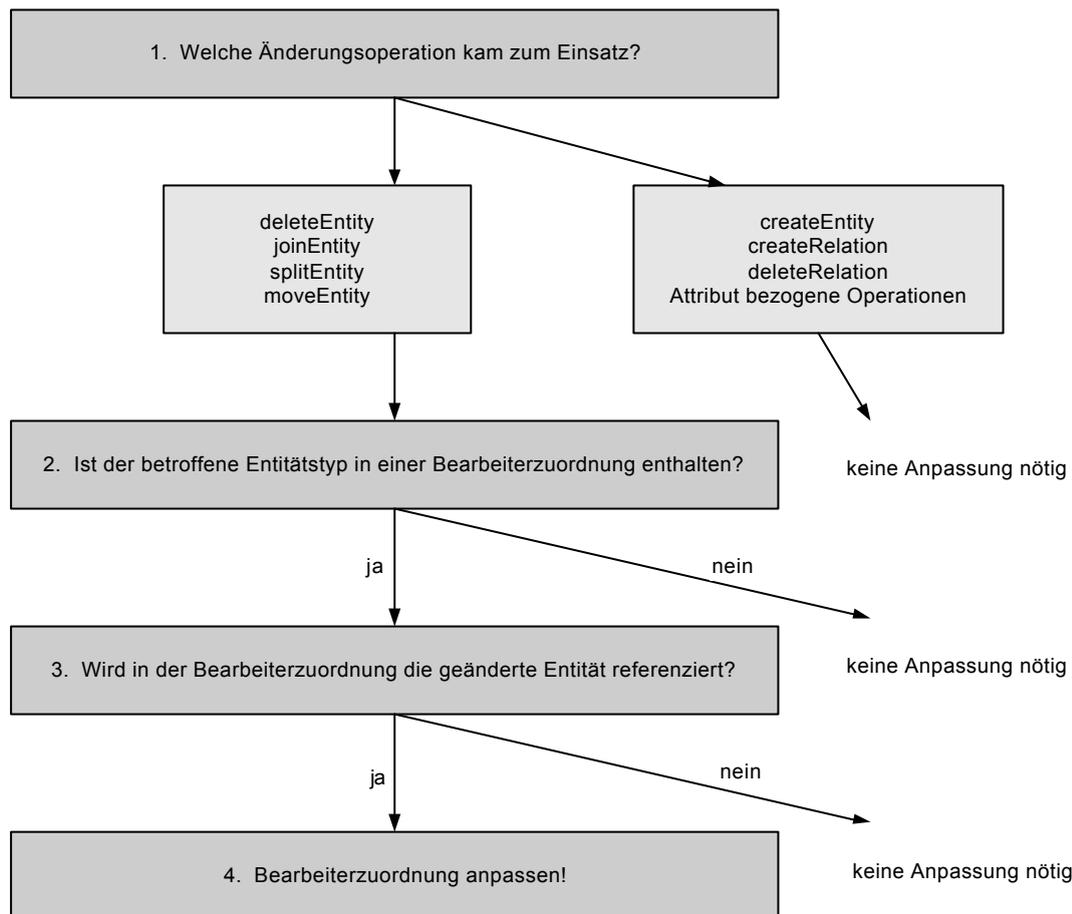
Wie leicht zu erkennen ist, sind additive Änderungsoperationen zum Einfügen von Entitäten oder Relationen für die Anpassung der Bearbeiterzuordnungen nicht kritisch, ebenso das Entfernen von Relationen aus dem Organisationsmodell. Neu eingefügte Entitäten können in bestehenden Bearbeiterzuordnungen naturgemäß noch nicht referenziert worden sein. Relationen werden in Bearbeiterzuordnungen prinzipiell nicht referenziert. Deshalb besteht in diesen Fällen kein Anpassungsbedarf für Bearbeiterzuordnungen.

Kritisch können dagegen Änderungsoperationen zum Entfernen, Vereinigen oder Teilen von Entitäten sein, wenn diese (veränderten) Entitäten in den Bearbeiterzuordnungen direkt referenziert werden. In diesen Fällen müssen entsprechende Anpassungen erfolgen.

Das Verschieben von organisatorischen Entitäten<sup>17</sup> ist an sich für Bearbeiterzuordnungen unkritisch. Wenn allerdings ein referenzierter Mitarbeiter einer anderen Stelle zugeordnet wird, sollte die Semantik der betroffenen Bearbeiterzuordnung durch den Modellierer überprüft werden. Damit wird sichergestellt, daß die ursprüngliche Absicht der direkten Mitarbeiterzuweisung erhalten bleibt.

Es läßt sich nun eine einfache Strategie zur Entscheidung über die Anpassung der Bearbeiterzuordnungen ableiten. Sie ist in Abbildung 12 dargestellt.

<sup>17</sup> Verschieben: Organisationseinheit zu anderer Organisationsgruppe, Stelle zu anderer Organisationseinheit, Stelle zu anderer Arbeitsgruppe, Mitarbeiter zu anderer Stelle zuordnen



**Abbildung 12** Entscheidungspfad zum Anpassungsbedarf von Bearbeiterzuordnungen nach Organisationsmodell-Änderungen

Nachdem geklärt ist, bei welchen Änderungsoperationen bestimmte Bearbeiterzuordnungen angepaßt werden müssen, stellt sich die Frage, wie diese Anpassungen vorgenommen werden können.

### 5.3.2 Wie können Bearbeiterzuordnungen angepaßt werden?

Zunächst sind zwei Fälle<sup>18</sup> von kritischen Organisationsmodell-Änderungen bezogen auf (elementare) Bearbeiterzuordnungen zu unterscheiden:

1. Änderungen, bei denen eine Entität tatsächlich aus der Organisation entfernt wurde und die deshalb in Zukunft nicht mehr referenziert werden kann,
2. Änderungen, nach denen die potentiellen Bearbeiter in der Organisation noch existieren, aber inzwischen anders referenziert werden müssen.

Im ersten Fall kann eine Bearbeiterzuordnung nach dem Löschen einer referenzierten Entität nicht mehr aufgelöst werden, weil ihr logischer Ausdruck nicht mehr definiert ist. Die Bearbeitermenge für diese Bearbeiterzuordnung bleibt leer. Eine automatische Anpassung ist auf dieser elementaren Ebene deshalb nicht möglich. Sinnvoll wäre eine Warnmeldung an den Modellierer mit der Aufforderung zur

<sup>18</sup> neben dem oben besprochenen, semantischen Sonderfall 'Mitarbeiter auf andere Stelle verschieben'

manuellen Anpassung der Bearbeiterformeln, zu der die betroffene Bearbeiterzuordnung gehört.

Auf der (komplexeren) Ebene der Bearbeiterformeln sind dagegen weitere Strategien möglich, auf die später in diesem Abschnitt eingegangen wird.

Im zweiten Fall existieren die potentiellen Bearbeiter noch, sind aber über die ursprüngliche Referenz nicht mehr zu erreichen. Dieser Fall liegt zum Beispiel bei der Anwendung der Änderungsoperationen *Teilen* und *Vereinigen* vor. Die Anpassung der Bearbeiterzuordnungen kann hier automatisch oder semiautomatisch erfolgen. Zwei Beispiele sollen das im folgenden demonstrieren.

### **Beispiel 9 Anpassung nach Split-Operationen**

Bestehende Bearbeiterzuordnung BZ:  $OG = X$

Änderung im Organisationsmodell:  $\text{split}(X) \rightarrow X1 + X2$

Die Bearbeiterzuordnung referenziert ursprünglich eine Organisationsgruppe  $X$ . Diese Organisationsgruppe  $X$  wird in  $X1$  und  $X2$  geteilt. Die Bearbeiterzuordnung kann anschließend nicht mehr aufgelöst werden.

**Automatische Anpassung:** Die Bearbeiterzuordnung ( $OG = X$ ) wird automatisch innerhalb der Bearbeiterformel durch diesen Ausdruck ersetzt:  $(OG = X1) \text{ OR } (OG = X2)$ .

Damit werden beide (Teil-)Organisationsgruppen referenziert.

**Semiautomatische Anpassung:** Um eine höhere Semantik zu erreichen, werden dem Benutzer die Ausdrücke  $(OG = X1)$ ,  $(OG = X2)$  und  $((OG = X1) \text{ OR } (OG = X2))$  als Alternativen zur Anpassung angeboten. Damit kann er entscheiden, welche Organisationsgruppe er referenzieren möchte.

### **Beispiel 10 Anpassung nach Join-Operationen**

Bestehende Bearbeiterzuordnungen  $BZ_1: OG = X$ ,  $BZ_2: OG = Y$

Änderung im Organisationsmodell:  $\text{join}(X, Y) \rightarrow Z$

Die Bearbeiterzuordnungen referenzieren ursprünglich die Organisationsgruppen  $X$  und  $Y$ . Diese Organisationsgruppen  $X$  und  $Y$  werden zu  $Z$  vereint. Die bestehenden Bearbeiterzuordnungen können auf Grundlage des neuen Organisationsmodells nicht mehr aufgelöst werden.

**Automatische Anpassung:** Die Bearbeiterzuordnungen ( $OG = X$ ) und ( $OG = Y$ ) werden automatisch durch den Ausdruck  $(OG = Z)$  ersetzt. Ausdrücke wie  $((OG = X) \text{ OR } (OG = Y))$  bzw.  $((OG = X) \text{ AND } (OG = Y))$  können weiter zu  $(OG = Z)$  vereinfacht, Ausdrücke wie  $((OG = X) \text{ AND NOT } (OG = Y))$  entfernt werden. Ein Eingreifen durch den Benutzer ist nicht unbedingt nötig.

Diese Anpassungsstrategien beziehen sich bisher auf Bearbeiterzuordnungen, die kleinsten Elemente der Bearbeiterformeln. Für die Anpassung der komplexeren Bearbeiterformeln, die sich aus AND-, OR- bzw. AND NOT-Verknüpfungen der Bearbeiterzuordnungen ergeben (vgl. Abschnitt 5.1), sind weitergehende Strategien möglich.

So kann nach dem Löschen einer Entität überprüft werden, ob die sie referenzierende Bearbeiterzuordnung in der Bearbeiterformel durch einen AND-, OR- oder AND NOT-Ausdruck verknüpft ist, um davon abhängig die Anpassungsstrategie zu entscheiden.

AND-Ausdrücke schränken die Bearbeitermenge ein. In diesem Fall sollte die Bearbeiterzuordnung zur Wahrung der Semantik explizit angepaßt werden.

**Beispiel 11 Anpassung von Bearbeiterformeln (BF) mit AND-Verknüpfungen**

BF: (OE = Behandlungsbereich) AND (Rolle = Arzt)

Änderung im Organisationsmodell: Rolle *Arzt* wird gelöscht

Würde hier (Rolle = Arzt) automatisch aus der Bearbeiterformel entfernt werden, bliebe: (OE = Behandlungsbereich). Damit würde ein erheblicher Teil der Semantik verloren gehen. Der Modellierer muß die Bearbeiterformel selbst anpassen.

OR-Ausdrücke erweitern die Bearbeitermenge. Hier wäre es möglich, die Bearbeiterzuordnung automatisch aus der Formel zu entfernen, sofern aus der Auflösung der übrigen Ausdrücke eine nicht leere Bearbeitermenge resultiert. Andernfalls sollte der Benutzer gewarnt werden.

**Beispiel 12 Anpassung von Bearbeiterformeln (BF) mit OR-Verknüpfungen**

BF: (OE = Behandlungsbereich) OR (Rolle = Arzt)

Änderung im Organisationsmodell: Rolle *Arzt* wird gelöscht

Wenn hier (Rolle = Arzt) automatisch aus der Bearbeiterformel entfernt wird, bleibt der Ausdruck (OE = Behandlungsbereich). Dieser reicht aus, um eine korrekte Bearbeitermenge zu berechnen.

AND NOT-Ausdrücke verkleinern die Bearbeitermenge. Hier wäre es ebenfalls möglich, die Bearbeiterzuordnung automatisch aus der Formel zu entfernen.

**Beispiel 13 Anpassung von Bearbeiterformeln (BF) mit AND NOT-Verknüpfungen**

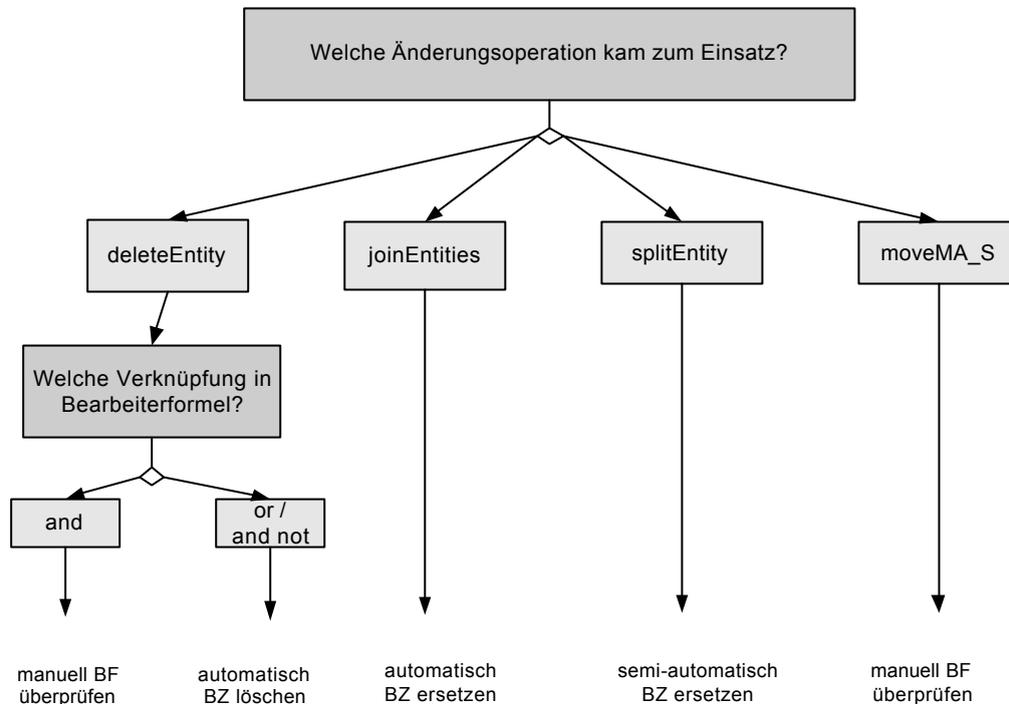
BF: (OE = Behandlungsbereich) AND NOT (MA = Brandt)

Änderung im Organisationsmodell: Mitarbeiter *Brandt* wird entfernt

Wenn (MA = Brandt) automatisch aus der Bearbeiterformel entfernt wird, bleibt der Ausdruck (OE = Behandlungsbereich). Dieser reicht wiederum aus, um eine Bearbeitermenge ohne Semantikverlust zu berechnen.

Die zuletzt genannten Anpassungsstrategien können dem Modellierer sehr viele manuelle Anpassungen der einzelnen Bearbeiterzuordnungen nach Löschoptionen im Organisationsmodell ersparen. Trotzdem muß der Modellierer noch die Möglichkeit haben, diese Anpassungen auch selbständig durchzuführen, zum Beispiel wenn er eine Entität durch eine andere ersetzen möchte. In diesem Fall wäre es nicht sinnvoll, beim Löschen der "alten" Entität alle entsprechenden Bearbeiterzuordnungen automatisch zu entfernen, um anschließend neue Bearbeiterausdrücke mit der "neuen" Entität zu schreiben.

Die vorangegangenen Beispiele haben gezeigt, daß die Anpassung der Bearbeiterformeln je nach Änderungsoperation automatisch, semi-automatisch oder manuell durch den Modellierer vorgenommen werden kann. Ein Entscheidungspfad für die jeweiligen Anpassungsstrategien ist in Abbildung 12 dargestellt.



**Abbildung 13** Strategien für die automatische, semi-automatische bzw. manuelle Anpassung der Bearbeiterformeln nach *kritischen* Organisationsmodell-Änderungen (BF Bearbeiterformel, BZ Bearbeiterzuordnung, moveMA\_S: Mitarbeiter auf andere Stelle verschieben)

Für komplexe, benutzerdefinierte Änderungsoperationen sind diese Strategien ebenfalls - auch kombiniert - anwendbar, da diese komplexen Operationen sich aus den elementaren bzw. vordefinierten Änderungsoperationen zusammensetzen.

Im folgenden soll nun geklärt werden, welche Personen überhaupt berechtigt sind, Änderungen am Organisationsmodell bzw. die Anpassung der Bearbeiterformeln vorzunehmen.

### 5.3.3 Wer führt die Anpassung durch?

Bisher haben wir von dem Modellierer als ändernde Person gesprochen. An dieser Stelle soll kurz geklärt werden, welche Personengruppen das Recht zur Änderung des Organisationsmodells bzw. zur Anpassung der Bearbeiterformeln erhalten sollten.

Da derartige Änderungen weitreichende Konsequenzen haben, zum Beispiel Zugriffs- und Ausführungsrechte für Aktivitäten ändern, sollten sie nur einem ausgewählten Personenkreis gestattet werden.

Sinnvollerweise sollte das Anlegen, Freigeben und Ändern von Organisationsmodellen nur von einem Systemadministrator bzw. einer Gruppe von

Systemadministratoren durchgeführt werden. Sie haben einerseits einen guten Überblick über die Strukturen der Organisation und sind andererseits nicht direkt in die späteren Geschäftsprozesse involviert (Sparr, 2001; S. 56 f.).

Die Definition und Pflege der Bearbeiterformeln finden im Kontext der Aktivitätenmodellierung statt. Nach Sparr (2001; S. 58) wäre es sinnvoll, dafür einen Prozeß- oder Aktivitätenmodellierer einzustellen, der einen guten Überblick sowohl über die Organisationsstrukturen als auch über die zu modellierenden Aktivitäten und Prozesse hat. Andernfalls können diese Aufgaben von dem Systemadministrator übernommen werden. Im Falle von Organisationsänderungen, die semi-automatische oder manuelle Anpassungen der Bearbeiterformeln erfordern, ist es jedoch sinnvoll, wenn diese Änderungen von derselben Person durchgeführt werden, die bereits den Kontext der Änderungen kennt.

Für laufende Prozeßinstanzen kann ein Prozeßverantwortlicher bestimmt werden. Dieser kann für einen kompletten Prozeß oder auch nur für einzelne Prozeßaktivitäten verantwortlich sein. Wenn im laufenden Betrieb Probleme auftauchen, zum Beispiel eine Bearbeiterformel nicht aufgelöst werden kann, kann er als Default-Bearbeiter eingesetzt werden. Er kann die anstehenden Aufgaben auch an andere Personen delegieren, sofern das Workflow-Management-System dafür einen entsprechenden Mechanismus vorsieht. Ist kein Prozeßverantwortlicher festgelegt, muß der Systemadministrator den laufenden Prozeß anpassen (Sparr, 2001; S. 63 ff.).

Zu Änderungen am Organisationsmodell oder an Bearbeiterformeln sind also nur wenige Personen berechtigt.

### **5.3.4 Wann wird die Anpassung durchgeführt?**

Für den Zeitpunkt der Anpassung von Bearbeiterformeln nach Änderungen des Organisationsmodells sind unterschiedliche Strategien denkbar. Die Anpassungen können prinzipiell sofort (*immediate*) oder aber zeitlich verzögert (*delayed/deferred*) nach der Organisationsmodell-Änderung vorgenommen werden.

#### **a) sofortige Anpassung**

Organisationsmodell-Änderungen finden normalerweise gekapselt in Änderungstransaktionen statt. Bis zum Abschluß der Änderungstransaktion durch Commit sind Zwischenzustände nach einzelnen Änderungsoperationen nach außen nicht sichtbar und ggf. auch nicht korrekt (vgl. Abschnitt 4.5). Die Anpassung von Bearbeiterformeln kann also frühestens im Rahmen des erfolgreichen Abschluß der Änderungstransaktion stattfinden.

Der wichtigste Vorteil einer sofortigen Anpassung von Bearbeiterformeln ist, daß die Bearbeiterformeln stets auf dem aktuellsten Stand sind. Außerdem finden semiautomatische oder manuelle Anpassungen im unmittelbaren Kontext der Modelländerung statt, so daß der Modellierer keine Schwierigkeiten hat, diese Änderungen (semantisch) einzuordnen.

Ein Nachteil dieser Strategie ist jedoch, daß sie sehr aufwendig ist. Nach jeder solchen Änderung muß das System für die Anpassungen der Bearbeiterformeln inklusive Korrektheitsprüfungen unterbrochen werden.

Wenn das Organisationsmodell weiter verändert wird, kann es außerdem vorkommen, daß soeben angepaßte Bearbeiterformeln schon wieder veraltet sind und angepaßt werden müssen. Durch solche *Lost Updates* wird wertvolle Rechenzeit vergeudet.

#### **b) verzögerte Anpassung**

Die Anpassung der Bearbeiterformeln kann auch zeitlich verzögert nach den Organisationsmodell-Änderungen stattfinden. Der genaue Zeitpunkt dafür kann entweder systemseitig festgelegt sein (zum Beispiel beim Schließen der Anwendung) oder explizit durch den Modellierer bestimmt werden.

Der Vorteil einer verzögerten Anpassung von Bearbeiterformeln liegt darin, daß Systemunterbrechungen seltener notwendig werden und damit längere Laufzeiten des Systems erreicht werden können. Außerdem werden die oben beschriebenen *Lost Updates* vermieden: Zwischenzustände des Organisationsmodells, die von späteren Änderungstransaktionen wieder aufgehoben werden, müssen nicht weiter propagiert werden.

Ein Nachteil dieser Strategie ist jedoch, daß bei jeder Auswertung einer Bearbeiterformel geprüft werden muß, ob die Bearbeiterformel noch gültig ist.

### **5.4 Zusammenfassung**

In diesem Abschnitt wurde diskutiert, welche Auswirkungen Organisationsmodell-Änderungen auf Crossreferenzen im *statischen* Fall haben. Im Mittelpunkt standen dabei Bearbeiterformeln, die ein konkretes Organisationsmodell referenzieren. Nach einer Änderung des Organisationsmodells können diese Bearbeiterformeln ggf. ungültig sein.

Es wurden Lösungsansätze geboten, die je nach Semantik der Änderung eine automatische, semiautomatische oder manuelle Anpassung der betroffenen Bearbeiterformeln erlauben. Außerdem wurde überlegt, welche Bearbeiterformeln überhaupt angepaßt werden müssen, wann und durch wen das geschehen kann.

## 6 Pflege der Arbeitslisten (dynamischer Fall)

Nachdem bisher in dieser Arbeit Änderungen des Organisationsmodells ausschließlich in bezug auf den statischen Fall betrachtet wurden, stehen nun die Auswirkungen der Modelländerungen auf *laufende* Instanzen eines Workflow im Mittelpunkt.

Welche Probleme können zur Laufzeit auftreten? Änderungen am Organisationsmodell können nun nicht mehr nur statische Bearbeiterzuordnungen der Aktivitäten von Prozeßmodellen (im Kontext von Workflows), sondern auch die Prozeßinstanzen betreffen. Da Arbeitslisten in der Regel bereits bei der Aktivierung des jeweiligen Prozeßschrittes einer Instanz berechnet werden, können nach Organisationsmodell-Änderungen veraltete oder ungültige Arbeitslisteneinträge auftreten.

Trotzdem müssen Änderungen am Organisationsmodell auch zur Laufzeit möglich sein, da das Fortschreiten eines Workflows aus wirtschaftlichen Gründen normalerweise nicht unterbrochen werden darf. Das gleiche Problem ergibt sich auch bei Änderungen der Bearbeiterformeln zur Laufzeit.

### 6.1 Grundlagen

Zunächst werden in diesem Abschnitt die grundlegende Prinzipien von Arbeitslisten und ihrer Verwaltung auf dem Server und den Klienten des Workflow-Management-Systems erläutert.

#### 6.1.1 Arbeitslisten

Für jeden angestoßenen Prozeß eines bestimmten Typs wird eine Workflow-Instanz von diesem Typ abgeleitet. Wird bei Ausführung dieser Instanz ein Prozeßschritt aktiviert, das heißt die zugehörige Aktivität als ausführbar markiert, so erhalten alle Prozeßbeteiligten, die berechtigt sind, diese Aktivität auszuführen, einen Eintrag in ihre aktuelle Arbeitsliste.

Zu jeder (eindeutigen) Stelle eines Organisationsmodells gehört eine solche Arbeitsliste. Nur diejenige Person, die diese Stelle innehat, hat lesenden Zugriff auf ihre Liste. In dieser Arbeitsliste sind alle anstehenden oder schon in Ausführung befindlichen Aktivitäten dieser Person aufgeführt. Falls die Person abwesend ist, können Aufträge mit Hilfe von Vertreterregelungen an andere Personen weitergereicht werden. Normalerweise erhält außerdem der Prozeßverantwortliche (zum Beispiel der Vorgesetzte) Zugriff auf fremde Arbeitslisten.

Zur Berechnung der Arbeitslisten werden zur Laufzeit die jeweiligen Bearbeiterformeln der aktivierten Prozeßschritte des Workflows aufgelöst. Die resultierenden Bearbeitermengen enthalten alle Adressaten, welche die entsprechenden Aktivitäten (*Workitems*) in ihre Arbeitslisten eingetragen bekommen.

Wenn ein Prozeßbeteiligter eine solche Aktivität aus seiner Arbeitsliste zur Ausführung auswählt, so wird der Eintrag bei allen anderen berechtigten Akteuren aus der Liste entfernt.

### 6.1.2 Serverseitige Verwaltung der Arbeitslisten

Die Verwaltung der Aufgaben (*Workitems*), die bestimmten Bearbeitern zugeordnet sind, erfolgt durch das Workflow-Management-System.

Abbildung 14 zeigt die internen Datenstrukturen, die im Workflow-Management-System ADEPT auf Serverseite für die Verwaltung von Arbeitslisten verwendet werden.

Dort werden sowohl alle am System angemeldeten Benutzer (Stellen) als auch alle vom Server kontrollierten Aktivitäteninstanzen, die sich im Zustand ACTIVATED, SELECTED oder RUNNING befinden, in mehrfach verketteten Listen verwaltet.

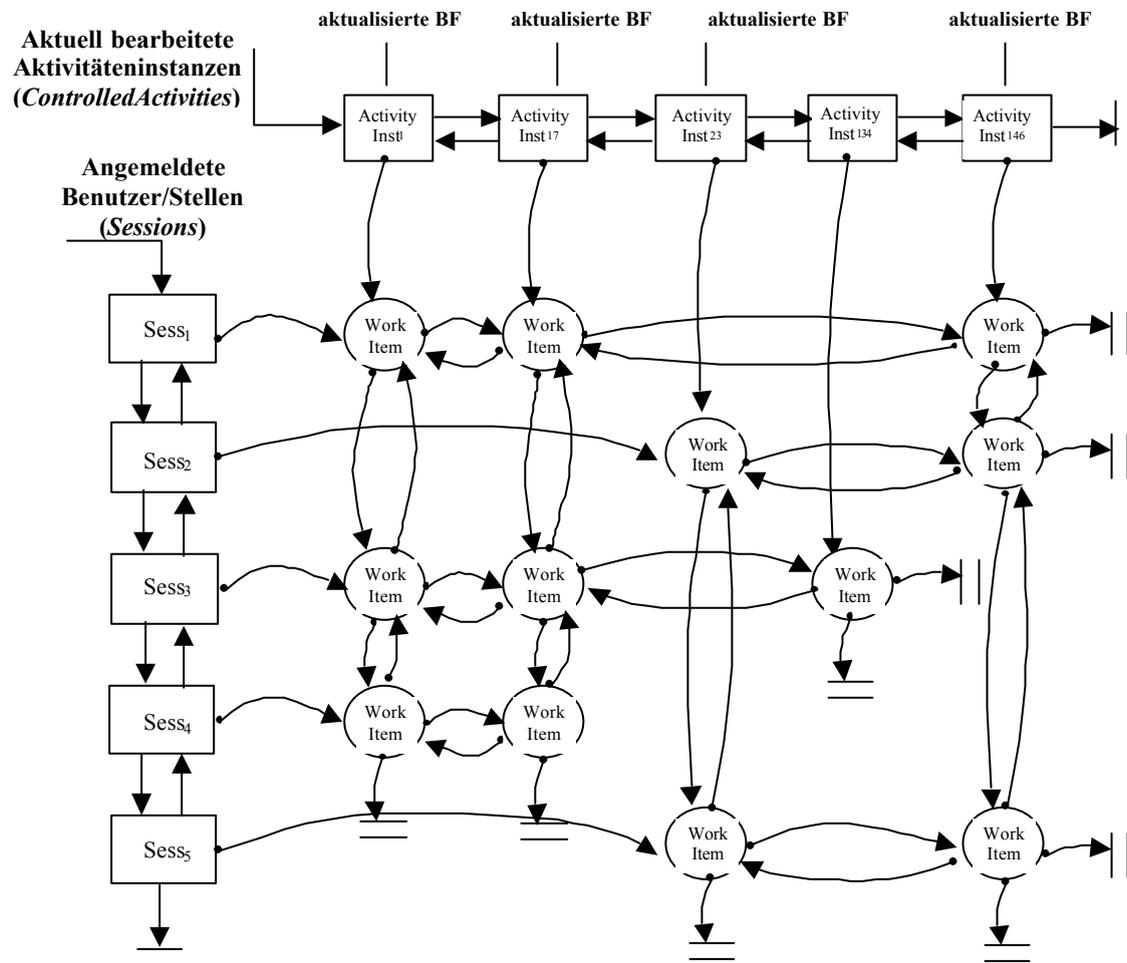


Abbildung 14 Datenstruktur zur Verwaltung von Benutzerarbeitslisten bei ADEPT (Reichert, 2002)

Über diese Listen kann ermittelt werden, welche Workitems einem angemeldeten Benutzer (mit einer bestimmten Stelle) zugeordnet wurden, und welchen angemeldeten Benutzern das Workitem einer bestimmten Aktivität zugeordnet wurde.

Für jede laufende Aktivität wird zusätzlich ihre *aktualisierte Bearbeiterformel* gespeichert (Abbildung 14). Bei ausgewählten oder laufenden Aktivitäten steht hier der konkrete Bearbeiter (bzw. seine Stelle). Außerdem werden *abhängige* Bearbeiterzuordnungen aufgelöst und in der Arbeitslistenverwaltung registriert. Beispielsweise wird der Ausdruck "*gleiche Organisationseinheit wie Bearbeiter der Aktivität 17*" bei Beendigung der *Aktivität 17* durch die *konkrete Organisationseinheit* ersetzt.

Die Bearbeiterformeln können auch modifiziert werden, um die zugehörigen Workitems an andere Bearbeiter zu delegieren. Dazu wird die entsprechende aktualisierte Bearbeiterformel in der Arbeitslistenverwaltung (Abbildung 14) überschrieben. In diesem Fall handelt es sich um eine *lokale* Änderung einer Aktivitäteninstanz und nicht um eine globale Schemaänderung, die alle Aktivitäten des Typs betreffen würde<sup>19</sup>.

Wenn sich nun ein neuer Benutzer im System anmeldet, wird anhand der aktualisierten Bearbeiterformeln berechnet, welche Workitems ihm zugeordnet werden.

Wird eine Aktivität aktiviert, so wird sie in die Aktivitätenliste der Arbeitslistenverwaltung aufgenommen. Die Bearbeiterformel wird ggf. aktualisiert und aufgelöst. Die jeweilige Workitem-Liste der angemeldeten Benutzer wird um das hinzugekommene Workitem ergänzt, sofern der Benutzer zu der berechneten Bearbeitermenge gehört.

Wenn ein Bearbeiter ein Workitem auswählt, geht die entsprechende Aktivität in den Zustand SELECTED (bzw. später beim Starten der Aktivität in den Zustand RUNNING) über. Falls das Workitem auch anderen Bearbeitern zugewiesen war, wird es nun aus deren Workitem-Listen entfernt. Ist eine Aktivität abgeschlossen (COMPLETED), werden ihre Einträge aus der Arbeitslistenverwaltung entfernt.

Auf diese Weise werden hier stets die aktuellen Zuordnungen von Workitems an die (potentiellen) Bearbeiter gespeichert. Darauf basierend werden die klientenseitigen Arbeitslisten erstellt.

---

<sup>19</sup> Dafür würde man die Bearbeiterformel der entsprechenden Aktivitätsdefinition im Prozeßmodell ändern.

### 6.1.3 Klientenseitige Verwaltung der Arbeitslisten

Wenn ein Benutzer sich beim Workflow-Management-System anmeldet, erfolgt in der Arbeitslistenverwaltung des Servers ein Session-Eintrag. Die aktuelle Workitem-Liste wird berechnet und an den Klienten übermittelt.

Falls sich nun die zugrundeliegende Workitemzuordnung beim Server (Abbildung 14) ändert, weil zum Beispiel eine Bearbeiterformel modifiziert wurde oder ein Workitem von einem anderen Bearbeiter selektiert wurde, sind die bereits berechneten Arbeitslisten bei den Klienten nicht mehr aktuell.

Um diese klientenseitigen Arbeitslisten zu aktualisieren, gibt es folgende Varianten:

- *Poll* - der Klient holt sich die Informationen zur Aktualisierung seiner Arbeitslisten aktiv beim Server.
- *Push* - die Aktualisierung wird durch den Server, der die Arbeitslisten verwaltet, gesteuert.

Bei der *Poll*-Methode wird die Arbeitsliste auf expliziten Wunsch des Klienten aktualisiert. Der Klient hat die volle Kontrolle über seine Arbeitsliste. Sie ändert sich nur, wenn er das möchte.

Der Informationsaustausch muß nicht nach jeder einzelnen serverseitigen Änderung stattfinden, sondern kann kompakt zum Beispiel in regelmäßigen Zeitabständen erfolgen. Dadurch wird der Rechen- und Kommunikationsaufwand verringert. Allerdings können Anfragen des Klienten auch dann erfolgen, wenn die Arbeitsliste sich seit der letzten Übertragung gar nicht geändert hat (Bauer & Dadam, 1998). Da der Klient keinerlei Informationen darüber hat, ob seine Arbeitsliste noch aktuell ist, kann sie unvollständig sein oder inzwischen ungültige Einträge enthalten. Deshalb ist es sinnvoll, die Arbeitsliste zumindest beim Zugriff auf Workitems zu validieren.

Die *Push*-Methode hat den Vorteil, daß der Klient sich nicht selbst um die Aktualisierung seiner Arbeitslisten kümmern muß. Er bekommt stets die neuesten Daten geliefert. Allerdings ist diese Methode mit einem hohen Rechen- und Kommunikationsaufwand verbunden. Außerdem hat der Klient keine Kontrolle über seine Arbeitsliste (Leymann & Roller, 2000). Sie kann sich häufig ändern, was als sehr störend empfunden werden kann. Um derartige Nachteile abzumildern, ist es sinnvoll, die Aktualisierung der Arbeitsliste eine gewisse Zeit zu verzögern und die Änderungen im Block zu übertragen (Bauer & Dadam, 1998).

## 6.2 Problemstellung

Wenn sich nun zur Laufzeit das Organisationmodell ändert, das von Bearbeiterformeln referenziert wird, dann sind zuvor berechnete Arbeitslisten unter Umständen nicht mehr aktuell. So können Arbeitslisten existieren, deren zugehörige Stelle nach der Organisationsmodell-Änderung nicht mehr vorhanden ist. Damit kommen die Aktivitäteninstanzen, die nur in solchen Arbeitslisten eingetragen sind, eventuell nicht mehr zur Ausführung. Ein weiteres Problem ist, daß aktivierte

Aktivitäten nach einer Vergrößerung der Bearbeitermenge nicht *allen* zur Ausführung berechtigten Personen angeboten werden. Dies scheint weniger kritisch zu sein als fehlende Bearbeiter, andererseits kann es dadurch zu unnötigen Verzögerungen im Arbeitsablauf kommen, weil die mögliche Entlastung für die betroffenen Mitarbeiter entfällt.

Besonders kritisch sind veraltete Arbeitslisten jedoch dort, wo falsche Zugriffsberechtigungen entstehen. Ein besonders schwerwiegendes Fehlverhalten eines existierenden Workflow-Management-Systems beschreibt Martschat (2001; S. 84):

**Beispiel 14: Fehlverhalten nach Entfernen des Bearbeiters zur Laufzeit**

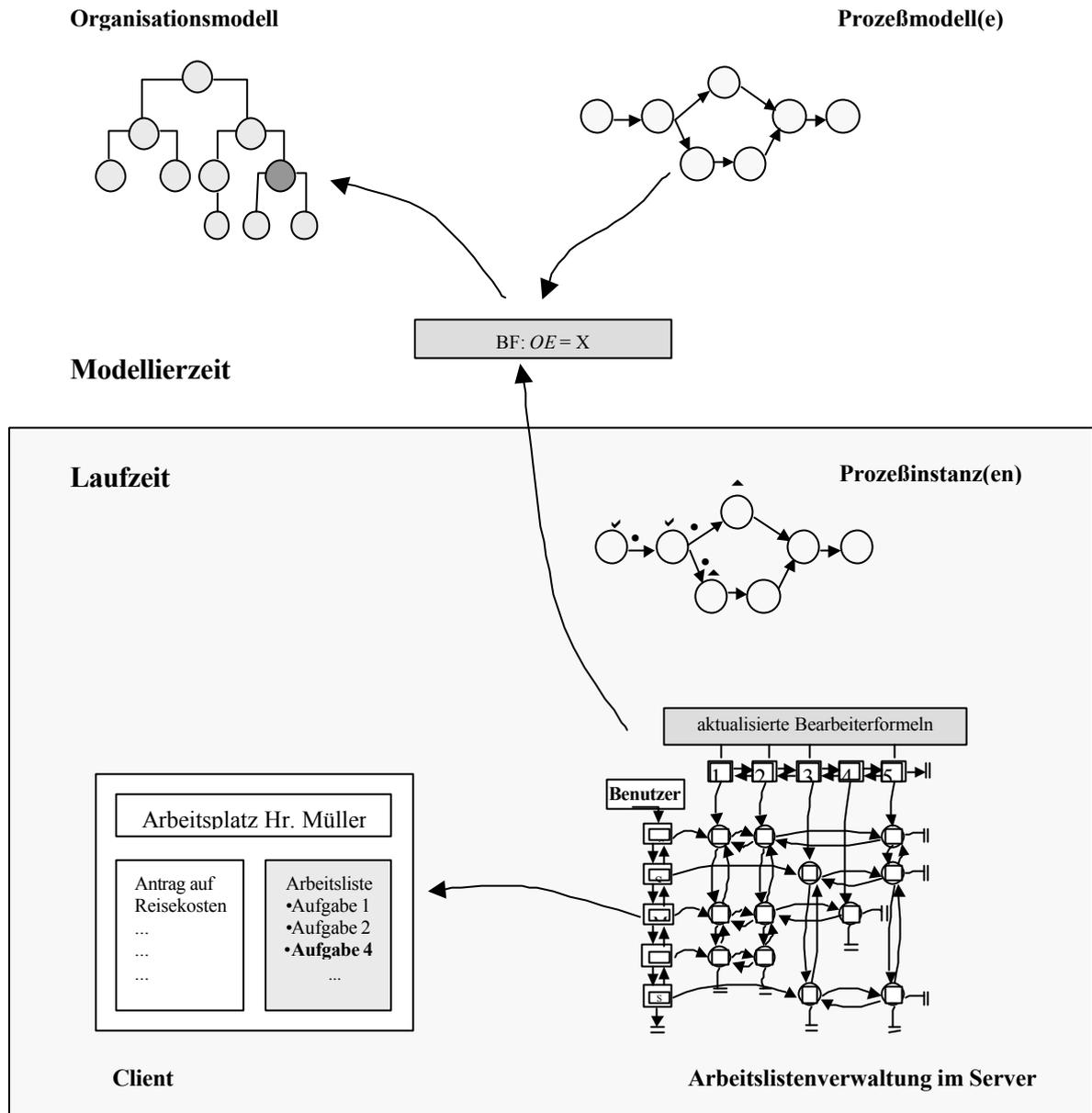
Einer Aktivität wird eine einzelne Person als Bearbeiter zugeordnet. Wenn diese Person zur Laufzeit aus dem Organisationsmodell entfernt wird, so ist die Aktivität ohne Bearbeiterzuordnung. Bei der Aktivierung dieser Aktivität wies MQSerries das entsprechende Workitem "jeder in der Run-Time-Komponente bekannten Person (vom Azubi bis zum Manager)" zu. Das widerspricht natürlich grundlegenden Sicherheitserwartungen an ein solches System.

Es wird also in vielen Fällen nötig sein, Arbeitslisten nach Organisationsmodell-Änderungen (bzw. auch nach Änderung der Bearbeiterformeln) neu zu berechnen. Ein großes Problem besteht dabei in der Pflege der Cross-Referenzen zwischen den verschiedenen Komponenten des Workflow-Managementsystems (Abbildung 15).

Nach einer kritischen Organisationsmodell-Änderung müssen demzufolge nicht nur die Bearbeiterformeln der modellierten Aktivitäten der Prozeßmodelle, sondern auch die jeweiligen, aktualisierten Bearbeiterformeln in der serverseitigen Arbeitslistenverwaltung sowie die betroffenen Arbeitslisteneinträge auf der Klientenseite angepaßt werden.

Da diese Anpassungen einen hohen Rechen- und Kommunikationsaufwand bedeuten, sollte genau geprüft werden, in welchen Fällen sie tatsächlich erfolgen müssen, und auch zu welchem Zeitpunkt. Es ist im allgemeinen nicht notwendig, jede Änderung sofort auf die Aktivitäteninstanzen zu propagieren. Welche Möglichkeiten es hierfür prinzipiell gibt, wird im nächsten Abschnitt diskutiert.

Wie durch das Beispiel 14 deutlich wurde, ist es auch von den Sicherheitsanforderungen der Anwendung abhängig, wie aktuell Arbeitslisten sein müssen. Einerseits muß sichergestellt werden, daß keine unberechtigten Zugriffe auf Aktivitäten und die damit verbundenen Daten stattfinden können, andererseits müssen die Aktivitäten aber überhaupt zur Ausführung kommen können. Deshalb ist es zwar erstrebenswert, daß alle Arbeitslisten stets auf dem aktuellen Stand sind. Dies erfordert jedoch einen erheblichen Aufwand, wie zum Beispiel notwendige Lesesperren der Datenstrukturen bei der Restrukturierung der Arbeitslistenverwaltung beim Server.



**Abbildung 15** Cross-Referenzen zwischen Organisationsmodell, Prozeßmodell sowie Prozeßinstanzen, serverseitigen Arbeitslistenverwaltung und klientenseitigen Arbeitslisteneinträgen zur Laufzeit

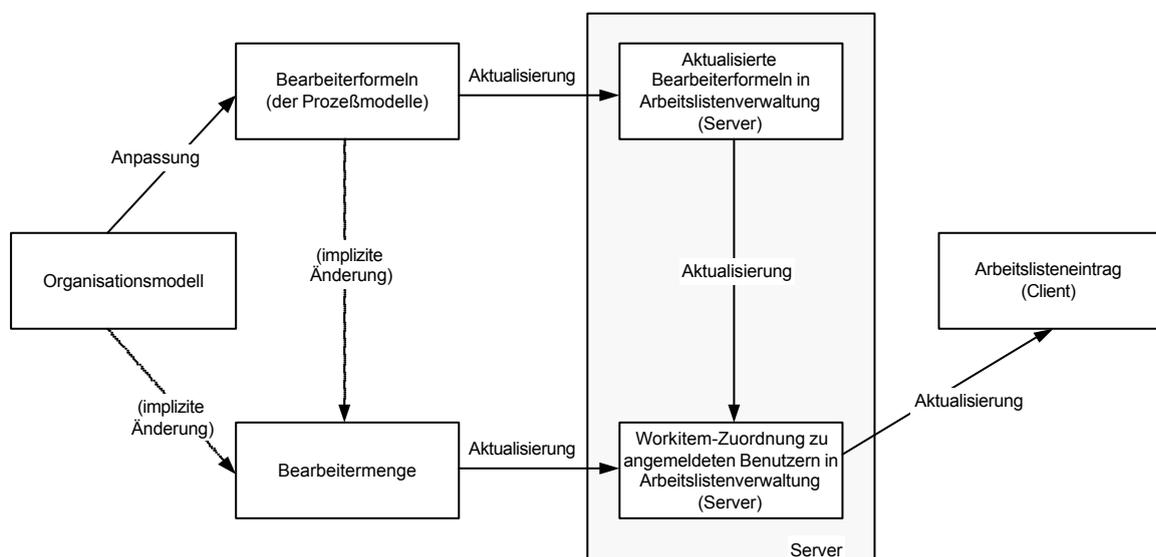
Da es nicht sinnvoll ist, nach jeder Änderung des Organisationsmodells sämtliche Arbeitslisten neu zu berechnen, müssen Strategien entwickelt werden, welche die Aktualisierung der Arbeitslisten effizient und korrekt gestalten.

Bei der Entscheidung für die eine oder andere Strategie sollte berücksichtigt werden, daß Organisationsmodell-Änderungen (und auch Änderungen der Bearbeiterformeln) im laufenden Betrieb normalerweise eher selten nötig sind. Änderungen in bezug auf Mitarbeiter werden dabei häufiger sein, als Änderungen von Stellen, Organisationseinheiten oder Rollen. Finden jedoch solche Änderungen statt, sind diese wahrscheinlich relativ komplex, so daß ggf. viele Anpassungen vorgenommen werden müssen.

Abbildung 15 zeigt, daß es verschiedene Komponenten in Workflow-Management-Systemen gibt, die nach einer Organisationsmodell-Änderung ggf. angepaßt werden müssen. Diese Komponenten können wiederum auch nachfolgende Aktualisierungen erfordern, wenn sie selbst entsprechend verändert wurden.

Die einzelnen Stufen dieser Abhängigkeiten sind folgende:

1. Das *Organisationsmodell* wurde geändert: Die *Bearbeiterformeln* müssen ggf. angepaßt werden (statischer Fall, vgl. Abschnitt 5).
2. Das *Organisationsmodell* wurde geändert: Die *Menge potentieller Mitarbeiter* hat sich dadurch eventuell geändert (vgl. Abschnitt 5).
3. Die Definition der *Bearbeiterformeln* wurde geändert: Die *Bearbeiterformeln* in der *Arbeitslistenverwaltung* müssen aktualisiert werden.
4. Die *Bearbeiterformeln* in der *Arbeitslistenverwaltung* wurden aktualisiert: Die berechneten *Bearbeitermengen* (Workitem-Zuordnungen) in der *Arbeitslistenverwaltung* müssen ebenfalls aktualisiert werden.
5. Die *Menge potentieller Mitarbeiter* hat sich verändert: Die berechnete *Bearbeitermenge* (Workitem-Zuordnung) in der *Arbeitslistenverwaltung* muß aktualisiert werden.
6. Die *Bearbeitermenge* (Workitem-Zuordnung) in der *Arbeitslistenverwaltung* des Servers hat sich geändert: *Arbeitslisteneinträge* der betroffenen Klienten müssen aktualisiert werden.



**Abbildung 16** Abfolge der Veränderungen vom Organisationsmodell, über Prozeßmodell, Arbeitslisten-Server bis zur Arbeitsliste des Klienten

Eine Änderung *einer* solchen Komponente, zum Beispiel des Organisationsmodells, kann *mehrstufige* Anpassungen erfordern. Abbildung 16 veranschaulicht den Zusammenhang dieser organisatorischen Veränderungen und ihrer nachfolgenden Anpassungen.

Änderungen des Organisationsmodells können demnach - wie in Abschnitt 5 dargelegt - sowohl Anpassungen der Bearbeiterformeln erfordern, als auch die potentielle Bearbeitermenge von Bearbeiterzuordnungen verändern.

Werden Bearbeiterformeln verändert, sei es als Folge von Organisationsmodell-Änderungen oder aus anderen Gründen (zum Beispiel Schemaänderung des Prozeßmodells), so müssen die entsprechenden in der Arbeitslistenverwaltung erfaßten Bearbeiterformeln aktualisiert und deren Workitem-Zuordnungen neu berechnet werden.

Die Workitem-Zuordnung muß auch dann neu berechnet werden, wenn die (potentielle) Bearbeitermenge sich verändert hat<sup>20</sup>. Das kann nach Organisationsmodell- oder Bearbeiterformel-Änderungen geschehen. Für Instanzen von Aktivitäten können die Bearbeiterformeln auch direkt in der Arbeitslistenverwaltung des Servers geändert werden, zum Beispiel um ein konkretes Workitem zu delegieren.

Aktualisierungsbedarf der Arbeitslisten der Klienten besteht, nachdem die Workitem-Zuordnungen in der Arbeitslistenverwaltung sich verändert haben (wie auch beim normalen Betrieb des Systems).

Im nächsten Abschnitt wird diskutiert, welche Möglichkeiten es prinzipiell gibt, die Arbeitslistenverwaltung zu aktualisieren.

### **6.3 Aktualisierung der Arbeitslistenverwaltung**

Im Mittelpunkt dieses Abschnitts steht die Aktualisierung der serverseitigen Arbeitslistenverwaltung. Die klientenseitige Verwaltung der Arbeitslisteneinträge wurde in Abschnitt 6.1.3 besprochen. Die dort erwähnten Methoden Push bzw. Poll werden zur Aktualisierung der Arbeitslisteneinträge beim (normalen) Fortschreiten des Workflows wie auch nach (außerordentlichen) organisatorischen Änderungen eingesetzt. Deshalb werden sie im folgenden nicht weiter betrachtet.

#### **6.3.1 Aktualisierungsvarianten**

Auf der Serverseite sind nach kritischen Änderungen des Organisationsmodells oder der Bearbeiterformeln die in der Arbeitslistenverwaltung registrierten Bearbeiterformeln und Workitem-Zuordnungen zu aktualisieren. Welche organisatorischen Änderungen konkret den Aktualisierungsbedarf der Arbeitslistenverwaltung nach sich ziehen, wird in Abschnitt 6.3.4 diskutiert.

Der Server<sup>21</sup> verwaltet alle aktivierten, ausgewählten oder laufenden Aktivitäten sowie sämtliche angemeldeten Benutzer mit ihren Workitem-Listen. Eine Aktualisierung der Arbeitslistenverwaltung bedeutet daher einen sehr hohen Rechenaufwand verbunden mit einer umfangreichen Restrukturierung der Daten.

---

<sup>20</sup> sowie beim An- und Abmelden von Benutzern im normalen Betrieb

<sup>21</sup> ggf. auch mehrere Server

Ferner muß sie in einem *kritischen Abschnitt* erfolgen, bei dem gleichzeitig kein weiterer Zugriff auf die entsprechenden Datenstrukturen des Servers gestattet ist. Deshalb sollten diese aufwendigen Aktualisierungen, die den normalen Betrieb unterbrechen, möglichst selten durchgeführt werden. Außerdem sollten sie weitgehend automatisch durch das System erfolgen, da eine manuelle Anpassung durch den Modellierer oder Prozeßverantwortlichen aufgrund der enormen Datenmenge in der Arbeitslistenverwaltung nicht praktikabel wäre.

Zunächst sind nach kritischen Änderungen des Organisationsmodells oder der Bearbeiterformeln der Prozeßmodelle zwei Varianten der Anpassung der Arbeitslistenverwaltung denkbar:

1. Es wird keine Aktualisierung der Dateneinträge vorgenommen.
2. Die Einträge der Arbeitslistenverwaltung werden aktualisiert.

Die erste Variante hat den Vorteil, daß der für den normalen Betrieb zusätzliche Aufwand der Aktualisierung der Arbeitslistenverwaltung entfällt. Darüber entsteht kein zusätzlicher Aufwand für die Implementierung einer entsprechenden Anpassungskomponente. Allerdings werden hier veraltete Workitem-Zuordnungen und damit veraltete Arbeitslisten-Einträge in Kauf genommen. Dadurch kann es, wie oben gezeigt (vgl. Beispiel 14), zu sicherheitsrelevanten Zugriffsrechtverletzungen oder zu sonstigem Fehlverhalten des Systems kommen.

Wenn beispielsweise ein Workitem von einem Bearbeiter bereits selektiert wurde, steht es *nur* noch in dessen Arbeitsliste. Was passiert, wenn zum Beispiel genau dieser Bearbeiter die Organisation(-seinheit) verläßt und dadurch die Berechtigung zur Durchführung der entsprechenden Aktivität wegfällt? Soll die Aufgabe dann automatisch an andere Bearbeiter delegiert werden? Soll sie abgebrochen werden? Oder muß der betroffene Bearbeiter sie noch selbst beenden, bevor er geht? Das könnte aber gerade bei langfristigen Aufgaben unmöglich sein, so daß hier eine Aktualisierung der Workitem-Zuordnung infolge der Organisationsmodell-Änderung unerlässlich ist.

Wird die Aktualisierung in der Arbeitslistenverwaltung nach der zweiten Variante vorgenommen, so ist das - wie oben dargelegt - mit einem enormen Aufwand und der umfangreichen Restrukturierung der Arbeitslisten-Datenstrukturen verbunden. Der entscheidende Vorteil ist aber, daß der Server auf einem aktuellen Stand ist, was insbesondere für sicherheitskritische Anwendungen von Bedeutung ist.

Wie diese notwendigen Aktualisierungen vorgenommen werden können und wie der dafür notwendige Aufwand minimiert werden kann, soll im folgenden genauer betrachtet werden.

### 6.3.2 Aktualisierung der Bearbeiterformeln

Zunächst stehen die auf Aktivitäteninstanz-Ebene der Arbeitslistenverwaltung gespeicherten, aktualisierten Bearbeiterformeln im Vordergrund (vgl. Abbildung 14). Diese müssen angepaßt werden, wenn sich die zugrundeliegende Bearbeiterformel im Prozeßmodell verändert hat. Diese Anpassung kann durch einfache Ersetzungstechniken erfolgen. Zu beachten ist dabei, daß einzelne Bearbeiterformeln in der Arbeitslistenverwaltung *überschrieben* (vgl. Abschnitt 6.1.2) worden sein können.<sup>22</sup> Diese auf Instanzebene modifizierte Bearbeiterformeln dürfen natürlich nicht automatisch angepaßt werden. Sie sollten ggf. durch den Prozeßverantwortlichen überprüft werden, damit die ursprüngliche Absicht der Modifikation erhalten bleibt und Lost Updates vermieden werden.

Um den Aufwand für die Anpassung der aktualisierten Bearbeiterformeln auf Aktivitäteninstanz-Ebene möglichst gering zu halten, kann die Semantik der Änderung von betroffenen Bearbeiterformeln ausgenutzt werden.

Ausdrücke, die in Bearbeiterformeln *ergänzt* werden, haben eine bestimmte Wirkung auf die Bearbeitermenge der Formel.

(Zusätzliche) AND- bzw. AND NOT-Ausdrücke schränken die Menge potentieller Bearbeiter ein. Wenn diese Ausdrücke in den Bearbeiterformeln der Arbeitslistenverwaltung nicht berücksichtigt werden, kann es in der Folge zu Zugriffsrechtverletzungen kommen, da das System von einer größeren Bearbeitermenge ausgeht, als tatsächlich vorgesehen ist. Wenn das nicht toleriert werden kann, müssen die betroffenen Bearbeiterformeln angepaßt werden (Beispiele 15, 16)<sup>23</sup>.

#### **Beispiel 15 zusätzliche AND-Verknüpfung in einer Bearbeiterformel (BF)**

ursprüngliche BF: (OE = *Behandlungsbereich*)

geänderte BF: (OE = *Behandlungsbereich*) AND (Rolle = *Arzt*)

Wenn "AND (Rolle = *Arzt*)" ignoriert wird, wird die Aufgabe nicht nur dem *Praxis-Arzt*, sondern auch dem *1.Arzthelfer* zugewiesen, der dafür jedoch nicht mehr berechtigt ist.

#### **Beispiel 16 zusätzliche AND NOT-Verknüpfungen in der Bearbeiterformel (BF)**

ursprüngliche BF: (OE = *Behandlungsbereich*)

geänderte BF: (OE = *Behandlungsbereich*) AND NOT (Rolle = *Arzt*)

Wenn "AND NOT (Rolle = *Arzt*)" ignoriert wird, wird die Aufgabe nicht nur dem *1.Arzthelfer*, sondern auch dem *Praxis-Arzt* zugewiesen, der dafür nicht mehr vorgesehen ist.

OR-Ausdrücke dagegen erweitern die potentielle Bearbeitermenge. In diesen Fällen könnte toleriert werden, daß die angenommene Bearbeitermenge kleiner ist, als die tatsächlich vorgesehene. Die Anpassung der Bearbeiterformel in der

<sup>22</sup> zum Beispiel durch ein *Flag* gekennzeichnet

<sup>23</sup> Diese Beispiele beziehen sich auf die ärztliche Praxis aus Beispiel 3, Abschnitt 2.2.2.

Arbeitslistenverwaltung ist dann nicht zwingend notwendig, sofern potentielle Bearbeiter mit der veralteten Formel ermittelt werden können (Beispiel 17)<sup>24</sup>.

#### **Beispiel 17 zusätzliche OR-Verknüpfungen in der Bearbeiterformel (BF)**

ursprüngliche BF: (OE = *Verwaltung*)

geänderte BF: (OE = *Verwaltung*) OR (Rolle = *Arzt*)

Wenn "OR (Rolle = *Arzt*)" ignoriert wird, wird die Aufgabe nur dem *2. Arzthelfer* zugewiesen, obwohl inzwischen auch der *Praxis-Arzt* dafür vorgesehen ist. Trotzdem kann die Aufgabe ausgeführt werden.

Werden AND- bzw. AND NOT-Ausdrücke aus Bearbeiterformeln *entfernt*, so bewirkt diese Änderung eine (potentielle) Vergrößerung der Bearbeitermengen. In diesen Fällen könnte wiederum toleriert werden, daß die angenommene Bearbeitermenge kleiner ist, als die tatsächlich vorgesehene. Die entsprechenden Bearbeiterformel auf Aktivitäteninstanz-Ebene in der Arbeitslistenverwaltung müssen nicht zwingend angepaßt werden.

Werden hingegen OR-Ausdrücke aus Bearbeiterformeln *entfernt*, so kann dadurch die Menge potentieller Bearbeiter eingeschränkt werden. Um Zugriffsrechtverletzungen zu vermeiden, sollten die aktualisierten Bearbeiterformeln in der Arbeitslistenverwaltung aktualisiert werden.

Nach der Anpassung der Bearbeiterformeln in der Arbeitslistenverwaltung steht nun die Aktualisierung der Workitem-Zuordnungen im Mittelpunkt der Betrachtungen.

### **6.3.3 Aktualisierung der Workitem-Zuordnung**

In der Arbeitslistenverwaltung des Servers sind alle Aktivitäten registriert, die sich in den Zuständen ACTIVATED, SELECTED oder RUNNING befinden (vgl. Abbildung 14). Aktivitäten, welche erst nach einer organisatorischen Veränderung aktiviert werden, referenzieren von vorherein das aktuelle Organisationsmodell bzw. eine aktuelle Bearbeiterformel, sofern diese angepaßt wurde. Wie aber kann mit den in der Arbeitslistenverwaltung bereits registrierten Aktivitäten verfahren werden? Die Diskussion der prinzipiellen Lösungsvarianten erfolgt nun anhand der genannten Zustände einer registrierten Aktivität.

#### **a) ACTIVATED**

Aktivierte Aktivitäten stehen erst zur Auswahl an, sind also weder ausgewählt oder gestartet worden. Sie werden abhängig von der zugrundeliegenden Bearbeiterformel einem oder mehreren Bearbeitern zugewiesen. In diesem Fall ist neben der aufwendigen, sofortigen Aktualisierung (*immediate worklist update*) eine verzögerte Aktualisierung der Workitem-Zuordnungen (*delayed/deferred worklist update*) ohne wesentliche Nachteile möglich.

---

<sup>24</sup> Das Beispiel bezieht sich auf die ärztliche Praxis aus Beispiel 3, Abschnitt 2.2.2.

Wenn eine Aktivität *mehreren* Bearbeitern als Workitem zugewiesen wurde und ein Mitarbeiter (bzw. Stelle) infolge einer Organisationsmodell-Änderung aus der Bearbeitermenge dieser Aktivität gelöscht wird, enthält diese Menge dennoch Bearbeiter. Damit kann das Workitem im weiteren Verlauf selektiert und bearbeitet werden. Gleiches gilt, wenn die Menge potentieller Mitarbeiter sich durch die Änderung vergrößert.

In diesen Fällen kann der Workflow weiter ausgeführt werden, ohne ins Stocken zu geraten bzw. ohne daß Zugriffsrechte verletzt werden - obwohl noch keine Aktualisierung der Workitemzuordnungen stattfand.

Wenn allerdings die Arbeitslistenverwaltung erst *verzögert* nach einer Organisationsmodell- oder Bearbeiterformel-Änderung (automatisch) aktualisiert wird, muß jedesmal beim Zugriff auf ein Workitem geprüft werden, ob der Bearbeiter tatsächlich berechtigt ist, diese Aufgabe auszuwählen. Dieser zusätzliche Aufwand wird bei der *sofortigen* Aktualisierung der Workitem-Zuordnungen vermieden. Hier kann sich das System auf die Aktualität der Einträge verlassen und muß beim Zugriff auf Workitems nicht explizit die Berechtigung des Bearbeiters überprüfen.

So eine Überprüfung auf Aktualität kann über einfache Zeitstempelverfahren realisiert werden. Die Workitem-Zuordnungen und die aktualisierten Bearbeiterformeln der Arbeitslistenverwaltung (vgl. Abbildung 14) werden mit Zeitstempeln versehen, die beim Zugriff auf die Workitems verglichen werden. Ist der Zeitstempel der Workitem-Zuordnungen älter als der Zeitstempel der Bearbeiterformel, muß die Zuordnung Neuberechnet werden.

Wurde ein Workitem nur einem Bearbeiter zugewiesen, der nach einer Veränderung nicht mehr berechtigt ist, die Aufgabe auszuwählen, sollte die Zuordnung zügig aktualisiert werden, um die Aktivität nicht zu blockieren. Das setzt allerdings voraus, daß dem Workflow-Management-System entsprechende Informationen vorliegen.

Eine weitere Möglichkeit ist, die Workitem-Zuordnungen in bestimmten Abständen bzw. bei freier Kapazität des Systems zu aktualisieren, wenn also gerade keine oder wenige Zugriffe auf den Server stattfinden.

#### **b) SELECTED**

Ausgewählte Aktivitäten sind zwar noch nicht gestartet worden, sie werden aber nur *genau einem* Bearbeitern zugewiesen. Wenn dieser Bearbeiter nach einer Änderung des Organisationsmodells oder der Bearbeiterformel nicht mehr berechtigt ist, diese Aufgabe auszuführen, gibt es folgendes Problem: Der betroffene Bearbeiter darf die Aufgabe nicht ausführen, gleichzeitig kann aber auch kein anderer berechtigter Bearbeiter die Aufgabe selektieren, da sie aktuell reserviert ist.

Eine Möglichkeit wäre es zu gestatten, daß der nicht berechtigte Bearbeiter das Workitem behält und ausführen darf. Das ist jedoch in vielen Fällen problematisch, weil sicherheitsrelevante Einschränkungen verletzt werden oder zum Beispiel der

betreffende Bearbeiter in der Organisation gar nicht mehr vorhanden ist und die Aufgabe demzufolge dauerhaft blockiert wird.

Eine andere Möglichkeit wäre es, den nicht mehr berechtigten Mitarbeiter über die Änderung seines Zugriffsrechts zu informieren und die Freigabe der Aufgabe zur Neuordnung zu fordern (bzw. durchzusetzen).

### c) RUNNING

Laufende Aktivitäten sind ebenfalls *genau einem* Bearbeiter zugewiesen. Wenn dieser Bearbeiter nach einer Änderung des Organisationsmodells oder der Bearbeiterformel nicht mehr berechtigt ist, diese Aufgabe auszuführen, gibt es ein ähnliches Problem wie bei selektierten Aktivitäten: Der betroffene Bearbeiter darf die Aufgabe nicht weiter ausführen, gleichzeitig kann auch kein berechtigter Bearbeiter die Aufgabe übernehmen.

Wenn die Workitem-Zuordnung nicht aktualisiert wird, bleibt die Aufgabe bei dem nicht mehr berechtigten Bearbeiter. Dadurch werden Ausführungsrechte verletzt, aber die Aufgabe kann eventuell zügig beendet werden. Problematisch ist diese Strategie bei langandauernden Aufgaben oder falls der Mitarbeiter die Organisation verläßt.

In diesen Fällen ist es sinnvoll, dem nunmehr unberechtigten Bearbeiter eine Nachricht über die Änderung zu senden, damit er die laufende Aktivität weiter delegieren bzw. an den Prozeßverantwortlichen zur Neuordnung zurückgeben kann. Dazu muß allerdings das Workflow-Management-System entsprechende Möglichkeiten zur Suspendierung oder Delegation von Aufgaben bieten.

Im nächsten Abschnitt soll nun kurz geklärt werden, welche konkreten Organisationsmodell-Änderungen neben Änderungen an Bearbeiterformeln für die Arbeitslistenverwaltung zur Laufzeit kritisch werden können.

## **6.4 Identifikation kritischer Organisationsmodell-Änderungen**

Um zu beurteilen, nach welchen Organisationsmodell-Änderungen Workitem-Zuordnungen überhaupt aktualisiert werden müssen, soll zunächst betrachtet werden, welche Auswirkungen bestimmte Änderungen auf die Mengen potentieller Bearbeiter haben.

Wie in Beispiel 8 (Abschnitt 5.2.2) gezeigt wurde, können Effekte organisatorischer Änderungen auf Bearbeiterzuordnungen und auf die zugehörigen Bearbeitermengen durchaus unterschiedlich sein. Um die Aktualität der Workitem-Zuordnungen zu bestimmen, ist es deshalb nicht ausreichend, nur zu prüfen, ob die Bearbeiterformeln auf Schemaebene angepaßt bzw. verändert wurden. Auch Organisationsmodell-Änderungen, die keine Anpassungen der Bearbeiterformeln zur Folge hatten, können die Bearbeitermengen von Aktivitäten verändern.

Ziel der Auflösung der Bearbeiterformeln sind potentielle Bearbeitermengen oder - präziser - eine Menge von *Stellen* potentieller Bearbeiter.

Je nach Semantik der Bearbeiterzuordnungen<sup>25</sup> sind ganz bestimmte Relationstypen und Entitätstypen aus dem Organisationsmodell<sup>26</sup> an der Auflösung der Ausdrücke involviert, bis über diesen "Pfad" die gesuchten Stellen erreicht sind. Dieser Zusammenhang ist in Beispiel 18 verdeutlicht.

**Beispiel 18: (an Auflösung der Bearbeiterformeln beteiligte Entitäts- und Relationstypen)**

BF: (OG = *ärztl. Praxis*)

Es werden alle Stelle gesucht, die zur angegebenen *Organisationsgruppe* (OG) gehören. Dazu müssen zunächst alle *Organisationseinheiten* (OE) gefunden werden, die zu dieser Organisationsgruppe gehören. Anschließend werden die *Stellen* (S) ermittelt, die zu den entsprechenden Organisationseinheiten gehören. An der Auflösung dieser einfachen Bearbeiterformel sind also die Entitätstypen OG, OE und S sowie die Relationstypen (OE, OG, gehört\_zu) und (S, OE, gehört\_zu) beteiligt.

Tabelle 14 zeigt die wichtigsten Bearbeiterzuordnungen und die an ihrer Auflösung beteiligten Entitäts- und Relationstypen.

Wenn also Entitäten oder Relationen eines Organisationsmodells verändert werden, können von solchen Änderungen nur ganz bestimmte Bearbeiterzuordnungen (bzw. -formeln) und die zugehörigen Bearbeitermengen betroffen sein.

So spielt es für die Bearbeiterformel (OG = *ärztl. Praxis*) aus Beispiel 18 keine Rolle, ob der Mitarbeiter *Brandt* eine neue Fähigkeit zugeordnet bekommt, weil für ihre Auswertung weder Mitarbeiter (MA), Fähigkeit (F) noch die zugehörige Relation vom Typ (MA, F, hat) relevant sind. Eine solche Änderung des Organisationsmodells hat also keinerlei Auswirkung auf die Bearbeiterformel (OG = *ärztl. Praxis*) und die zugehörige Bearbeitermenge. In diesem Fall muß eine auf dieser Bearbeiterformel basierende Workitemzuordnung nicht Neuberechnet werden.

Wird dagegen eine neue Stelle der Organisationseinheit Verwaltung zugeordnet, so hat diese organisatorische Änderung Einfluß auf die Bearbeitermenge der Bearbeiterformel (OG = *ärztl. Praxis*), obwohl die von der Bearbeiterformel referenzierte Organisationsgruppe selbst unverändert bleibt. An der Auswertung der (unveränderten) Bearbeiterformel (OG = *ärztl. Praxis*) sind sowohl Entitäten der Typen Stelle und Organisationseinheit als auch Relationen vom Typ (S, OE, gehört\_zu) beteiligt (vgl. Tabelle 14).

Auf diese Weise kann enger eingekreist werden, welche Bearbeitermengen nach einer Organisationsmodell-Änderung neu berechnet werden müssen. Doch nicht jede Änderungsoperation des Organisationsmodells hat Auswirkung auf Bearbeitermenge von Bearbeiterzuordnungen (bzw. -formeln).

---

<sup>25</sup> (vgl. Tabelle 12; Abschnitt 5.1)

<sup>26</sup> (vgl. Organisations-Metamodell; Abschnitt 2.1.1)

Bearbeiterzuordnung	involvierte Relationentypen	involvierte Entitätstypen
AG = X	(S, AG, gehört zu)	S, AG
AGl = X	(S, AG, leitet)	S, AG
AGu = X	(S, AG, gehört zu) (S', AG, leitet)	S, AG
F = X	(MA, F, hat) (S, MA, gehört zu)	MA, F, S
MA = X	(S, MA, gehört zu)	S, MA
OE = X	(S, OE, gehört zu)	S, OE
OE+ = X	(S, OE, gehört zu) (OE, OE', ist übergeordnet)	S, OE
OE+n = X	(S, OE, gehört zu) (OE, OE', ist übergeordnet)	S, OE
OE- = X	(S, OE, gehört zu) (OE', OE, ist übergeordnet)	S, OE
OE-n = X	(S, OE, gehört zu) (OE', OE, ist übergeordnet)	S, OE
OEl = X	(S, OE, leitet)	S, OE
OEu = X	(S, OE, gehört zu) (S', OE, leitet)	S, OE
OG = X	(OE, OG, gehört zu) (S, OE, gehört zu)	OE, OG, S
R = X	(S, R, hat)	S, R
R+ = X	(S, R, hat) (R', R, spez.)	S, R
S = X	<i>keine</i>	<i>S (direkt referenziert)</i>
S+ = X	(S, S', ist übergeordnet)	S
S+n = X	(S, S', ist übergeordnet)	S
S- = X	(S', S, ist übergeordnet)	S
S-n = X	(S', S, ist übergeordnet)	S

**Tabelle 14 In die Auflösung von Bearbeiterzuordnungen involvierte Entitäts- und Relationstypen**

Wenn Änderungsoperationen zur Manipulation von Attributen angewendet oder neue Entitäten nur eingefügt<sup>27</sup> werden, hat das generell keinen Einfluß auf Bearbeitermengen, da sie nicht von Bearbeiterzuordnungen referenziert werden können.

Außerdem ist eine Organisationsmodell-Änderung in jedem Fall dann für eine Bearbeitermenge unkritisch, wenn die Änderung andere Entitäts- oder Relationstypen betrifft, als an der Auflösung der zugrundeliegenden Bearbeiterformel beteiligt sind. Dies gilt auch im Zusammenhang mit hierarchischen Beziehungen oder Spezialisierungen von Entitäten (vgl. Tabelle 14).

Wenn dagegen genau die in einer Bearbeiterformel referenzierte Entität durch eine Änderungsoperation des Organisationsmodells manipuliert wird, verändert sich (meist) auch die Bearbeitermenge dieser Bearbeiterformel.

<sup>27</sup> Entität einfügen heißt: noch nicht *durch Relationen* verbunden, zum Beispiel eine isolierte, noch nicht zugeordnete Entität vom Typ Fähigkeit

Außerdem ist eine Veränderung des Organisationsmodells für die Bearbeitermenge dann kritisch, wenn die veränderten Entitäten oder Relationen genau auf dem "Auflösungspfad" der zugrundeliegenden Bearbeiterzuordnung liegen. Dazu müssen die veränderten Entitäten oder Relationen des Organisationsmodells notwendigerweise den gleichen Typ haben, wie die in die Bearbeiterzuordnung involvierten (vgl. Tabelle 14).

In diesen Fällen kommt eine Aktualisierung der Arbeitslisten aufgrund von Organisationsmodell-Änderungen in Frage.

## 6.5 Beispiel für organisatorische Änderungen zur Laufzeit

In diesem Abschnitt sollen an einem einfachen Beispiel verschiedene organisatorische Änderungen im laufenden Betrieb demonstriert werden. Grundlage der Betrachtungen ist das Organisationsmodell einer ärztlichen Praxis (Beispiel 3 aus Abschnitt 2.2.2). Als Beispielprozeß soll eine fiktive Blutuntersuchung dienen. Der zugehörige Ablaufgraph der Aktivitäten, ihre Bearbeiterformeln (BF) und die daraus berechneten Bearbeitermengen (BM) sind in Abbildung 17 dargestellt.

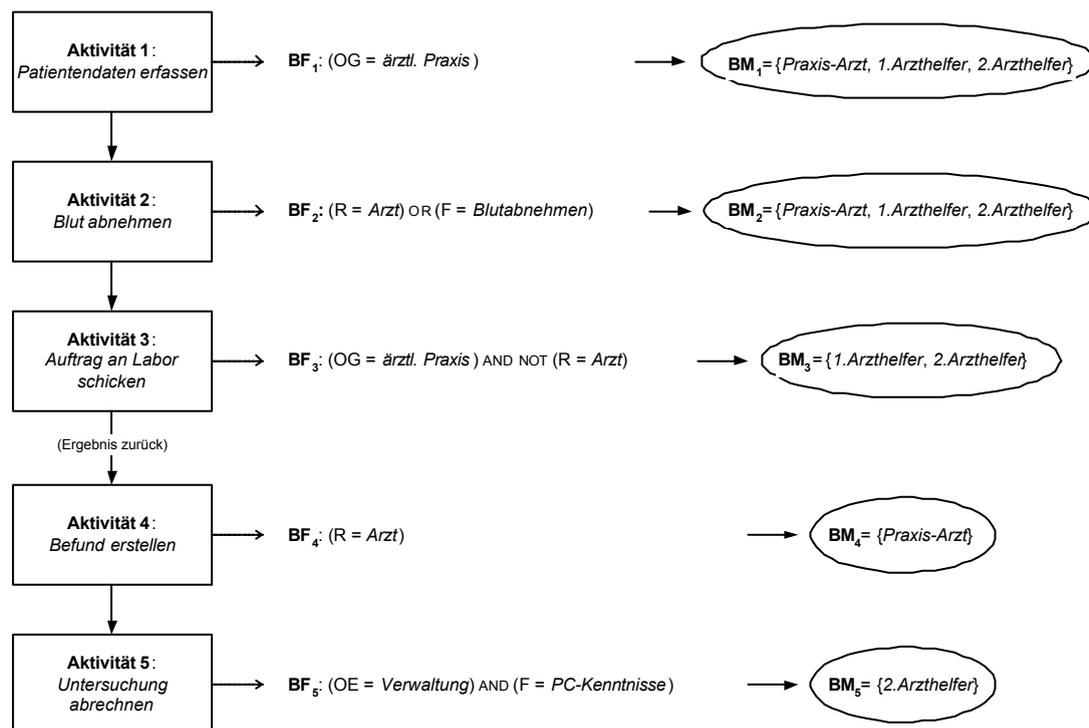


Abbildung 17: Einfaches Modell für Prozeß "Blutabnehmen" mit Bearbeiterformeln (BF) und den daraus bestimmten Bearbeitermengen (vgl. Organisation aus Beispiel 3, Abschnitt 2.2.2)

Im folgenden werden einfache Szenarien dargestellt, bei denen die fokussierte Aktivität sich zunächst im Zustand ACTIVATED befindet, in dem entsprechende Workitems also *mehreren* Bearbeitern angeboten werden können. Anschließend wird dasselbe Szenario mit der Aktivität im Zustand SELECTED betrachtet, wobei ein Workitem dann *genau einem* Bearbeiter zugeordnet wird. Auf die Darstellung des

Szenarios mit der Aktivität im Zustand RUNNING wird verzichtet, da es dem Szenario SELECTED ähnelt.

Die Szenarien orientieren sich an den wichtigsten Varianten von organisatorischen Änderungen (vgl. Abbildung 16):

- Änderung des Organisationsmodells mit Auswirkung auf die Bearbeitermenge *ohne* Anpassung der Bearbeiterformel (Szenarien 1a, 1b)
- Änderung des Organisationsmodells *mit* Anpassung der Bearbeiterformel auf *Typeebene* (Szenarien 2a, 2b)
- Änderung einer aktualisierten Bearbeiterformeln in der Arbeitslistenverwaltung des Servers auf *Instanzebene* (Szenarien 3a, 3b)

### Szenario 1: Organisationsmodell ändern *ohne* Anpassung der Bearbeiterformel

Im *Behandlungs-Bereich* wird eine Stelle *Assistenz-Arzt* geschaffen, zu der die Rolle *Arzt* gehört.

a) Aktivität im Zustand ACTIVATED	
Ausgangssituation	Aktivität 2 ( <i>Blut abnehmen</i> ): ACTIVATED aktualisierte BF <sub>2</sub> : (R = <i>Arzt</i> ) OR (F = <i>Blutabnehmen</i> ) BM <sub>2</sub> = { <i>Praxis-Arzt</i> , 1. <i>Arzthelfer</i> , 2. <i>Arzthelfer</i> }
resultierende BM*	BM <sub>2</sub> * = { <i>Praxis-Arzt</i> , <i>Assistenz-Arzt</i> , 1. <i>Arzthelfer</i> , 2. <i>Arzthelfer</i> } Es gilt: BM <sub>2</sub> * ⊃ BM <sub>2</sub>

Nach dieser additiven Änderung des Organisationsmodells *wächst* die Bearbeitermenge, die Bearbeiterformeln bleiben unverändert. Die Arbeitslistenverwaltung muß nicht aktualisiert werden, da das Workitem bereits (berechtigten) Bearbeitern zugeordnet wurde. Wenn der neue Stelleninhaber sich anmeldet, erhält er automatisch einen Eintrag in seine Arbeitsliste.

b) Aktivität im Zustand SELECTED	
Ausgangssituation	Aktivität 2 ( <i>Blut abnehmen</i> ): SELECTED aktualisierte BF <sub>2</sub> : (S = 1. <i>Arzthelfer</i> ) BM <sub>2</sub> = {1. <i>Arzthelfer</i> }
resultierende BM*	BM <sub>2</sub> * = {1. <i>Arzthelfer</i> } Es gilt: BM <sub>2</sub> * = BM <sub>2</sub>

Da das Workitem in diesem Falle von einem *berechtigten* Bearbeiter ausgewählt wurde, muß die Arbeitslistenverwaltung nicht aktualisiert werden.

Falls nach einer subtraktiven Organisationsmodell-Änderung der Bearbeiter nicht mehr berechtigt wäre (zum Beispiel weil seine Stelle gelöscht wurde), müßte er das Workitem freigeben. Die Freigabe des Workitems kann auch durch den Prozeßverantwortlichen (hier der *Praxis-Arzt*) erfolgen. Die Aktivität wird dann wieder als ACTIVATED gekennzeichnet und die Workitemzuordnung wird neu berechnet. Das betroffene Workitem kann nun wieder selektiert werden.

Alternativ zur Freigabe kann das Workitem auch direkt an andere Bearbeiter delegiert werden.

**Szenario 2: Organisationsmodell ändern mit Anpassung der Bearbeiterformel**

Die *Verwaltung* wird gelöscht, mit ihr wird auch die Stelle 2. *Arzthelfer* entfernt.

a) Aktivität im Zustand ACTIVATED	
Ausgangssituation	Aktivität 5 ( <i>Untersuchung abrechnen</i> ): ACTIVATED aktualisierte BF <sub>5</sub> : (OE = <i>Verwaltung</i> ) AND (F = <i>PC-Kenntnisse</i> ) BM <sub>5</sub> = {2. <i>Arzthelfer</i> }
resultierende BM*	BM <sub>5</sub> * = ∅ Es gilt: BM <sub>5</sub> * ≠ BM <sub>5</sub>

Die Organisationsmodell-Änderung erfordert eine Anpassung der Bearbeiterformel:

(OE = *Verwaltung*) AND (F = *PC-Kenntnisse*).

Sie wird in diesem Beispiel *manuell* angepaßt zu: (F = *PC-Kenntnisse*).

Damit ist die in der Arbeitslistenverwaltung registrierte Bearbeiterformel veraltet: Der 2. *Arzthelfer* ist nicht mehr berechtigt (da die Stelle entfernt wurde), die Untersuchung abzurechnen. Die Zuordnung von Workitems wird auf Basis der angepaßten Bearbeiterformel neu berechnet. Die resultierende Bearbeitermenge lautet dann:

BM<sub>5</sub>\*\* = {1. *Arzthelfer*}

Damit wird das Workitem in die Arbeitsliste des 1. *Arzthelfers* geschrieben und kann wieder ausgewählt werden.

b) Aktivität im Zustand SELECTED	
Ausgangssituation	Aktivität 5 ( <i>Untersuchung abrechnen</i> ): SELECTED aktualisierte BF <sub>5</sub> : (S = 2. <i>Arzthelfer</i> ) BM <sub>5</sub> = {2. <i>Arzthelfer</i> }
resultierende BM*	BM <sub>5</sub> * = ∅ Es gilt: BM <sub>5</sub> * ≠ BM <sub>5</sub>

Nach der Anpassung der Bearbeiterformeln (vgl. Szenario 2a) muß der 2. *Arzthelfer* darüber informiert werden, daß er nicht mehr berechtigt ist, die Untersuchung abzurechnen. Er (bzw. alternativ der Prozeßverantwortliche) muß das Workitem freigeben. Die Aktivität wird als ACTIVATED gekennzeichnet. Dann wird die Workitemzuordnung wie bei Szenario 2a neu berechnet.

**Szenario 3: Bearbeiterformel auf Instanzebene ändern**

Da in diesem konkreten Fall der Patient nur Türkisch kann, muß der Bearbeiter über die Fähigkeit *Türkisch* verfügen. Die Bearbeiterformel wird für genau diese Instanz um den Ausdruck (F = *Türkisch*) ergänzt.

a) Aktivität im Zustand ACTIVATED	
Ausgangssituation	Aktivität 1 ( <i>Patientendaten erfassen</i> ): ACTIVATED aktualisierte BF <sub>1</sub> : (OG = <i>ärztl. Praxis</i> ) BM <sub>1</sub> = { <i>Praxis-Arzt</i> , 1. <i>Arzthelfer</i> , 2. <i>Arzthelfer</i> }
resultierende BM*	BM <sub>1</sub> * = {2. <i>Arzthelfer</i> }
	Es gilt: BM <sub>1</sub> * ⊂ BM <sub>1</sub>

Die *aktualisierte* Bearbeiterformel in der Arbeitslistenverwaltung wird wie folgt überschrieben:

(OG = *ärztl. Praxis*) AND (F = *Türkisch*)

Damit sind die Workitemzuordnungen veraltet und müssen neu berechnet werden. Nur der *2. Arzthelfer* ist berechtigt, bei diesem Patienten die Daten zu erfassen.

<b>b) Aktivität im Zustand SELECTED</b>	
Ausgangssituation	Aktivität 1 ( <i>Patientendaten erfassen</i> ): SELECTED aktualisierte BF <sub>1</sub> : (S = <i>2. Arzthelfer</i> ) BM <sub>1</sub> = { <i>2. Arzthelfer</i> }
resultierende BM*	BM <sub>1</sub> * = { <i>1. Arzthelfer</i> } Es gilt: BM <sub>1</sub> * ≠ BM <sub>1</sub>

Die aktualisierte Bearbeiterformel wurde überschrieben. Damit sind die Workitemzuordnungen veraltet. Der *2. Arzthelfer*, der die Aufgabe selektiert hatte, ist nicht mehr berechtigt, sie auszuführen. Er muß darüber informiert werden und das Workitem freigeben bzw. kann das System serverseitig die Aufgabe entziehen.

Die Workitemzuordnung wird auf Basis der überschriebenen Bearbeiterformel neu berechnet. Die resultierende Bearbeitermenge lautet dann: BM<sub>5</sub>\* = {*1. Arzthelfer*}

Damit wird das Workitem in die Arbeitsliste des *1. Arzthelfers* geschrieben und kann von ihm ausgewählt werden.

Nach einer (kritischen) organisatorischen Änderung kann es bei einmal *gestarteten* Aktivitäten sinnvoll sein, sie von der nunmehr unberechtigten Person dennoch ausführen zu lassen. Gerade bei kurz andauernden Aufgaben, wie in diesem Beispiel einer *Blutuntersuchung*, wäre ein Abbruch und die Neuordnung der Aufgabe unverhältnismäßig aufwendig gegenüber dem Risiko einer unberechtigten Ausführung. Bei langfristigen Aktivitäten ist diese Strategie im allgemeinen nicht möglich, so daß die Aufgabe neu zugeordnet bzw. weiter delegiert werden muß.

## **6.6 Zusammenfassung**

In diesem Abschnitt wurde betrachtet, welche Auswirkungen organisatorische Änderungen auf laufende Workflow-Instanzen haben. Im Mittelpunkt standen dabei die Datenstrukturen der Arbeitslistenverwaltung des Servers und die klientenseitigen Arbeitslisten, die aufgrund solcher Änderungen veraltet sein können.

Es wurde diskutiert, nach welchen organisatorischen Änderungen Aktualisierungen welcher Workflow-Komponenten notwendig werden. Die wichtigsten Varianten der Aktualisierung von Bearbeiterformeln und Workitemzuordnung wurden gegenübergestellt und bewertet.

## 7 Diskussion verwandter Ansätze und Themen

In den vorangegangenen Abschnitten dieser Arbeit wurde ein umfassendes Konzept zur Modellierung von Organisationen sowie zur Änderung dieser Modelle entwickelt.

In den nun folgenden Abschnitten werden verwandte Ansätze und weiterführende Anwendungsthemen diskutiert.

### 7.1 Organisationsmodellierung

Die Modellierung von Organisationen im Kontext von Workflow-Management-Systemen oder anderen Informationssystemen wurde bereits in einigen Arbeiten thematisiert. Meist wurde dabei ein Organisations-Metamodell in Form eines ER-Diagramms mit seinen Konstrukten vorgestellt, allerdings ohne auf Änderungsaspekte weiter einzugehen (Jablonski, Schlundt & Wedekind, 2001; Rosemann & zur Mühlen, 1997).

In der vorliegenden Arbeit wurde als Ausgangspunkt für die Organisationsmodellierung zunächst ebenfalls ein Organisations-Metamodell in Form eines ER-Diagramms gewählt, das zugrundeliegende Konzept wurde anschließend jedoch in eine Mengendarstellung übertragen und um die Definition eines Organisationsmodells erweitert. Auf diese Weise konnten mit Hilfe der Mengendarstellung einfache und komplexe (Mengen-)Operationen zur Änderung von Organisationsmodellen abgeleitet werden.

Zu Organisationsmodell-Änderungen gibt es bisher kaum Beiträge.

Van der Aalst und Jablonski (2000) beleuchten in ihrer Arbeit zwar verschiedene Aspekte<sup>28</sup> von Änderungen im Kontext von Workflow-Management-Systemen - darunter auch Änderungen des Organisationsmodells - bieten aber keine praktischen Lösungen, wie Änderungsoperationen oder Behandlungsansätze zur Crossreferenz-Problematik, an. Sie identifizieren vier allgemeine Typen von Änderungen (Van der Aalst und Jablonski, 2000; S. 269):

1. "*extent*" - neue Entität<sup>29</sup> einfügen,
2. "*reduce*" - Entität löschen
3. "*replace*" - Mischung aus *extent* und *reduce*, Entität ersetzen,
4. "*re-link*" - Entität neuordnen (ohne Entität einzufügen oder zu löschen).

Bezogen auf ein Organisationsmodell können solche Änderungen durch die in Abschnitt 4 der vorliegenden Arbeit beschriebenen elementaren und komplexen Änderungsoperationen problemlos realisiert werden.

---

<sup>28</sup> Die Autoren beziehen sich auf fünf Perspektiven eines Workflows: Prozeß, Organisation, Information, Operation und Integration

<sup>29</sup> Begriff *Entität* ist hier nicht auf das Organisationsmodell beschränkt, sondern umfaßt beispielsweise auch Aktivitäten der Prozeßmodelle.

Klarmann (2001a) identifiziert acht Kategorien struktureller Änderungen von Organisationen, die ebenfalls durch die in Abschnitt 4 beschriebenen Änderungsoperationen realisiert werden können (in Klammern gesetzt):

1. Mitarbeiter wechselt Stelle innerhalb der Organisation (vgl. *move*)
2. Vereinen von Organisationselementen (vgl. *join*)
3. Teilen von Organisationselementen (vgl. *split*)
4. Repartitionierung<sup>30</sup> (durch *createRelation* und *deleteRelation* simulierbar)
5. Löschen von Organisationselementen (vgl. *deleteEntity*)
6. Austausch von Organisationseinheiten<sup>31</sup> (Änderung der Bearbeiterformel)
7. Kreieren von Organisationseinheiten (vgl. *createEntity*)
8. Restrukturierung von Beziehungen zwischen Organisationselementen (durch *createRelation* und *deleteRelation* realisierbar)

Konkrete Änderungsoperationen werden in diesem Beitrag (Klarmann, 2001a) nicht definiert. Der Autor bietet in (Klarmann, 2001b) einen interessanten graphenbasierten Ansatz zur Modellierung von Organisationen an. Er verwendet sowohl für die Beschreibung von Bearbeitern, als auch für die Spezifikation der Anforderungen von Aktivitäten *Konzeptgraphen*, also bipartite, gerichtete Graphen, die aus Konzept- und Beziehungsknoten aufgebaut sind.

Um einen geeigneten Bearbeiter für eine anstehende Aufgaben zu finden, wird derjenige Bearbeiter gesucht, dessen zugehöriger Konzeptgraph strukturell und inhaltlich mit dem Anforderungsgraphen der Aufgabe übereinstimmt. Indem Erfahrungen aus fehlgeschlagenen Versuchen, die manuell gelöst wurden, einfließen, kann die zugrundeliegende Wissensbasis sukzessive erweitert werden. In diesem Sinne wird das System als lernfähig beschrieben.

Die Konzepte von Klarmann (2001a, b) sind eher theoretischer Natur, so daß sie sich für reelle Workflow-Management-Systeme wenig nutzen lassen. Insbesondere kann der Modellierer leicht durch die komplexe Struktur der Konzeptgraphen überfordert werden.

## **7.2 Organisatorische Änderungen zur Laufzeit in existierenden Workflow-Management-Systemen**

Nachdem organisatorische Änderungen bisher eher unter konzeptuellen Aspekten diskutiert wurden, stellt sich die Frage, welche Möglichkeiten existierende Workflow-Management-Systeme tatsächlich in dieser Hinsicht bieten. Exemplarisch sollen an dieser Stelle die beiden gängigen Systeme Staffware 2000 (Staffware plc.) und MQSeries Workflow (IBM) betrachtet werden. Ein systematischer Vergleich dieser beiden Workflow-Management-Systeme findet sich bei Martschat (2001).

---

<sup>30</sup> Wechselseitiges Tauschen der Zuordnung, zum Beispiel:  $a$  und  $b$  gehören zu  $X$ ,  $c$  und  $d$  zu  $Y$ . Nach der Änderung sollen  $a$  und  $c$  zu  $X$ ,  $b$  und  $d$  zu  $Y$  gehören.

<sup>31</sup> Vertauschen der Zuordnung von Prozeßschritten zu Organisationseinheiten

Außerdem wird auch der Prototyp des abteilungseigenen Workflow-Management-Systems ADEPT hinsichtlich der Möglichkeit organisatorischer Änderungen zur Laufzeit kurz bewertet.

### 7.2.1 Staffware 2000

Staffware 2000 bietet zur Modellierung der Organisation die Konstrukte Benutzer (User), Gruppe und Rolle an. Hierarchische Strukturen lassen sich (bis auf die Definition von Supervisoren) nicht abbilden. Aktivitäten können demzufolge Benutzern, Gruppen oder Rollen zugeordnet werden.

Bei der Definition der Bearbeiterzuordnung prüft Staffware sofort, ob die ausgewählte Entität in der Organisation existiert. Wird eine Entität anschließend gelöscht, obwohl sie von einer Aktivität referenziert wird, ist dies ohne Warnhinweis möglich. Die betroffene Aktivität wird dann aufgrund ihrer verwaisten Referenz im Verlauf der Workflows keinem Bearbeiter mehr zugewiesen, der Workflow kommt zum Stehen. Der Administrator und ggf. der Supervisor erhalten eine (unauffällige) Meldung über dieses nicht zugeordnete Workitem. Lediglich der Supervisor hat das Recht, das entsprechende Workitem weiter zu delegieren.

Wird im laufenden Workflow ein Benutzer gelöscht, der noch Einträge in seiner Arbeitsliste hat, so passiert bei diesem Klienten zunächst gar nichts. Er kann weiterhin alle Workitems auswählen, bearbeiten und bekommt auch neue Workitems zugeteilt, solange er am System angemeldet ist. Dadurch läuft der Workflow zwar ohne Unterbrechung weiter, jedoch können durch dieses Prinzip wesentliche Zugriffs- und Ausführungsrechte verletzt werden.

Wird im Verlauf eines Workflows eine Aktivität einem anderen Benutzer zugewiesen<sup>32</sup>, obwohl sie schon in einer Arbeitsliste aufgeführt ist, so kann nur der ursprüngliche Bearbeiter sie ausführen. Die Änderung der Bearbeiterzuordnung wird für diese Instanz nicht mehr ausgeführt.

Interessant ist, daß Bearbeiterzuordnungen, die auf Gruppen basieren, dynamisch ausgewertet werden. Wenn im laufenden Workflow eine Gruppe gelöscht wird, deren Mitglieder entsprechende Arbeitslisteneinträge haben, so werden diese Workitems (leicht zeitverzögert) invalidisiert. Wenn allerdings ein nunmehr unberechtigter Benutzer ein solches Workitem vor oder auch während dieser Latenzzeit bearbeitet, so ist das ohne weiteres möglich.

Insgesamt reagiert Staffware weitgehend stabil auf organisatorische Änderungen. Problematisch sind die Verletzungen von Zugriffs- und Ausführungsrechten sowie die Gefahr von Deadlocks. Darüber hinaus sind die Möglichkeiten zur Modellierung von Organisationen bzw. zur Workitemzuordnung bei Staffware sehr beschränkt.

---

<sup>32</sup> Änderung der Bearbeiterzuordnung auf Schemaebene

### 7.2.2 MQSeries Workflow

MQSeries Workflow bietet zur Modellierung von Organisationen die Elemente Rolle, Organisationseinheit und Person an. Hierarchien können bei Organisationseinheiten modelliert werden. Außerdem können Leitungsfunktionen (Manager, Koordinator) angegeben werden.

Die Zuweisung von Aktivitäten erfolgt entweder statisch an eine konkrete Person oder dynamisch über abhängige oder unabhängige Bearbeiterzuordnungen, die erst zur Laufzeit aufgelöst werden. Zur Definition dieser Bearbeiterzuordnungen können nur Entitäten herangezogen werden, die bereits im Organisationsmodell existieren. Es wird darüber hinaus aber nicht geprüft, ob überhaupt Personen in der Organisation existieren, die die Bedingungen der Bearbeiterausdrücke erfüllen und als potentielle Bearbeiter in Frage kommen. Falls eine Aktivität zur Laufzeit keinem Bearbeiter zugeordnet werden kann, wird der Administrator informiert (Martschat, 2001).

Wenn zur Laufzeit Personen aus dem Organisationsmodell entfernt werden sollen, wird diese Änderung (genauer der Import in die Runtime-Datenbank) abgewiesen, falls die betroffene Person noch am System angemeldet ist oder Einträge in ihrer Arbeitsliste hat. Andernfalls kann diese Änderung jederzeit durchgeführt werden (Martschat, 2001). Ein nach dem Löschen eines Bearbeiters aufgetretenes Fehlverhalten von MQSeries wurde in Beispiel 14 beschrieben.

Änderungen der Bearbeiterzuordnungen auf Schemaebene zur Laufzeit werden auf laufende Prozeßinstanzen nicht propagiert (Martschat, 2001).

Insgesamt reagiert MQSeries eher restriktiv auf organisatorische Änderungen. Es verfügt zwar über Kontrollmechanismen, die bestimmte kritische Änderungen abweisen. Trotzdem kam es zu einem sicherheitskritischen Fehlverhalten, wie von Martschat (2001) beschrieben.

### 7.2.3 ADEPT

Auch beim Prototyp des abteilungseigenen Workflow-Management-Systems ADEPT sind organisatorische Änderungen zur Laufzeit derzeit nicht korrekt durchführbar.

So ist es nicht möglich, im laufenden Betrieb des Workflows einen Mitarbeiter aus der Organisation zu entfernen, der noch Einträge in seiner Arbeitsliste hat. In diesem Fall tritt ein Ausnahmefehler auf. Die Änderung wird nicht ausgeführt.

Das Löschen einer Organisationseinheit, deren Mitglieder noch Einträge in ihren Arbeitslisten haben, wird zwar ausgeführt, führt jedoch zu Inkonsistenzen: Obwohl die Organisationseinheit gelöscht ist, bleiben zugehörige Stellen und Mitarbeiter erhalten. Als Organisationseinheit wird bei ihnen NULL eingetragen, was zu einem Systemfehler führt. Eine rechtzeitige Warnmeldung an den Modellierer erfolgt nicht.

Die Möglichkeiten zu organisatorischen Änderungen sind bei ADEPT also derzeit noch nicht ausreichend gelöst. Hier kann die vorliegende Arbeit mit ihren neuen,

implementierbaren Konzepten einen wichtigen Beitrag zur Weiterentwicklung von ADEPT leisten.

## **7.3 Verwandte und weiterführende Themen**

### **7.3.1 Modellierung von Ressourcen**

In einem Organisationsmodell werden die Strukturen einer Organisation abgebildet, das heißt die Personen, die zu der Organisation gehören, mit ihren Beziehungen untereinander, ihren Fähigkeiten, Rollen usw.

Ein solches Modell kann auch in der Weise erweitert werden, daß sich nicht nur potentielle *menschliche* Agenten, sondern auch weitere Ressourcen wie zum Beispiel Maschinen, Geräte oder auch Räume, die zu der Organisation gehören, abbilden lassen.

Mit Hilfe eines solchen Ressourcenmodells kann die Zuteilung von Ressourcen gesteuert werden.

Zur Mühlen (1999) schlägt zur Modellierung von Ressourcen ein Metamodell vor, das als zentrale Komponente einen Typ Workflow Beteiligter (*Workflow Participant*) hat, der von Workflow-Aktivitäten referenziert werden kann. Ein *Workflow Participant* ist entweder eine Rolle, eine elementare Ressource oder eine Menge von Ressourcen. Ressourcen besitzen ein Attribut, das ihren Typ als *technisch* oder *menschlich* kennzeichnet. Weitere sinnvolle Attribute von Ressourcen können Qualifikationen (zum Beispiel Sprachkenntnisse), Kapazitäten (zum Beispiel die Größe eines Raumes) usw. sein, die von den Aktivitäten auch referenziert werden dürfen. Beschränkungen, wie die Gültigkeit von Vertretungsrechten, können ebenfalls in Form von Attributen modelliert und angesprochen werden.

Interessant ist auch der Vorschlag zur Mühlens (1999), Daten der Verlaufshistorie des Workflows zur *optimierten* Zuordnung von Aktivitäten zu nutzen. So ist aus den Verlaufsdaten (theoretisch) ersichtlich, welche Person wie oft welche Aktivitäten ausgeführt hat (das heißt, ob sie Erfahrung mit solchen Aufgaben hat), wie lange sie durchschnittlich dafür braucht, oder wer früher aufgetretene Fehler im Workflow beheben konnte. Auf diese Weise können die Personen (bzw. allgemeiner: Ressourcen) definierten Niveaustufen (*Levels*) zugeordnet werden. Im Fehlerfall kann damit beispielsweise die Workflow-Engine schnell entscheiden, welcher Workflow-Beteiligte diesen speziellen Fehler beheben kann. Aufgaben mit sehr hoher Priorität könnten dann zum Beispiel sehr erfahrenen Personen zugewiesen werden. In diesem Sinne sind viele erweiterte Anwendungen des Ressourcenmodells denkbar.

### **7.3.2 Autorisierungskonzepte in Workflow-Management-Systemen**

Ein wesentliches Anwendungsgebiet von Organisationsmodellen in Workflow-Management-Systemen ist die Zuweisung von Zugriffs-, Ausführungs- und

Änderungsrechten. In heutigen Workflow-Management-Systemen werden meist *rollenbasierte* Ansätze angewendet. Dabei werden diese Rechte nicht einem festen Bearbeiter sondern einer bestimmten Rolle zugewiesen (NIST<sup>33</sup>; Ferraiolo & Kuhn, 1992). Auf diese Weise wird vermieden, daß bei einem Wechsel der Person auch die Rechtevergabe angepaßt werden muß. Die Rolle ist im Gegensatz zum Mitarbeiter ein relativ stabiles Konstrukt - außerdem ein überschaubares: In Organisationen existieren gewöhnlich weit weniger Rollen als Mitarbeiter.

Nach Bertino, Ferrari und Atluri (1999) reichen rollenbasierte Zugriffsmethoden jedoch allein nicht aus, um bestimmte Einschränkungen (constraints) bei der Autorisierung von Rollen oder Benutzern zu beschreiben. Typische Beispiele für solche Einschränkungen sind exklusive Rechte (*Separation of Duties: "Vier-Augen-Prinzip"*). Nach diesem Konzept dürfen Ausführungs- und Kontrollfunktionen nicht an ein und diesselbe Person vergeben werden, um dem Mißbrauch dieser Rechte vorzubeugen. Bertino, Ferrari und Atluri (1999) zeigen, daß derartige Einschränkungen sowie generell die Zuweisung von Rollen und Benutzern zu den verschiedenen Aufgaben formal als Klauseln in einem logischen Programm ausgedrückt und auf ihre Konsistenz hin überprüft werden können.

Auch bei dem in dieser Arbeit vorgestellten Prinzip der Bearbeiterformeln (Abschnitt 5) lassen sich mit Hilfe von formalen logischen Ausdrücken Regeln für die Zuweisungen von Aktivitäten an Bearbeiter formulieren. Insbesondere durch dynamische Bearbeiterzuordnungen können auch die von Bertino, Ferrari und Atluri (1999) beschriebenen Beschränkungen wie *Separation of Duties* realisiert werden.

In dem abteilungseigenen Workflow-Management-System ADEPT werden Zugriffs-, Ausführungs- und Änderungsrechte mit Hilfe von solchen Bearbeiterformeln (Abschnitt 5) vergeben (Sparr, 2001). Dabei können prinzipiell alle organisatorischen Entitätstypen aus dem Organisationsmodell angesprochen werden: Es sind also neben der rollenbasierten Vergabe von Rechten, der direkten Rechtevergabe an Mitarbeiter, auch indirekten Zuweisungen von Rechten über Organisationseinheiten, Arbeitsgruppen usw. möglich. Diese bei ADEPT verwendete statische und dynamische Zuweisung von Rechten ist also ausgesprochen ausdrucks mächtig.

### 7.3.3 Schemaevolution in Workflow-Management-Systemen

Ein wichtiges Forschungsthema im Kontext von Workflow-Management-Systemen ist die Flexibilisierung dieser Systeme hinsichtlich der Abläufe von Prozessen. Im Vordergrund standen dabei bisher vor allem Ad-hoc-Änderungen von (einzelnen) Workflow-Instanzen sowie planbare Abweichungen vom üblichen Verlauf des Workflows (Reichert, 2001).

---

<sup>33</sup> National Institute of Standards and Technology

Inzwischen ist auch die Schemaevolution von Prozeßmodellen in den Fokus der Forschung gerückt (Rinderle, Reichert & Dadam, 2002). Dabei werden nicht nur einzelne Prozeßinstanzen, sondern die zugrundeliegenden Prozeßmodelle, also Prozeßtypen an neue Bedürfnisse des Anwenders angepaßt.

Nach einer Änderung eines Prozeßschemas sollen bereits erzeugte Prozeßinstanzen ungestört weiterlaufen können. Durch geeignete Versionierungskonzepte kann sichergestellt werden, daß verschiedene Instanzen nach dem neuen und nach dem alten Schema parallel weiterlaufen. Bei langandauernden Prozessen kann es aber unter Umständen (zum Beispiel infolge einer Gesetzesänderung) nicht akzeptabel sein, diese betroffenen Prozeßinstanzen weiter nach dem alten Schema laufen zu lassen. In solchen Fällen muß es möglich sein, das neue Schema auf die laufende Prozeßinstanz zu übertragen. Rinderle, Reichert und Dadam (2002) stellen einen neuen Ansatz vor, mit dem die Verträglichkeitsprüfung von Prozeßinstanzen nach einer Schemaänderung und die anschließende automatische Migration der Instanzen korrekt, konsistent und effizient erfolgen kann.

Auch im Kontext von Organisationsmodellen ist Schemaevolution von Interesse. In dieser Arbeit wurden Änderungsoperationen und -strategien in bezug auf Organisationsmodelle vorgestellt. Prinzipiell könnten auch die zugrundeliegenden Organisations-Metamodelle auf Schemaebene verändert werden. Zu diesem Zwecke wurde hier das *konfigurierte* Organisations-Metamodell (vgl. Abschnitt 2.1.2) eingeführt, mit der Einschränkung, daß keine neuen Entitäts- oder Relationstypen über das (Basis-)Metamodell hinaus eingefügt werden können.

Wenn beispielsweise bei der Zusammenlegung von zwei Kliniken die bisher unterschiedlichen Organisationsmodelle vereint werden müssen, kann es nötig sein, ein gemeinsam gültiges, konfiguriertes Organisations-Metamodell zu bilden.

Generell kann sich nach einer Organisationsmodell-Änderung das Problem ergeben, daß einige laufende Prozeßinstanzen das alte Organisationsmodell referenzieren sollen, während neue Prozeßinstanzen das aktuelle Organisationsmodell referenzieren (beispielsweise auch ab einem bestimmten Stichtag). In diesen Fällen wird es nötig sein, verschiedene Versionen von Organisationsmodellen zu führen. Gerade im Krankenhausumfeld ist zum Beispiel eine genaue Dokumentation der Abläufe und Verantwortlichkeiten über 30 Jahre Pflicht. Zur Lösung dieses Problem können geeignete Versionierungskonzepte beitragen. So schlagen Kradolfer und Geppert (1999) zur Versionierung von Workflow-Schemata eine Baumstruktur vor, bei der abgeleitete Versionen als Sohnknoten der ursprünglichen Version gespeichert werden. Ein vergleichbares Verfahren würde sich auch zur Versionierung von Organisationsmodellen anbieten. Allerdings können Änderungen am Organisationsmodell relativ häufig vorgenommen werden, zum Beispiel durch Personalwechsel, so daß ggf. sehr viele Versionen zu verwalten wären. Deshalb sind auch - je nach Bedarf der Anwendung - Zwischenlösungen denkbar, etwa vergleichbar mit Backups in regelmäßigen Abständen.

## 8 Zusammenfassung und Ausblick

Workflow-Management-Systeme unterstützen die Modellierung, Analyse und Steuerung von Geschäftsprozessen. Um hierbei ein möglichst breites Spektrum an Prozessen abbilden zu können, ist es notwendig, Workflow-Management-Systeme flexibel zu gestalten.

Dabei stand bisher vor allem die flexible Gestaltung von Workflow-Abläufen im Mittelpunkt der Arbeiten (speziell bei ADEPT). Wenig Beachtung wurde dagegen Änderungen der zugrundeliegenden Organisationen und den dadurch notwendig werdenden Anpassungen von Organisationsmodellen geschenkt. Gerade aber in (großen) Organisationen finden häufig Veränderungen, zum Beispiel durch Personalwechsel oder Umstrukturierungen, statt.

In dieser Arbeit wurden deshalb erstmals umfangreiche Konzepte für Organisationsmodell-Änderungen vorgestellt, die weit über einfache rollenbasierte Zugriffskontrollen hinaus gehen.

Die Grundlage für die Arbeit bildet ein formales, mächtiges Organisations-Metamodell, das es wegen seiner umfangreichen Entitäts- und Relationstypen erlaubt, Organisationen vollständig, präzise, semantisch korrekt und konsistent abzubilden. Insbesondere definiert dieses Organisations-Metamodell eine Reihe von Konsistenzbedingungen, die von den daraus abgeleiteten Organisationsmodellen erfüllt werden müssen.

Anhand von einfachen Szenarien wurden wichtige Kategorien von Organisationsmodell-Änderungen illustriert. Im einzelnen sind dies die elementaren Änderungen *Einfügen*, *Löschen* und *Ändern*, sowie die komplexen Änderungen *Vereinigen*, *Teilen* und *Verschieben (Neuzuordnen)* von Organisationselementen.

Es wurde eine vollständige Menge von Operationen entwickelt, die solche Änderungen des Organisationsmodells zulassen. Dabei wird ein korrektes, konsistentes Organisationsmodell durch die Anwendung einer Menge von diesen Änderungsoperationen stets wieder in ein korrektes, konsistentes Organisationsmodell überführt. Als Grundlage für diese Änderungsoperationen wurden Änderungsprimitive formuliert, die sich durch ihre klare (mengenbasierte) Semantik und präzise Vor- und Nachbedingungen auszeichnen. Die darauf aufsetzenden elementaren Änderungsoperationen sind generisch gehalten, so daß sie auf jeden Entitäts- bzw. Relationstyp anwendbar sind. Sie sind damit nicht auf ein bestimmtes Organisations-Metamodell beschränkt. Um darüber hinaus semantisch höherwertige Änderungen zu realisieren, wurden komplexe Änderungsoperationen vorformuliert, die auf Änderungsprimitiven und/oder elementaren Änderungsoperationen aufsetzen. Die Einhaltung der geforderten Korrektheits- und Konsistenzkriterien wurde durch ein umfangreiches Prüfkonzept sichergestellt.

Da zwischen dem Organisationsmodell und anderen Komponenten eines Workflow-Management-Systems zahlreiche Querbezüge (Crossreferenzen) bestehen, muß

beachtet werden, daß diese Referenzen oder die daraus abgeleiteten Datenstrukturen bei unkontrollierten Änderungen des Organisationsmodells verwaist bzw. nicht mehr aktuell wären. Insbesondere könnte die Zuordnung von aktivierten Workflow-Aktivitäten zu Bearbeitern von solchen Änderungen betroffen sein. In der Folge werden Aktivitäten möglicherweise falsch oder gar nicht mehr zugewiesen, wodurch Sicherheitsbestimmungen verletzt werden oder - noch schlimmer - der gesamte Workflow ins Stocken geraten kann.

Zur Behandlung dieser Probleme wurde zunächst der *statische* Fall betrachtet, bei dem noch keine Workflow-Instanzen berücksichtigt werden. Im Mittelpunkt standen hier die ein konkretes Organisationsmodell referenzierenden Bearbeiterformeln. Es wurden Lösungsansätze geboten, welche - je nach Semantik der Änderung - eine automatische, semiautomatische oder manuelle Anpassung der nunmehr veralteten Bearbeiterformeln erlauben. Außerdem wurden Überlegungen dazu angestellt, welche Bearbeiterformeln überhaupt angepaßt werden müssen sowie wann und durch wen das geschehen kann.

Anschließend wurde betrachtet, welche Auswirkungen organisatorische Änderungen im *dynamischen* Fall auf laufende Workflow-Instanzen haben. Der Fokus lag dabei einerseits auf Datenstrukturen der Arbeitslistenverwaltung des Workflow-Servers und andererseits auf den klientenseitigen Arbeitslisten, die aufgrund solcher Änderungen veraltet sein können. Es wurde für verschiedene Arten von organisatorischen Änderungen jeweils diskutiert, welche Workflow-Komponenten wie aktualisiert werden müssen. Die wichtigsten Varianten der Aktualisierung von Bearbeiterformeln und Arbeitslisten wurden dazu gegenübergestellt und bewertet.

Die in dieser Arbeit vorgestellten Konzepte bieten gute Voraussetzungen für eine tatsächliche Umsetzung in Workflow-Management-Systemen. Wie sich diese Systeme unter reellen Bedingungen mit großen Datenmengen und zum Beispiel zeitkritischen Anwendungen verhalten, konnte im Rahmen dieser Arbeit allerdings nicht geprüft werden.

Für weiterführende Arbeiten wäre es sicher interessant, auf Organisationsmodelle zugeschnittene Versionierungskonzepte zu entwickeln, die auch eine Rückverfolgung der Organisationsentwicklung über lange Zeiträume gestattet. Auch die Erweiterung der Organisationsmodelle zur Abbildung von Ressourcen bietet interessante Perspektiven. In diesem Sinne wäre es auch sinnvoll, das Bearbeiterformel-Konzept auf referenzierbare Attribute auszuweiten, um so beispielsweise Anforderungen an Gültigkeitszeiträume, Kapazitäten oder ähnliches explizit ausdrücken zu können.

## Quellenverzeichnis

- Bauer, T., & Dadam, P. (1998): *Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse*. Ulmer Informatik-Berichte, 98-02.
- Bauer, T., Reichert, M. & Fries, T. (2001): *DataStorageLayer: Architektur der OrgModelStorage - white paper - Version 1.0*. Interner Bericht der Abteilung Datenbanken und Informationssysteme, Universität Ulm, Fakultät für Informatik.
- Bertino, E., Ferrari, E. & Atluri, V. (1999): The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Transactions on Information and System Security*, 2-1, S. 65-104.
- Dadam, P. (1996): *Verteilte Datenbanken und Client/Server-Systeme. Grundlagen, Konzepte, Realisierungsformen*. Springer: Berlin, Heidelberg.
- Ferraiolo, D. & Kuhn, R. (1992): Role-Based Access Control *Proceedings of the 15th National Computer Security Conference, Vol II*, S. 554-563. (s. auch <http://hissa.ncsl.nist.gov/rbac/paper/rbac1.html> (Stand 07.07.2002)).
- IBM: *MQSeries Workflow - Product Overview - IBM Software*. <http://www-3.ibm.com/software/ts/mqseries/workflow/> (Stand 07.07.2002)
- Jablonski, S., Böhm, M. & Schulze, W. (Hrsg) (1999): *Workflow Management: Entwicklung von Anwendungen und Systemen; Facetten einer neuen Technologie*. Heidelberg, dpunkt-Verlag.
- Jablonski, S., Schlundt, M. & Wedekind, H. (2001): Eine generische Komponente zur rechnergestützten Nutzung von Aufbauorganisationen. *Informatik Forschung und Entwicklung*, 16, S. 23-34.
- Klarmann, J. (2001a). A Comprehensive Support for Changes in Organizational Models of Workflow Management Systems. In: *Proceedings of the 4th International Conference on Information Systems Modelling (ISM '01)*, S. 165-172.
- Klarmann, J. (2001b). Using Conceptual Graphs for Organization Modelling in Workflow Management Systems. In: *Proceedings of the Conference Professionelles Wissensmanagement (WM 2001)*, S. 19-23.
- Konyen, I. & Reichert, M. (1996): *Organisatorische Aspekte bei der computerbasierten Unterstützung medizinisch-organisatorischer Prozesse*. Interne Ulmer Informatik-Berichte, DBIS 22.

- Konyen, I., Reichert, M. & Schultheiß, B. (1996): *Organisationsstrukturen einer Universitätsklinik am Beispiel der Uni-Frauenklinik Ulm*. Interne Ulmer Informatik-Berichte, DBIS 18.
- Kradolfer, M. & Geppert, A. (1999): Dynamic Workflow Schema Evolution Based on Workflow Type Versioning and Workflow Migration. In: *Proceedings of the International Conference on Cooperative Information (CoopIS '99)*, S. 104-114.
- Leymann, F. & Roller, D. (2000). *Production Workflow: Concepts and techniques*. New Jersey, Prentice-Hall.
- Martschat, U. (2001): *Vergleich und Bewertung von Production Workflow-Management-Systemen*. Diplomarbeit, Universität Ulm, Fakultät für Informatik.
- NIST - National Institute of Standards and Technology: *Role Based Access Control*. <http://csrc.nist.gov/rbac/> (Stand 07.07.2002)
- Oracle Corporation (2002): *Data Sheet Oracle Workflow 11i*. [http://www.oracle.com/appsnet/products/procurement/collateral/ds\\_workflow.html](http://www.oracle.com/appsnet/products/procurement/collateral/ds_workflow.html), (Stand 16.04.2002).
- Reichert, M. (2000): *Dynamische Ablaufänderungen in Workflow-Management-Systemen*. Dissertation, Universität Ulm, Fakultät für Informatik.
- Reichert, M. (2002): *Arbeitslistenverwaltung im ADEPT-Workflow-Management-System*. Internes Arbeitspapier, Abteilung Datenbanken und Informationssysteme, Universität Ulm, Fakultät für Informatik.
- Reichert, M. & Dadam, P. (1998): ADEPTflex - Supporting Dynamic Changes of Workflows Without Losing Control. *Journal of Intelligent Information Systems, Kluwer Academic Publ.*, 10(2), S. 93-129.
- Rinderle, S., Reichert, M. & Dadam, P. (2002): *Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata*. Ulmer Informatik-Berichte, 2002-01.
- Rosemann, M. & zur Mühlen, M. (1997): Modellierung der Aufbauorganisation in Workflow-Management-Systemen: Kritische Bestandsaufnahme und Gestaltungsvorschläge. In: *EMISA Forum, Mitteilungen der GI-Fachgruppe "Entwicklungsmethoden für Informationssysteme und deren Anwendung"*. Bonn, S. 78-86.
- Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T. & Reichert, M. (1996): *Prozeßentwurf am Beispiel eines Ablaufs aus dem OP-Bereich*. Interne Ulmer Informatik-Berichte, DBIS 6.

Sparr, S. (2001): *Ausführungs-, Zugriffs- und Änderungsrechte in adaptiven Prozess-Management-Systemen*. Diplomarbeit, Universität Ulm, Fakultät für Informatik.

Staffware plc.: *Business Process Management and Workflow Solutions*.  
<http://www.staffware.com/> (Stand 07.07.2002)

van der Aalst, W. M. P. & Jablonski, S. (2000): Dealing with workflow change: identification of issues and solutions. *International Journal of Computer Systems Science & Engineering*, 15-5, S. 267-276.

Ultimus GmbH: *Ultimus Workflow Suite*. <http://www.ultimus-workflow.de/phtml/workflowm.htm> (Stand 07.07.2002)

zur Mühlen, M. (1999): Resource Modeling in Workflow Applications. In: Becker, J., zur Mühlen, M. & Rosemann, M. (Hrsg) (1999): *Proceedings of the 1999 Workflow Management Conference, Münster, November 9<sup>th</sup> 1999*, S. 137-153.

## Abbildungsverzeichnis

Abbildung 1	Zusammenhang zwischen Organisationsmodell, Prozeßmodell und Prozeßinstanzen.....	6
Abbildung 2	Organisations-Metamodell von ADEPT.....	10
Abbildung 3	Beispiel für ein einfaches Organisationsmodell einer ärztlichen Praxis.....	18
Abbildung 4	Einordnung der Änderungsoperationen in bezug auf die Modellierungsebene .....	26
Abbildung 5	Zusammenhang zwischen Änderungsprimitiven, elementaren und vordefinierten Änderungsoperationen und Änderungstransaktionen.....	27
Abbildung 6	Beispiel 5a: Arbeitsgruppen vereinigen.....	44
Abbildung 7	Beispiel 5b: Arbeitsgruppe teilen.....	44
Abbildung 8	Beispiel 5c: Stelle anderer Organisationseinheit zuordnen.....	45
Abbildung 9	Problem der Anpassung von Bearbeiterzuordnungen (BZ) nach Änderungen des Organisationsmodells (OM).....	54
Abbildung 10	Verhältnis der Bearbeitermengen vor (BM) und nach (BM*) der Organisationsmodell-Änderung.....	56
Abbildung 11	Varianten der Anpassung von Bearbeiterformeln .....	57
Abbildung 12	Entscheidungspfad zum Anpassungsbedarf von Bearbeiterzuordnungen nach Organisationsmodell-Änderungen.....	59
Abbildung 13	Strategien für die automatische, semi-automatische bzw. manuelle Anpassung der Bearbeiterformeln nach <i>kritischen</i> Organisationsmodell-Änderungen .....	62
Abbildung 14	Datenstruktur zur Verwaltung von Benutzerarbeitslisten bei ADEPT (Reichert, 2002) .....	66
Abbildung 15	Cross-Referenzen zwischen Organisationsmodell, Prozeßmodell sowie Prozeßinstanzen, serverseitigen Arbeitslistenverwaltung und klientenseitigen Arbeitslisteneinträgen zur Laufzeit .....	70
Abbildung 16	Abfolge der Veränderungen vom Organisationsmodell, über Prozeßmodell, Arbeitslisten-Server bis zur Arbeitsliste des Klienten .....	71
Abbildung 17:	Einfaches Modell für Prozeß "Blutabnehmen" mit Bearbeiterformeln (BF) und den daraus bestimmten Bearbeitermengen.....	80

## Tabellenverzeichnis

Tabelle 1	Beziehungstypen und ihre Kardinalitäten .....	12
Tabelle 2	Szenarien personeller Änderungen von Organisationen.....	21
Tabelle 3	Szenarien einfacher struktureller Änderungen von Organisationen .....	21
Tabelle 4	Szenarien komplexer struktureller Änderungen von Organisationen.....	22
Tabelle 5	Änderungskategorien mit Verweis auf Beispiel-Szenarien .....	23
Tabelle 6	Entitätstypen mit semantisch sinnvolle Änderungskategorien.....	23
Tabelle 7	Überblick über relevante Änderungsprimitive .....	28
Tabelle 8	Eigenschaften von Änderungstransaktionen .....	38
Tabelle 9	Mengen und Funktionen für Algorithmus zur Konsistenzsicherung .....	40
Tabelle 10	Entitätstypen und ihre komplexen Änderungen .....	43
Tabelle 11	Muster der Anpassung von Relationen nach Split- oder Join- Operationen.....	45
Tabelle 12	Typen von Bearbeiterzuordnungen.....	52
Tabelle 13	Anpassungsbedarf von Bearbeiterzuordnungen (BZ) nach bestimmten Organisationsmodell-Änderungen .....	58
Tabelle 14	In die Auflösung von Bearbeiterzuordnungen involvierte Entitäts- und Relationstypen.....	79

## Abkürzungsverzeichnis

	Abschnitt
AG	Entitätstyp <i>Arbeitsgruppe</i> eines Organisations-Metamodells ..... 2.1.1
BA	Bearbeiterausdruck..... 5.1
BAttributes	Basis-Attribute eines konfigurierten Organisations-Metamodells ..... 2.1.2
BF	Bearbeiterformel..... 5.1
BM	Bearbeitermenge ..... 5.2.2
BZ	Bearbeiterzuordnung ..... 5.1
EAttributes	Erweiterung der Basis-Attributmenge eines Organisationsmodells ..... 2.2.1
F	Entitätstyp <i>Fähigkeit</i> eines Organisations-Metamodells ..... 2.1.1
MA	Entitätstyp <i>Mitarbeiter</i> eines Organisations-Metamodells..... 2.1.1
OA	Menge der Organisations-Attribute eines Organisationsmodells ..... 2.2.1
OAV	Menge der Wertzuweisungen von Organisations-Attributen eines Organisationsmodells ..... 2.2.1
OE	<sup>1</sup> Entitätstyp <i>Organisationseinheit</i> eines Organisations-Metamodells ..... 2.1.1 <sup>2</sup> Menge von Organisations-Entitäten eines Organisationsmodell ..... 2.2.1
OET	Menge von Organisations-Entitätstypen eines konfigurierten Organisations-Metamodells ..... 2.1.2
OG	Entitätstyp <i>Organisationsgruppe</i> eines Organisations-Metamodells ..... 2.1.1
OM	Organisationsmodell..... 2.2.1
OMM	Organisations-Metamodell ..... 2.1.2
OR	Menge von Organisations-Relationen eines Organisationsmodells ..... 2.2.1
ORT	Menge von Organisations-Relationstypen eines konfigurierten Organisations-Metamodells ..... 2.1.2
OTA	Menge von Organisations-Typattributen eines konfigurierten Organisations-Metamodells ..... 2.1.2
R	Entitätstyp <i>Rolle</i> eines Organisations-Metamodells ..... 2.1.1
S	Entitätstyp <i>Stelle</i> eines Organisations-Metamodells ..... 2.1.1

## Anhang

### A Ergänzung zu Algorithmus 1 (Abschnitt 4.5.2)

Effekte der elementaren Änderungsoperationen auf die Mengen *CreatedEntities*, *DeletedEntities*, *ChangedRelations* und die Funktion *counter(ChangedRelations)*. Betrachtet werden nur die Operationen *createEntity*, *deleteEntity*, *createRelation* und *deleteRelation*, da die Kardinalitätsbeschränkungen für Operationen zur Manipulation von Attributen nicht relevant sind.

Elementare Änderungsoperation	Effekt
<i>createEntity</i> (OM, entitylabel, entitytype)	Sei (entitylabel, entitytype) = oe <i>card_createEntity</i> (oe) <i>CreatedEntities</i> := <i>CreatedEntities</i> ∪ {oe}
<i>createRelation</i> (OM, oe1, oe2, relationtype)	<i>card_createRelation</i> (oe1, oe2, relationtype) /* linke Seite */ if r = (oe1, left, relationtype) ∈ <i>ChangedRelations</i> <i>counter</i> (r) := <i>counter</i> (r) + 1 else <i>ChangedRelations</i> = <i>ChangedRelations</i> ∪ {r} <i>counter</i> (r) = 1 /* rechte Seite */ if r = (oe2, right, relationtype) ∈ <i>ChangedRelations</i> <i>counter</i> (r) := <i>counter</i> (r) + 1 else <i>ChangedRelations</i> = <i>ChangedRelations</i> ∪ {r} <i>counter</i> (r) = 1
<i>deleteRelation</i> (OM, or)	Sei or = (oe1, oe2, relationtype) <i>card_deleteRelation</i> (or) /* linke Seite */ if r = (oe1, left, relationtype) ∈ <i>ChangedRelations</i> <i>counter</i> (r) := <i>counter</i> (r) - 1 else <i>ChangedRelations</i> = <i>ChangedRelations</i> ∪ {r} <i>counter</i> (r) = - 1 /* rechte Seite */ if r = (oe2, right, relationtype) ∈ <i>ChangedRelations</i> <i>counter</i> (r) := <i>counter</i> (r) - 1 else <i>ChangedRelations</i> = <i>ChangedRelations</i> ∪ {r} <i>counter</i> (r) = - 1
<i>deleteEntity</i> (OM, oe)	<i>card_deleteEntity</i> (oe) <i>DeletedEntities</i> := <i>DeletedEntities</i> ∪ {oe} <i>RelationsToBeDeleted</i> : {(oe1, oe2, relationtype)   oe1 = oe ∨ oe2 = oe} forall r ∈ <i>RelationsToBeDeleted</i> do <i>card_deleteRelation</i> (or) done

Für eine Änderungstransaktion mit  $T = op_1 \dots op_n$  lassen sich diese Größen einfach bestimmen:

```

card_Transaction(T)
  for i = 0 to n do
    switch type(opi)
      case createEntity :   card_createEntity
      case createRelation : card_createRelation
      case deleteEntity :   card_deleteEntity
      case deleteRelation : card_deleteRelation
  done

```

## B Anzupassende Relationen bei Split- und Join-Operationen bestimmter Entitätstypen des Organisationsmodells

Entitätstyp	Operation	Anpassen (automatisch)	Entscheiden (manuell)
Org-Gruppe	split	-	(OE,OG, gehört_zu)
Org-Gruppe	join	(OE,OG, gehört_zu)	
Arb-Gruppe	split	-	(S, AG, gehört_zu) (S, AG, leitet)
Arb-Gruppe	join	(S, AG, gehört_zu)	(S, AG, leitet)
Org-Einheit	split	(OE', OE, übergeordnet)	(OE, OE', ist_übergeordnet) (S, OE, gehört_zu) (S, OE, leitet)* (OE, OG, gehört_zu)
Org-Einheit	join	(S, OE, gehört_zu) (OE, OE', ist_übergeordnet)	(S, OE, leitet) (OE, OG, gehört_zu)* (OE', OE, ist_übergeordnet)
Rolle	split	(S, R, besitzt)* (R', R, generalisiert)	(R, R', generalisiert)
Rolle	join	(S, R, besitzt) (R, R', generalisiert)	- -
Stelle	split	(S, OE, gehört_zu) (S', S, fachlich_übergeordnet) (S', S, disz_übergeordnet)	(S, AG, gehört_zu) (S, AG, leitet) (S, OE, leitet) (S, R, besitzt)* (S, MA, hat) (S, S', fachlich_übergeordnet) (S, S', disz_übergeordnet)
Stelle	join	(S, OE, gehört_zu) (S, OE, leitet) (S, S', fachlich_übergeordnet) (S, S', disz_übergeordnet) (S', S, fachlich_übergeordnet) (S', S, disz_übergeordnet) (S, AG, gehört_zu) (S, AG, leitet) (S, MA, hat)	

**fett:** Typ der geänderte Entität \* Semantik beachten