

Ausführungs-, Zugriffs- und Änderungsrechte in adaptiven
Prozess-Management-Systemen

Sonja Sparr

2001

Inhaltsverzeichnis

1	Einleitung	7
1.1	Prozessorientierte Anwendungssysteme	7
1.2	Limitationen existierender Systeme	8
1.3	Problemstellung der Arbeit	9
1.4	Aufgabenstellung der Diplomarbeit	10
1.5	Aufbau der Arbeit	11
2	Stand der Technik	13
2.1	Konzepte zur Rechteverwaltung in Informationssystemen	13
2.1.1	Zugriffskontroll-Listen	13
2.1.2	Rollenbasierte Zugriffskontrolle	15
2.1.3	Weitere Zugriffsformen	15
2.2	Zugriffskontrolle in existierenden PMS	16
2.2.1	Production Workflow	16
2.2.2	Ad-hoc Workflow	22
2.2.3	Vergleich der beschriebenen PMS	24
2.3	Neue Ansätze für die Rechteverwaltung in WfMS	25
3	Szenarien	27
3.1	Das Grundszenario	27
3.2	Mögliche Änderungen im Workflow	29
4	Das ADEPT-Prozess-Management-System	33
4.1	Grundidee von ADEPT	33
4.2	Prozess-und Organisations-Metamodell	33
4.2.1	Prozess-Metamodell	33
4.2.2	Organisations-Metamodell	35
4.2.3	Bearbeiterformeln	37
4.3	Prozessausführung in ADEPT	40
4.3.1	Standardablauf	40
4.3.2	Abweichungen vom Standardablauf	41
5	Modellierungsrechte	55
5.1	Organisationsmodellierung	55
5.1.1	Anlegen von Benutzern	55

5.1.2	Anlegen von Organisationsmodellen	56
5.1.3	Freigeben von Organisationsmodellen	56
5.1.4	Ändern von Organisationsmodellen	56
5.1.5	Anfragen an das Organisationsmodell	57
5.2	Aktivitätenmodellierung	57
5.2.1	Aktivitätenvorlagen erstellen	57
5.2.2	Aktivitätenvorlagen freigeben	58
5.2.3	Aktivitätenvorlagen ändern	58
5.2.4	Aktivitätenvorlagen löschen	58
5.3	Prozessmodellierung	59
5.3.1	Anlegen von Prozessmodellen	59
5.3.2	Freigeben von Prozessmodellen	59
5.3.3	Anfragen zu Prozessmodellen	60
5.3.4	Ändern von Prozessmodellen	60
5.3.5	Löschen von Prozessmodellen	60
6	Ausführungs- und Zugriffsrechte	61
6.1	Prozessbeteiligte Personen	61
6.1.1	Aktive Prozessbeteiligte	61
6.1.2	Passive Prozessbeteiligte	62
6.1.3	Prozessverantwortliche	63
6.2	Ausführungsrechte	63
6.2.1	Rechte für die Manipulation von Prozessinstanzen	64
6.2.2	Ausführung von Aktivitäteninstanzen	67
6.2.3	Vertreterregelungen	68
6.2.4	Ausnahmebehandlungen	70
6.3	Zugriffsrechte	70
6.3.1	Aktueller Workflow-Status	71
6.3.2	Zugriff auf Datenelemente	72
6.3.3	Historie	73
6.3.4	Arbeitslisten	75
7	Anpassungs- und Änderungsrechte	77
7.1	Problemstellung	77
7.2	Planbare Abweichungen	77
7.2.1	Überspringen von Aktivitäten ohne Nachholen	77
7.2.2	Überspringen mit Nachholen	80
7.2.3	Rücksprung über Fehlerkanten	82
7.2.4	Benutzerinitiierte Rücksprünge	83
7.2.5	Fazit	83
7.3	Ad-hoc Änderungen	83
7.3.1	Rechtekonzept für das dynamische Einfügen von Aktivitäten	83
7.3.2	Rechtekonzept für das dynamische Löschen von Aktivitäten	98

<i>INHALTSVERZEICHNIS</i>	5
7.3.3 Rechtekonzept für das dynamische Verschieben von Aktivitäten	107
7.3.4 Weitergehende Betrachtungen	114
7.3.5 Fazit	115
8 Zusammenfassung	117

Kapitel 1

Einleitung

Im Zuge der zunehmenden Globalisierung der Märkte ist es für Unternehmen überlebenswichtig, zunehmend ihre Geschäftsprozesse koordiniert und reibungslos ablaufen zu lassen. Da in einem Unternehmen mit vielen hundert Mitarbeitern nicht jeder einzelne Mitarbeiter den genauen Ablauf der verschiedenen Prozesse kennt, ist es notwendig, die vorhandenen Geschäftsprozesse, auch Arbeitsabläufe genannt, durch ein Computer-System abzubilden. Dadurch wird der einzelne Mitarbeiter unterstützt und entlastet, indem z. B. anstehende Aktivitäten fristgerecht den zuständigen Bearbeitern über Arbeitslisten vorgelegt werden. Ein solches System wird Prozess-Management- oder auch Workflow-Management-System (kurz: PMS bzw. WfMS) genannt [LeRo00].

Da alle Geschäftsprozesse sehr häufig Änderungen der sie umgebenden Umwelt unterworfen sind, ist es für PMS nötig, auch auf diese Änderungen eingehen zu können. Je nach Bereich, in dem das PMS eingesetzt wird, sind Abweichungen vom normalen Ablauf bereits von vornherein bekannt oder sie treten erst während der Bearbeitung eines Prozesses auf. In heutigen PMS, wie MQSeries Workflow [LeRo00] oder Staffware [KaDe96] sind solche Abweichungen vom Standardablauf jedoch nicht vorgesehen bzw. werden nur sehr eingeschränkt unterstützt. Dementsprechend werden von diesen Systemen auch die mit solchen (dynamischen) Abweichungen bzw. Änderungen zusammenhängenden Fragestellungen nicht adressiert. Hierunter fallen, neben Korrektheits- und Konsistenzaspekten [Reic00b], auch Sicherheitsaspekte. Ziel dieser Diplomarbeit ist es, geeignete Konzepte für die Definition und Verwaltung entsprechender Berechtigungen zu finden.

1.1 Prozessorientierte Anwendungssysteme

Aufgrund des in der Geschäftswelt herrschenden Konkurrenzdrucks müssen am Markt angebotene Produkte und Dienstleistungen fortlaufend verbessert sowie die Entwicklungszeiten drastisch reduziert werden (Time-to-Market). Da Materialkosten meist nicht weiter gesenkt werden können, muss zur Verbesserung des Produktangebotes zum einen das Einsparpotenzial zum Zeitpunkt der Produktfertigung besser ausgeschöpft und zum anderen die Erbringung von Dienstleistungen weiter ausgebaut werden. Dies ist jedoch nur möglich, wenn die Fertigungs- oder Dienstleistungsprozesse und die mit ihnen zusammenhängenden Tätigkeiten optimiert ablaufen. Voraussetzung hierfür ist, dass die ‚betrieblichen Strukturen und Abläufe systematisch erfaßt, analysiert und optimiert werden‘ [Reic00b].

Wurden diese Maßnahmen durchgeführt, müssen die optimierten Abläufe in einem Prozess-Management-System modelliert werden, so dass sie intelligent und flexibel koordiniert werden können und damit die beteiligten Akteure optimal unterstützen. Bisher ist es häufig so, dass es in den einzelnen Bereichen einer Organisation heterogene Anwendungssysteme gibt, die

zwar die Daten und Funktionen im jeweiligen Anwendungskontext bereitstellen, die jedoch den globalen Ablauf nicht abbilden und unterstützen. Beispielsweise kann ein Patient in einem Krankenhausinformationssystem (KIS) aufgenommen werden. Dieses bietet oftmals aber keine Unterstützung, um z. B. eine bestimmte Untersuchung in der Radiologie anzufordern, die dann vom Radiologieinformationssystem (RIS) automatisch unterstützt wird (bereichs- bzw. systemübergreifende Prozesse). Erscheint der Patient in der Radiologie, muss der Mitarbeiter die Patientendaten manuell aus dem KIS anfordern und ins RIS übernehmen. Der globale Ablauf ist vielmehr nur in den Köpfen der Mitarbeiter bekannt. Dadurch ergeben sich Schwierigkeiten bei der Einarbeitung neuen Personals und der Ausfall von Mitarbeitern ist problematischer, da das Humankapital sehr hoch ist. Außerdem ist ein Ablauf, der nur in den Köpfen der Mitarbeiter existiert, auch wesentlich fehleranfälliger.

Die Aufgabe eines PMS ist an dieser Stelle leicht ersichtlich. Bei der Aufnahme eines Patienten wird direkt festgelegt, welchen „Weg“ der Patient in der Klinik zu durchlaufen hat. Der Standardablauf, den jeder Patient bei der Aufnahme im Krankenhaus zu bewältigen hat, könnte so aussehen, dass der administrativen Aufnahme ein EKG und eine Blutentnahme folgen. Das PMS würde nun dafür sorgen, dass nach der Aufnahme der Patientendaten, der Patient direkt bei der zentralen Stelle zur Anfertigung des EKGs angemeldet wird und diese Stelle bereits Vorbereitungen treffen kann.

Dadurch dass Mitarbeiter bei der Ausführung ihrer Arbeit ablaufbezogen unterstützt werden – das PMS bietet die richtige Information der richtigen Person zum richtigen Zeitpunkt und verknüpft mit der richtigen Anwendungsfunktion aktiv an – können die anstehenden Aufgaben mithilfe eines PMS effizient durchgeführt werden [DRK00].

1.2 Limitationen existierender Systeme

Wie in [Reic00c] beschrieben, sind Prozess-Management-Systeme nur dann auf breiter Basis sinnvoll einsetzbar, wenn sich Prozessmodelle und -instanzen rasch und kostengünstig an geänderte Ablauf- und Organisationsstrukturen adaptieren lassen. Ein durch den Rechner erzwungenes, starres schematisches Vorgehen bei der täglichen Arbeit stößt dagegen in vielen Bereichen auf wenig Akzeptanz [DRK00]. Im Kontext von Krankenhausprozessen etwa, sind Variationen im Verlauf einer Krankheit oder eines geplanten Behandlungsprozesses üblich und Ausdruck der den Abläufen zugrunde liegenden Variabilität - das unvorhersehbare Ereignis stellt hier in gewisser Weise ein „normales“ Phänomen dar, auf das das klinische Personal flexibel reagieren können muss und auch dazu in der Lage ist, dies zu tun [DRK00].

Aus diesen Gründen sollte es den Anwendern in Ausnahmesituationen möglich sein, in sehr flexibler Form Abweichungen vom geplanten, d. h. dem vormodellierten Workflow vorzunehmen. Für die Praxistauglichkeit von Prozess-Management-Systemen ist es deshalb von enormer Wichtigkeit, dass die Systeminitiative nicht zu stark betont und die Anwender nicht zu sehr in ihrem Handeln eingeschränkt werden. Anders ausgedrückt: Jedes starre System ist - auch bei ansonsten idealer Prozessunterstützung - zum Scheitern verurteilt. Beispielsweise muss bei der Planung einer Untersuchung in Ausnahmesituationen eine Intervention auch ohne die ansonsten übliche elektronische Terminvereinbarung notfallmäßig durchgeführt werden können. Terminverschiebungen, Untersuchungsabsagen oder -wiederholungen können ebenfalls zu Änderungen in vormodellierten Ablaufplänen führen.

Ein prozessorientiertes Anwendungssystem muss berechnete Anwender im Bedarfsfall deshalb die Möglichkeit einräumen, einzelne Prozessschritte oder Folgen von Prozessschritten zu wiederholen, zu ändern oder auszulassen. Darüber hinaus muss es möglich sein, dynamisch Arbeitsschritte zu einem laufenden Workflow hinzuzufügen und sie durch das PMS koordi-

nieren, überwachen und dokumentieren zu lassen. Schließlich müssen Möglichkeiten gegeben sein, laufende Prozesse abubrechen, wenn deren Voraussetzungen nicht mehr erfüllt sind. Auf Systemebene sind dann bereits laufende Schritte evtl. abubrechen oder durchgeführte Schritte zu stornieren.

In Bezug auf diese Flexibilitäts- und Dynamikanforderungen weist die an sich viel versprechende WF-Technologie derzeit noch Schwächen auf. Ad-hoc Abweichungen vom vorgeplanten Ablauf sind - falls überhaupt - nur in engen Grenzen möglich. Auch die Abbildung und Kontrolle von Abhängigkeiten zwischen verschiedenen WF-Instanzen wird nur unzureichend unterstützt. Auch lassen sich zeitliche Abhängigkeiten zwischen Aktivitäten in den heutigen Prozess-Management-Systemen kaum bis gar nicht abbilden.

1.3 Problemstellung der Arbeit

Heutige Prozess-Management-Systeme sehen in der Regel vor, dass die zu unterstützenden Arbeitsabläufe präzise modelliert werden können, so dass zur Laufzeit eines Prozesses keine Abweichungen notwendig werden. In vielen Einsatzbereichen von PMS können jedoch Abweichungen vom Standardablauf auftreten, in einigen Bereichen ist sogar die Abweichung eher der Normalfall als eine Ausnahme. Im Einsatzgebiet eines Krankenhauses etwa kann sich der Gesundheitszustand eines Patienten rapide verschlechtern, so dass eine Untersuchung aktuell gar nicht mehr oder vorgezogen durchgeführt werden muss. Die Benutzer müssen auf diese Abweichungen direkt eingehen können, ohne dass sie in ihrer Arbeit eingeschränkt werden.

Genauso wie sich der Ablauf eines Arbeitsprozesses ändern kann, können auch Veränderungen in Bezug auf ihre Bearbeiter auftreten. Wenn sich, um beim Beispiel des Krankenhauses zu bleiben, der Gesundheitszustand eines Patienten rapide verschlechtert und der behandelnde Arzt aktuell nicht verfügbar ist, muss ein anderer Arzt Einsicht in die Patientenakte erhalten, um ihn optimal versorgen zu können. In diesem Fall könnte darüber diskutiert werden, ob der behandelnde Arzt im Nachhinein die Berechtigung bestätigen muss oder nicht.

Die skizzierten Beispiele zeigen, wie Abweichungen vom Standardablauf häufig auftreten und auf sie geeignet reagiert werden können muss.

Berechtigungen jeglicher Art, d. h. Zugriffs-, Ausführungs- und Änderungsrechte, sollten in einem PMS unabhängig von der Personalstruktur eines Unternehmens definierbar sein (z. B. über Rollen). Dies hat z. B. den Vorteil, dass Berechtigungen nicht entzogen oder neu vergeben werden müssen, wenn sich die Position eines Mitarbeiters im Unternehmen verändert. Es müssen dann lediglich seine Rollenzugehörigkeiten überprüft und gegebenenfalls modifiziert werden, aber nicht etwaige Berechtigungen. Die Verteilung von Berechtigungen wäre somit übersichtlicher und besser zu pflegen, da sie kompakt für alle Benutzer, die die gleiche Rolle besitzen, definiert werden können. Entsprechende Berechtigungen sollten bereits zum Zeitpunkt der Modellierung definiert werden. Das hätte zur Folge, dass immer eine Berechtigung existiert und keine z. B. vergessen werden kann. Auf der Grundlage entsprechender Berechtigungen muss das System direkt entscheiden können, wer berechtigt ist, eine Aktivität zu bearbeiten oder eine Aktivität in eine laufende Instanz einzufügen.

Zur Definition von Berechtigungen bieten sich Bearbeiterformeln oder Bearbeiterzuordnungsausdrücke an, die z. B. direkt mit Prozessaktivitäten oder möglichen Abweichungen kombiniert sind. So ist es leicht möglich, Änderungen an den Berechtigungen durchzuführen, da dies nur an einer Stelle erfolgen muss und nicht für jeden einzelnen Benutzer notwendig wird. Das hat den Vorteil, dass die Gefahr relativ gering ist, eine zu ändernde Stelle zu

übersehen. Die Gewährung und der Entzug von Berechtigungen wird somit pflegbar und kann einfacher durchschaut werden.

Wie oben erwähnt, werden Berechtigungen üblicherweise bereits zum Zeitpunkt der Modellierung definiert. Während der Laufzeit eines Prozesses werden diese immer wieder ausgewertet, so dass eine optimale Berücksichtigung der Unternehmensstruktur gewährleistet ist und ein Benutzer nur die Berechtigungen für einen Prozess erhält, die er auch besitzen darf. Schließlich ist es nicht gewollt, dass jeder Benutzer alles tun darf. Schon aus rein juristischen Gründen darf etwa eine Pflegekraft im Krankenhaus keine Operation anordnen oder durchführen.

Soll eine Prozessvorlage neu in den Vorlagenkatalog (Repository) aufgenommen werden, sollte direkt bei der Modellierung eine prozessverantwortliche Person, mit vordefinierten Zugriffsberechtigungen festgelegt werden. Dies hat zur Folge, dass, wenn eine Tätigkeit keinem Mitarbeiter zugewiesen werden kann, zumindest der Prozessverantwortliche die Tätigkeit entweder selbst bearbeiten oder an einen anderen Mitarbeiter delegieren kann.

1.4 Aufgabenstellung der Diplomarbeit

In heutigen kommerziell verfügbaren Prozess-Management-Systemen wird von einem strikt vorgegebenen Prozessablauf (Production Workflow) ausgegangen, in dem zu Ausführungszeit keine Abweichungen vom Standardablauf vorgesehen sind. Dementsprechend gibt es auch keine Notwendigkeit bzw. Möglichkeit, für Ausnahmesituationen Zugriffs- oder Änderungs-berechtigungen zu definieren.

In dieser Diplomarbeit soll zunächst ein Überblick zu gängigen Zugriffskontrollmechanismen im allgemeinen und in existierenden PMS im speziellen gegeben werden. Die in diesem Zusammenhang angebotenen Beschreibungskonzepte sollen evaluiert und einander vergleichend gegenübergestellt werden.

Um die Notwendigkeit von (dynamischen) Abweichungen vom Standardablauf evident zu machen, soll ein Szenario entwickelt werden. Anhand dieses Beispiels soll gezeigt werden, dass im Zusammenhang mit Ablaufänderungen entsprechende Autorisierungskonzepte unerlässlich sind.

Um wichtige Grundlagen für die Definition von Ausführungs-, Zugriffs- und Änderungsrechten in Prozess-Management-Systemen zu erarbeiten, sollen zuerst grundlegende Aspekte des in der Abteilung Datenbanken und Informationssysteme der Universität Ulm entwickelte des ADEPT-PMS¹ behandelt werden. Hierzu gehören u.a. ein Metamodell für die Definition von Organisationsstrukturen und Personal sowie die Festlegung von Bearbeitermengen. Diese Konzepte sollen einerseits der Definition von Ausführungs- und Zugriffsrechten für die „normale“ Ausführung von Prozess- und Aktivitäteninstanzen dienen, andererseits sollen mit ihnen Anpassungs- und Änderungsrechte definiert werden können.

In Bezug auf Abweichungen vom Standardablauf ist zwischen zwei Arten von Abweichungen zu unterscheiden: Planbaren Abweichungen und Ad-hoc Abweichungen. Für beide Arten müssen angemessene Berechtigungskonzepte definiert werden. Da planbare Abweichungen be-

¹Application Development Based on Encapsulated Premodeled Process Templates. Das ADEPT-PMS besteht aus den Komponenten ADEPT_{base}, ADEPT_{flex}, ADEPT_{time}, ADEPT_{distribution} und ADEPT_{workflow}. Für genauere Informationen steht eine kurze Beschreibung der Komponenten Online unter der URL <http://www.informatik.uni-ulm.de/dbis> [Stand 13.6.2001] zur Verfügung.

reits zum Zeitpunkt der Modellierung bekannt sind und daher auch bei der Modellierung von Prozessabläufen berücksichtigt werden können. Ad-hoc Abweichungen jedoch, sind während der Modellierung einer Prozessvorlage noch nicht bekannt und können daher nicht von vornherein berücksichtigt werden.

Für sämtliche Arten von Berechtigungen soll gelten, dass sie auf eine möglichst einfache und kompakte Weise definiert werden können und während ihrer Lebenszeit auch einfach zu warten sind. Ferner sollen die in Kap. 1.3 beschriebenen Bedingungen und Einschränkungen bei der Definition von Berechtigungen eingehalten werden.

1.5 Aufbau der Arbeit

Die Arbeit gliedert sich wie folgt: In Kapitel 2 werden verschiedene Zugriffskonzepte für Informationssysteme im allgemeinen und für existierende Prozess-Management-Systeme (MQ-Series Workflow, Staffware, WorkParty, ProMInanD) im speziellen erläutert.

Im darauffolgenden Kapitel 3 wird ein einfaches Anwendungsszenario entwickelt, das die Notwendigkeit von Ablaufänderungen und damit zusammenhängende Autorisierungskonzepte evident macht.

Als neuartiges PMS wird in Kapitel 4 das ADEPT-System vorgestellt, das bereits Anpassungen und Abweichungen während zur Laufzeit einer Prozessinstanz vorsieht.

In Kapitel 5 werden Modellierungsrechte beschrieben, die zur Festlegung und Änderung von Organisationsstrukturen sowie von Prozess- und Aktivitätenvorlagen benötigt werden.

Kapitel 6 befasst sich mit den Ausführungs- und Zugriffsrechten für Standardabläufe. Eingegangen wird hier auf prozessbeteiligte Personen, Ausführungsrechte für das im Rahmen einer Prozessausführung zu bearbeitenden Aktivitäten, sowie Rechte für den Zugriff auf prozessrelevante Daten.

Auf der Grundlage der in den vorangehenden Kapiteln beschriebenen Aspekten, bildet Kapitel 7 den Schwerpunkt dieser Arbeit. Betrachtet werden die Anpassungs- und Änderungsrechte bei Abweichungen vom Standardablauf. Hier wird auf Berechtigungen bei planbaren und bei ad-hoc Abweichungen getrennt eingegangen.

Kapitel 8 gibt eine kurze Zusammenfassung dieser Arbeit wieder.

Kapitel 2

Stand der Technik

2.1 Konzepte zur Rechteverwaltung in Informationssystemen

Um Daten vor unerlaubten Zugriffen und Manipulationen zu schützen, existieren verschiedene Verfahren. So gibt es z. B. Zugriffskontroll-Listen, in denen für jeden Benutzer seine Berechtigungen auf verschiedene Objekte festgelegt sind. Als weitere Möglichkeit der Zugriffskontrolle gibt es den rollenbasierten Zugriff. Hier werden die Berechtigungen nicht für jeden Benutzer im einzelnen festgelegt, sondern jeweils für eine Gruppe von Benutzern, die in der Organisation dieselben Kompetenzen oder Eigenschaften besitzen (z. B. Arzt, Sachbearbeiter, Pflegepersonal, Arzthelfer, Systemverwalter).

Im Folgenden werden Zugriffskontroll-Listen und rollenbasierte Zugriffsmethoden genauer beschrieben. Im Anschluss daran werden weitere Zugriffsmethoden kurz erläutert. Im zweiten Teil dieses Kapitels werden die Zugriffskontrollmechanismen in existierenden Prozess-Management-Systemen vorgestellt. Das Kapitel schließt mit der Beschreibung eines neueren Ansatzes für die Zugriffskontrolle in PMS.

2.1.1 Zugriffskontroll-Listen

Die Grundlage für Zugriffskontroll-Listen bildet die Zugriffskontroll-Matrix. Wie in [Bann92] beschrieben, ist sie in Form einer Tabelle aufgebaut, in der die Zeilen einzelnen Subjekten, also Benutzern, zugeordnet sind und die Spalten den Objekten entsprechen. Jeder Tabelleneintrag repräsentiert somit die Menge der Zugriffsrechte eines Subjektes auf ein bestimmtes Objekt.

Tabelle 2.1 zeigt ein Beispiel für eine Zugriffskontroll-Matrix. Die dort verwendeten Zugriffsrechte sind *read* (r), *write* (w), *execute* (x) und *own* (o). Der Eigentümer von Objekten kann Rechte an andere Benutzer weitergeben oder deren Berechtigungen einschränken.

Die Verwendung einer Zugriffskontroll-Matrix macht eine vollständig zentrale Rechteverwaltung und Rechtekontrolle erforderlich, da die Rechteverteilung im System nur aus dieser Matrix ersichtlich ist. Da Benutzer üblicherweise auf die meisten Objekte keinen Zugriff haben, ist diese Zugriffsform in Verbindung mit einer großen Zahl von Benutzern und Objekten nicht effizient. Daher wird die Matrix oftmals nach Zeilen oder Spalten zerlegt.

Wird die Zugriffskontroll-Matrix nach Spalten zerlegt, erhält man die sogenannten Zugriffskontroll-Listen. Daraus ergibt sich, dass für jedes Objekt eine eigene Zugriffskontroll-Liste, auch ACL (*Access Control List*) genannt, existiert. Eine Zugriffskontroll-Liste wird für ein Objekt angelegt, wenn das Objekt erzeugt wird. Der Benutzer, der das Objekt erzeugt hat, wird direkt als Eigentümer festgelegt und kann somit über die Weitergabe bzw. Einschränkung der Rechte bzgl. dieses Objektes verfügen. Mit der Erzeugung eines Objektes

Objekte \rightarrow Subjekte \downarrow	Modul Bibliothek	PASCAL Compiler	Mail.txt	Gehalt.tab	System- Drucker	System Uhr
Mike	rw	x	—	—	w	r
Mary	rw	x	—	—	w	r
Werner	r	x	—	—	w	r
Peter	—	—	—	—	w	r
Susi	r	x	orw	—	w	r
Chef	—	—	r	orw	w	r
System- Manager	rw	rwX	w	—	ow	rw
System	o	o	—	—	w	or

Tabelle 2.1: Beispiel für eine Zugriffskontroll-Matrix (aus [Bann92])

und der zugehörigen ACL kann auch eine Standard-Rechtebelegung für andere Benutzer mit vergeben werden, diese werden jedoch auch vom Eigentümer kontrolliert. Abb. 2.1 gibt ein Beispiel für eine ACL:

ACL_{Objekt}:

[(Subjekt ₁	\mapsto	Rechte _{e₁}),
	(Subjekt ₂	\mapsto	Rechte _{e₂}),
	(Gruppe ₁	\mapsto	Rechte _{e₂}),
	...		
	(Subjekt _k	\mapsto	Rechte _{e_j}]

Abb. 2.1: Beispiel einer einfachen Zugriffskontroll-Liste (aus [Bann92])

Es fällt auf, dass sogar einer Gruppe von Subjekten entsprechende Rechte zugewiesen werden können. Dies ist aber nur in einzelnen Systemen möglich und geht dann in den rollenbasierten Zugriff (s. Kap. 2.1.2) über.

Ein Vorteil einer ACL gegenüber einer Zugriffskontroll-Matrix ist der kleinere Speicherbedarf, da sie nur die Benutzer enthält, die wirklich berechtigt sind, auf das Objekt zuzugreifen, nicht jedoch diejenigen Benutzer, die nicht zum Zugriff berechtigt sind, d. h. ACLs enthalten weniger Einträge. Wird nun von einem Benutzer versucht, auf ein Objekt zuzugreifen, wird der Zugriff anhand der entsprechenden ACL überprüft und freigegeben, wenn die entsprechenden Rechte spezifiziert sind. In den meisten Systemen beschränken sich die Zugriffsrechte auf eine Teilmenge der in [Dada99] aufgeführten Rechte. Zu diesen Rechten gehören u.a.:

- *read*: Lesen des Objektes (Inhalt des Objektes)
- *write*: Schreiben des Objektes
- *change*: Verändern des Objektes
- *delete*: Löschen des Objektes
- *execute*: Ausführen eines (Programm-) Objektes
- *see/search*: Feststellen der Objekt-Existenz

- *control*: Ändern der Zugriffsrechte
- *own*: Eigentümer des Objektes

Auf die ACL eines Objektes kann lediglich der Benutzer direkten Zugriff haben, der die Eigentumsrechte an diesem Objekt besitzt, anderen Benutzern muss der direkte Zugriff auf die ACL unzugänglich gemacht werden.

Ein Nachteil dieser Art der Festlegung der Zugriffsrechte auf ein Objekt ist, dass für jedes Objekt die Berechtigungen für jeden einzelnen Benutzer einzeln gesetzt werden müssen. Es ist nicht möglich die Berechtigungen globaler, z. B. abteilungsbezogen zu setzen.

2.1.2 Rollenbasierte Zugriffskontrolle

Der größte Unterschied der rollenbasierten Zugriffskontrolle, auch RBAC (*Role Based Access Control*) genannt, zu ACLs ist, dass die Berechtigungen nicht an einzelne Benutzer vergeben werden, sondern anhand von Rollen festgemacht werden ([NIST], [FeKu92]). Das bedeutet, jedem Benutzer wird, wie in [FCK95] beschrieben, entsprechend seiner Qualifikation und seinen Verantwortlichkeiten eine oder mehrere Rollen zugeteilt. Eine Rolle besitzt ein oder mehrere Berechtigungen. Die Beziehungen zwischen Benutzern, Rollen und Berechtigungen sind in Abb. 2.2 dargestellt, wobei die Doppelpfeile die Bedeutung einer n:m-Beziehung haben.



Abb. 2.2: Beziehung zwischen Benutzern, Rollen und Berechtigungen (aus [FCK95])

Wird eine neue Applikation oder Anwendungsfunktion in das Unternehmen eingeführt bzw. entfernt, müssen die Zugriffsberechtigungen nicht für jeden einzelnen Benutzer neu festgelegt werden, sondern sie müssen nur für jede Rolle spezifiziert werden. Da es in einem Unternehmen i.d.R. mehr Benutzer als Rollen gibt, reduziert sich der Aufwand für die Definition von Berechtigungen entsprechend. Wird ein neuer Mitarbeiter eingestellt, muss dieser nur den entsprechenden Rollen zugeteilt und nicht eine große Zahl an Zugriffsberechtigungen für ihn angegeben werden. Verändert sich die Position eines Mitarbeiters im Unternehmen, kann seine Rollenzugehörigkeit ebenfalls schnell an die neue Position angepasst werden.

Ein weiterer Beweggrund zur Verwendung von RBAC ist die einfache Darstellung der Unternehmensstruktur mit Hilfe der Rollen [Bark97]. Durch Rollen können neben der Darstellung der Unternehmensstruktur auch hierarchische Beziehungen zwischen Rollen modelliert werden. Hier werden Rechte von einer Rolle auf eine höher liegende Ebene vererbt, d. h. „Rolle A beerbt Rolle B“ bedeutet, dass Tätigkeiten, die von Personen der Rolle A ausgeführt werden dürfen auch von Personen der Rolle B ausführbar sind.

Alles in allem wird durch die Verwendung von RBAC der Aufwand für die Administration von Zugriffsberechtigungen erleichtert und deren Spezifikation flexibler. Nebenbei wird die Einhaltung unternehmensspezifischer Sicherungsmaßnahmen erzwungen.

2.1.3 Weitere Zugriffsformen

Die bisher vorgestellten Zugriffskontrollen sind einfache, zum Teil aber auch sehr aufwendige Methoden, um Daten und Anwendungen vor unbefugtem Zugriff zu schützen. Sie sind leicht durch verschiedene Zusätze zu erweitern. Beispielsweise können beide Verfahren so erweitert

werden, dass der Benutzer nur Daten einsehen kann, die in einem bestimmten Zeitfenster oder an einem bestimmten Terminal erzeugt wurden. Auch sind Verfahren möglich, bei denen der Zugriff auf ein Objekt vom Wert, also Inhalt, des Objektes abhängig ist.

Ein Problem, das von den bisher beschriebenen Verfahren noch nicht berücksichtigt wird, ist die Verwendung ausschließender oder exklusiver Rechte. Das bedeutet, wenn der Benutzer Rolle X besitzt, darf er nicht auch die Rolle Y besitzen, um einen möglichen Missbrauch der Daten oder seiner Position betreiben zu können. So darf z. B. im Krankenhaus auf einer Station nicht die Person für die Bestellung von Medikamenten verantwortlich sein, die in der Verwaltung die Bestellung endgültig freigibt, so dass eine gewisse Kontrollfunktion ausgeübt werden kann. Ein Beispiel für die Realisierung exklusiver Rechte wird in Kap. 2.3 beschrieben.

2.2 Zugriffskontrolle in existierenden PMS

In diesem Abschnitt werden die Zugriffskontrollmechanismen in existierenden Prozess-Management-Systemen näher untersucht.

2.2.1 Production Workflow

Die sogenannten Production Workflow Systeme, wie MQSeries Workflow [LeRo00], Staffware [KaDe96] und WorkParty [Rupi97, RSD97], unterstützen vor allem stark strukturierte repetitive Arbeitsprozesse. Prozesslogik (Kontroll- und Datenfluss) und Anwendungsfunktionen bzw. -code sind bei diesen Systemen relativ strikt voneinander getrennt. Dadurch sind abgebildete Arbeitsabläufe austauschbar bzw. erweiterbar ohne dass der Anwendungscode geändert werden muss. Die große Schwäche dieser Systeme ist ihre Starrheit bzw. nicht vorhandene Fähigkeit, dem Anwender in Ausnahmesituationen die benötigte Flexibilität einzuräumen. Ablaufänderungen durch den Benutzer sind entweder explizit verboten oder wenig gefordert, d. h. die Anwender verhalten sich eher passiv und reagieren nur auf Aufträge, die sie vom Production Workflow System erhalten. Dadurch ergibt sich die Problematik der Änderungen in einem Workflow nicht. Diese Starrheit schränkt, neben anderen fehlenden Fähigkeiten, die breite Einsetzbarkeit von Production Workflow Systemen erheblich ein. Der Fokus dieser Systeme liegt derzeit klar im Banken- und Versicherungsbereich, gefolgt von den Service-orientierten Bereichen (z. B. Call Center). Als Beispiele werden im Folgenden MQSeries Workflow, Staffware und WorkParty beschrieben.

2.2.1.1 MQSeries Workflow

Das IBM Produkt MQSeries Workflow (ehemals: IBM FlowMark [Roll96]) ist seit 1994 kommerziell verfügbar. Prozesse werden in Form von Aktivitätensetzen modelliert, wobei der Kontrollfluss getrennt vom Datenfluss beschrieben wird [LeAl94]. Wie andere Production Workflow Systeme auch, verfügt MQSeries Workflow über eine Komponente zur Modellierung von Organisations- und Personalstrukturen. Auf Basis des für ein Unternehmen definierten Organisationsmodells können für Prozessmodelle Ausführungs- und Zugriffsberechtigungen festgelegt werden. Während der Ausführung eines Workflows können dann die potenziellen Bearbeiter einer Aktivität auf der Grundlage der ihr zugeordneten Bearbeiterformel und des verwendeten Organisationsmodells ermittelt werden. Das Organisations-Metamodell von MQSeries Workflow ist in Abb. 2.3 dargestellt.

Die Modellierung der Organisations- und Personalstruktur beschränkt sich bei MQSeries Workflow auf Organisationseinheiten (OE), Mitarbeiter (MA) und Rollen (R). Einem Mitarbeiter wird eine Vertraulichkeitsstufe (0-9) zugeordnet und er kann maximal einer Organi-

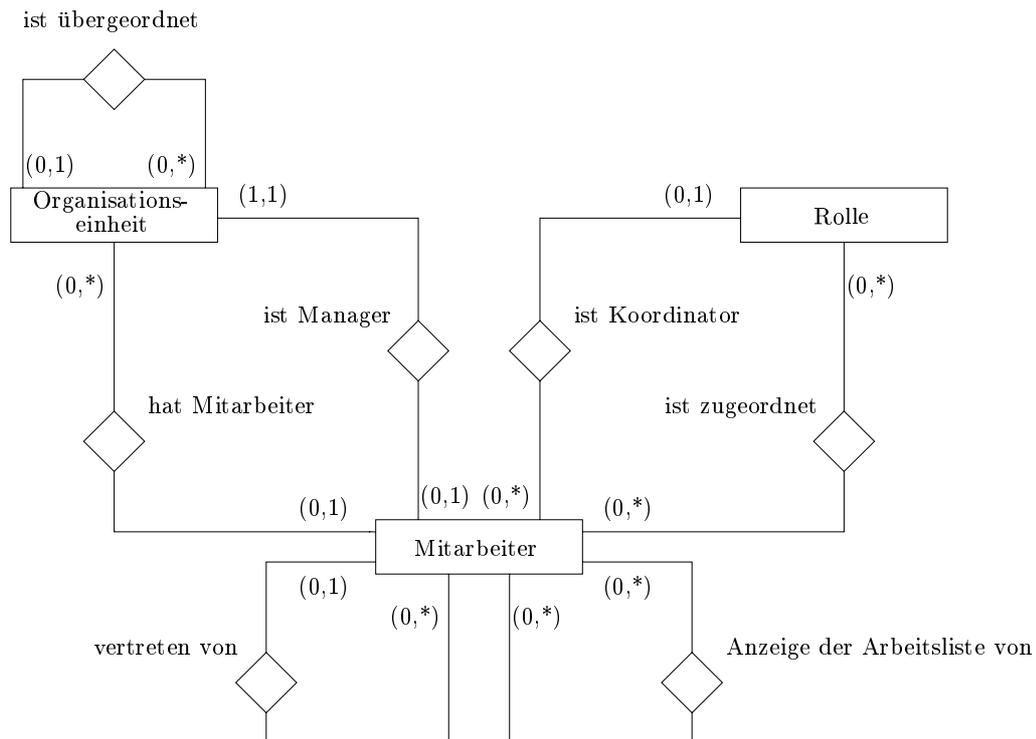


Abb. 2.3: Organisations-Metamodell von MQSeries Workflow (aus [Reic00a])

sationseinheit angehören. Rollen beschreiben eine Menge von Kompetenzen und/oder Fähigkeiten, von denen ein Mitarbeiter mehrere besitzen kann. Ein Krankenhausarzt könnte z. B. die Rollen *Stationsarzt* und *Ambulanzzarzt* einnehmen. Jedem Mitarbeiter kann eine Person als Vertreter zugewiesen werden und er selbst kann für beliebig viele andere Personen ein Vertreter sein. Einem Vertreter werden alle Arbeitslisteneinträge der zu vertretenden Person in seiner Arbeitsliste angezeigt ab dem Zeitpunkt, zu dem die zu vertretende Person im System als abwesend gekennzeichnet ist. Ältere Arbeitslisteneinträge werden jedoch nicht angezeigt. Jeder Mitarbeiter kann dazu berechtigt sein, Prozessvorlagen, Organisationsstrukturen oder Programme zu definieren und Prozessinstanzen von Vorlagen bestimmter Kategorien zu erzeugen. Auch kann ein Mitarbeiter, je nach Berechtigung, auf die Arbeitslisten anderer Mitarbeiter zugreifen. Eine Organisationseinheit kann beliebig vielen Organisationseinheiten übergeordnet, aber nur einer einzigen direkt untergeordnet sein. Jede Organisationseinheit hat genau einen Leiter.

Ein Nachteil dieser Organisationsstruktur ist, dass Mitarbeiter nur einer einzigen Organisationseinheit angehören können. Um beim Beispiel eines Krankenhauses zu bleiben, es ist nicht möglich zu modellieren, dass ein Arzt auf zwei verschiedenen Stationen tätig ist. Auch können Rollen nicht Organisationseinheiten zugeordnet werden, wodurch diese Zuordnung unabhängig von personellen Umstrukturierungen wäre. Ein weiterer Nachteil sind die mangelhaften Konzepte für die Beschreibung von Mitarbeiter-Qualifikationen und Vertreterregelungen. Ersteres erfordert evtl. eine große Zahl von Rollendefinitionen, letzteres erzwingt häufige Änderungen in der Definition von Vertreterregelungen, z. B. beim Ausscheiden eines Mitarbeiters. Beides ist sehr wartungsintensiv. Hinzu kommt, dass Vertreterregelungen nicht in Abhängigkeit von Aufgabenkategorien definierbar sind. Das heißt, der Vertreter einer bestimmten Person vertritt diese in allen Tätigkeitsbereichen. Dies ist für die Praxis nicht immer ausreichend [KRS96].

The screenshot shows a software interface for a medical workstation. The title bar reads 'Stationsarbeitsplatz - Stationsarzt'. The interface is divided into several sections:

- Stationsübersicht (Room Overview):** Shows two rooms, Zimmer 57 and Zimmer 58. Zimmer 57 contains icons for 'Fr. Schiebe' (frakt. Abrasio, Aus OP zurück), 'Fr. Mayer' (Interruptio im OP), and 'Fr. Heinrich' (IVF, In OP bringen). Zimmer 58 contains icons for 'Fr. Apfel' (frakt. Abrasio).
- Einbestellungsplan GYN IV (Friday, July 18, 1997):** A table listing patients, diagnoses, and phone numbers.

Patientin	Diagnose-OP	Tel.Nr.
Fr. Mayer (Abruf)	Interruptio	07551-22456
Fr. Dampf (Jeberpruefung der Hysteroskopie Befunde)	dg/op	
Fr. Heinrich (Abruf in Ausfuehrung)	IVF	
Fr. Apfel (Stationaere Aufnahme)	frakt. Abrasio	07551-22456
Fr. Frenze (Jeberpruefung der Befunde)	IVF	0731-3051
Fr. Schiebe (Rueckmeldung)	frakt. Abrasio	
- Anstehende Aufgaben (Pending Tasks):** A table with columns for Patientin, Tätigkeit, Bemerkung, Ausführen, and Aktualisieren.

Patientin	Tätigkeit	Bemerkung	Ausführen	Aktualisieren
Fr. Schiebe	OP-Bericht erstellen			
Fr. Schiebe	Klinische Entlassung			
Fr. Förster	Überprüfung der Befunde	ängstliche Patientin		
Fr. Hansen	Überprüfung der Befunde			
- Arbeitsliste (Task List):** A yellow box with a red arrow pointing to the 'Anstehende Aufgaben' table.
- Arbeitslisteneinträge (Task List Entries):** A yellow box with a red arrow pointing to the 'Anstehende Aufgaben' table.
- Navigation and Control:** Buttons for 'Maßnahme anordnen', 'OP-Plan', 'Umschalten zu Ambulanz', 'Notfallaufnahme', 'Tag', 'Woche', 'Monat', 'neuer Termin', 'Termin verschieben', 'Termin sperren', 'Termin löschen', and 'Patientenstammdaten'.

Abb. 2.4: Beispiel einer Arbeitsliste (aus [Reic01])

Für Dialogschritte ist es in MQSeries Workflow notwendig, festzulegen, in welchen Benutzerarbeitslisten (s. Abb. 2.4) die Aktivität zur Laufzeit eingetragen werden soll, d. h. es muss festgelegt sein, wem die Aktivität zur Bearbeitung angeboten werden soll.

Zur Festlegung einer Bearbeiterzuordnung gibt es zwei Varianten, *unabhängige* und *abhängige* Bearbeiterzuordnungen. Bei unabhängigen Bearbeiterzuordnungen wird eine Rolle und/oder eine Organisationseinheit angegeben, die potenziellen Bearbeitern der Aktivität zugeordnet sein muss. Zusätzlich kann ein „Mindestlevel“ angegeben werden, den Bearbeiter besitzen müssen. In Bezug auf Organisationseinheiten wird unterschieden, ob nur die Mitarbeiter dieser Organisationseinheit einbezogen werden sollen oder auch die Mitarbeiter untergeordneter Organisationseinheiten. Nachteilig ist, dass Organisationseinheiten nicht ODER-verknüpfbar sind, etwa „OE = Station 2a ODER OE = Ambulanz“.

Im Allgemeinen kann es mehrere Personen geben, die einem Bearbeiterausdruck genügen. Bei einer Bearbeiterzuordnung „OE = Station 2a UND R = Stationsarzt“ (vgl. Abb. 2.5 Aktivität F) etwa, erscheint die Aktivität in den Arbeitslisten aller Mitarbeiter, die zur „Station 2a“ gehören und die Rolle „Stationsarzt“ besitzen.

Alternativ dazu bezieht sich eine abhängige Bearbeiterzuordnung auf eine konkrete Person. Meist nimmt der Bearbeiterausdruck dann Bezug auf die Ausführungshistorie des Workflows (vgl. Abb. 2.5 Aktivität C). MQSeries Workflow unterstützt die folgenden abhängigen Bearbeiterzuordnungen:

- Bearbeiter = Prozessadministrator
- Bearbeiter = Starter des Prozesses oder dessen Vorgesetzter

- Bearbeiter = Starter einer Vorgängeraktivität oder dessen Vorgesetzter
- Bearbeiter \neq Starter der Vorgängeraktivität (Vier-Augen-Prinzip)
- Bearbeiter = Person mit User-ID XY
- Bearbeiter = dynamisch zugewiesene Person

Bei dynamisch zugewiesenen Bearbeitern ruft die Aktivität (vgl. Abb. 2.5 Aktivität E) eine Anwendung s auf, die Daten in einen Ausgabecontainer S schreibt. Hierauf können Folgeaktivitäten (vgl. Abb. 2.5 Aktivitäten G und H) über ihre Eingabecontainer Bezug nehmen.

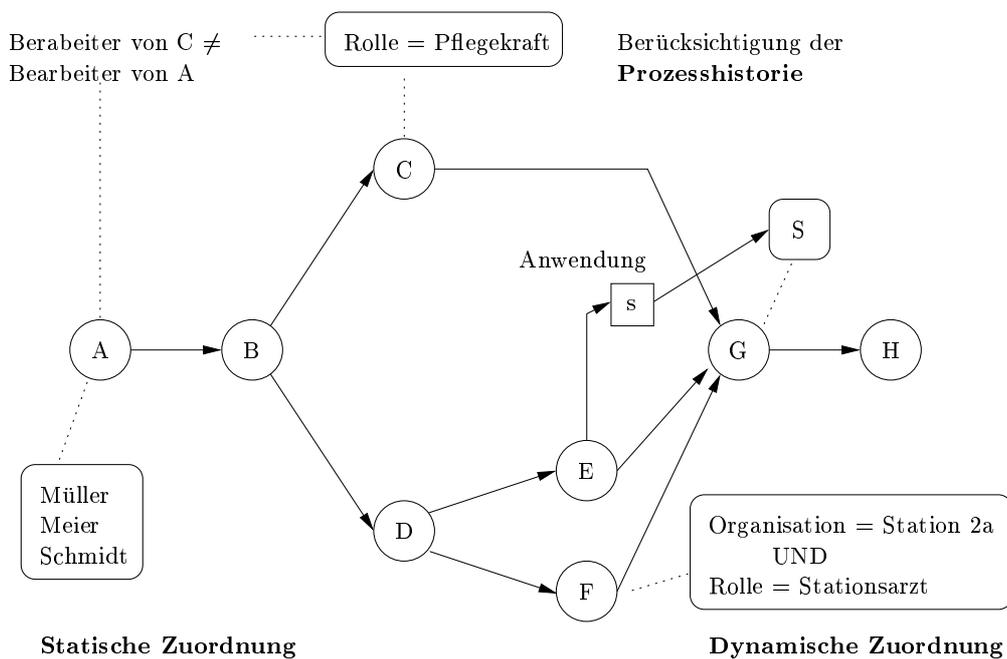


Abb. 2.5: Mitarbeiterzuordnungen in MQSeries Workflow

Neben der Festlegung von Mitarbeiterzuordnungen im Hinblick auf die Ausführung von Aktivitäten, gibt es in MQSeries Workflow noch die Möglichkeit, Rechte festzulegen, welche Bearbeiter welche Prozessmodelle instantiiieren können. Diese können ebenfalls über eine disjunktive (ODER) oder konjunktive (UND) Verknüpfung einer Rolle und Organisationseinheit spezifiziert werden.

2.2.1.2 Staffware

Staffware wurde von Staffware House entwickelt und ist, ebenso wie MQSeries Workflow, ein Production Workflow System. In der Organisationsmodellierung wird zwischen Person, Rolle und Gruppe unterschieden, wobei eine Rolle eine Funktion oder Fähigkeit darstellt, die durch genau eine Person oder eine Gruppe erfüllt wird. Eine Rolle entspricht daher eher einer Stelle und fasst nicht – wie bei PMS sonst üblich – eine Menge von Personen mit derselben Kompetenz zusammen. Um mehreren Personen die gleiche Rolle zuweisen zu können, müssen die betreffenden Personen in einer Gruppe zusammengefasst werden und dieser Gruppe wird anschließend eine Rolle zugeordnet. Eine Zuordnung mehrerer Rollen zu einer Gruppe

unabhängig von Personen ist jedoch nicht möglich, wodurch Gruppen unabhängig von personellen Umstrukturierungen wären. Eine Gruppe kann lediglich aus einer beliebigen Anzahl von Personen bestehen.

Zugriffsberechtigungen werden in Staffware so definiert, dass einer Aktivität Personen oder Rollen direkt zugewiesen werden. Es ist also nicht möglich, Berechtigungen zu definieren, die unabhängig von Personen sind. Durch die Definition von beliebig vielen Leitern innerhalb einer Gruppe sind Untergruppen oder Teams abbildbar. Ein Gruppenleiter hat neben den Funktionen, die er in der Gruppe wahrnehmen muss, die Möglichkeit, im Staffware-Clienten Zugriffsberechtigungen an eine oder mehrere Personen und Weiterleitungen an eine Gruppe oder Person zu erteilen. Zur Erläuterung hier zwei einfache Beispiele:

1. Erhält Person B eine Zugriffsberechtigung auf Person A, erscheinen in der Arbeitsliste von Person B seine eigenen Tätigkeiten und alle Tätigkeiten von Person A. Er kann nun auch die Tätigkeiten von Person A ausführen. Person A sieht aber dennoch all seine Tätigkeiten in seiner Arbeitsliste.
2. Auf Person A wird eine Weiterleitung eingerichtet. Erhält Person A nun neue Tätigkeiten, erscheinen diese neuen nicht mehr in seiner Arbeitsliste. Die neuen Tätigkeiten erscheinen lediglich in der Arbeitsliste der Person, an die weitergeleitet wurde.

Auf diese Weise wäre eine Art von Vertreterregelungen bestimmbar.

2.2.1.3 WorkParty

WorkParty [Rupi97, RSD97] ist ein kommerzielles, von Siemens Nixdorf entwickeltes PMS. Wie in [FrSc97] beschrieben, findet die Entwicklung eines Workflows, in Folge der Trennung von Ablauflogik (Kontroll- und Datenfluss) und Anwendungscode, in zwei Phasen statt. Die Ablauflogik, d. h. die Modellierung der Prozesse oder Abläufe, wird in Form von Ablaufdiagrammen dargestellt. Es sind einfache Sequenzen ebenso abbildbar, wie Alternativen, Parallelitäten und Schleifen. Sind parallele Wege modelliert, kann einer dieser Wege auch als verzichtbar definiert werden. Allerdings sind weder zeitliche Aspekte noch FOR-Schleifen modellierbar. Durch die Definition von Prozessvariablen und deren Verknüpfung mit Aktivitäten wird der Datenfluss implizit modelliert. Für die Aktivitäten werden potentielle Bearbeiter über entsprechende Ausdrücke festgelegt, wobei auf ein zugrundegelegtes Organisationsmodell Bezug genommen wird. Anschließend werden Programme entwickelt bzw. existierende Programm-Module verwendet, die den Anwender in der Ausführung der Aktivitäten unterstützen, wie z. B. Bildschirmmasken zur Ein- und Ausgabe von Anwendungsdaten. Nur in dieser Phase können Schritte definiert werden, die auch andere Funktionen integrieren, wie einen Terminplaner oder den Zugriff auf eine elektronische Akte. Zur Laufzeit der Prozessinstanzen besteht keine Möglichkeit, einzelne Prozessschritte mit anderen Anwendungen und Funktionen zu kombinieren.

WorkParty bietet die Möglichkeit, bei der Beschreibung der Aktivitäten Programme anzugeben, die in Ausnahmefällen ausgeführt werden. Diese Ausnahmesituationen betreffen lediglich (System-)Fehler und Abbruch oder Unterbrechung eines Prozesses. Es ist z. B. nicht möglich, eine bereits abgeschlossene Aktivität zurückzusetzen, wenn bereits eine nachfolgende Aktivität gestartet wurde. Ein Benutzer kann lediglich den aktuellen Prozessschritt nochmals bearbeiten. Bei Aufruf der verschiedenen Ausnahmeprogramme werden diese genauso behandelt wie die üblichen Aktivitätenprogramme. Ein Vorteil dieser Ausnahmeprogramme ist z. B., dass sie an unterschiedlichen Stellen wiederverwendet werden können.

WorkParty unterstützt, wie in Abb. 2.6 dargestellt, eine realitätsnahe Darstellung der Organisationsstruktur. Organisationseinheiten können hierarchisch angeordnet sein, werden von einer Stelle geleitet und haben bestimmte Zuständigkeiten. Diesen Kompetenzen werden sowohl Ressourcen als auch Aufgaben zugeteilt. Mitarbeiter besetzen eine oder mehrere Stellen, sind aber nur genau einer Stammstelle zugeteilt und sind als Stellvertreter von beliebig vielen Stellen definierbar. Zu jedem Mitarbeiter und jeder Stelle gehören bestimmte Befugnisse. Diese werden über Kompetenzen modelliert und verbinden so die Mitarbeiter bzw. nichtleitenden Stellen mit den Organisationseinheiten, wodurch die Flexibilität einer Organisation widerspiegelt wird. Durch die Definition von Kompetenzen und deren Zuordnung zu verschiedensten Entitäten des Organisations-Modells entsteht eine beliebige Verfeinerung der Organisation.

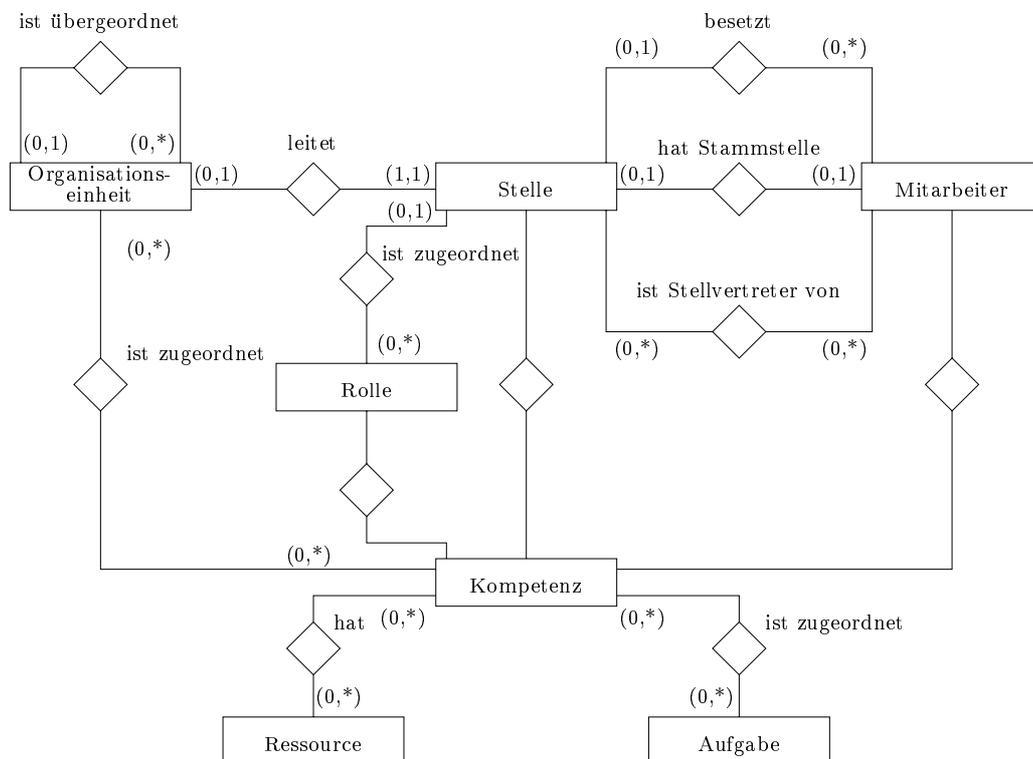


Abb. 2.6: Organisations-Metamodell von WorkParty (aus [FrSc97])

Bereits zur Modellierzeit werden für den Fall der Abwesenheit eines Mitarbeiters die Vertreterregelungen für Stellen über die Mitarbeiter festgelegt. Ein Nachteil dieser Vertreterregelung ist, dass die Personalstruktur zum Zeitpunkt der Modellerstellung genau bekannt sein muss. Herrscht in einem Bereich oder der ganzen Organisation eine hohe Wechselrate des Personals, wie es in einem Krankenhaus der Fall sein kann, ist dies sehr aufwendig. Bei jedem Wechsel müssen dann die Vertreterregelungen neu definiert werden. Sinnvoller wäre daher eine Definition der Vertreterregelungen nur anhand der Stellen, d.h. der Arzt XY wird durch den Arzt AB vertreten. Auch werden die Einträge der Arbeitsliste des abwesenden Mitarbeiters nicht automatisch in der Arbeitsliste des Vertreters angezeigt, sondern der Mitarbeiter muss sich am PMS explizit als Vertreter anmelden. Dies wiederum setzt voraus, dass ihn jemand über die wahrzunehmende Vertretung in Kenntnis setzen muss.

2.2.2 Ad-hoc Workflow

Ad-hoc Workflow Systeme dienen der Unterstützung von schwach strukturierten Prozessen, die in genau der gleichen Form meist nicht wiederholt werden. Dadurch existiert nur eine geringe oder keine Steuerung des Workflows durch das System. Die Spezifikation und Koordination der einzelnen Schritte legt der Anwender dynamisch fest. In der Regel ist ein Ad-hoc Workflow nur von kurzer Dauer und umfasst wenige Schritte. Einige wenige, i.d.R. dokumentenorientierte Systeme, gestatten es zudem, dass von dem vormodellierten Workflow abgewichen werden kann. Allerdings sind entsprechende Abweichungen in heutigen Systemen mit großen Einschränkungen verbunden. Insbesondere wird die Konsistenz der Prozesse nicht systemseitig geprüft (z.B. Aufruf von Schritten mit fehlenden Daten möglich). Als Beispiel eines Ad-hoc Workflow Systems wird im Folgenden ProMInanD [KaRa90, KaRa91] beschrieben.

2.2.2.1 ProMInanD

ProMInanD ist ein mappenbasiertes PMS, dessen Arbeitsweise an eine Handakte mit mehreren Dokumenten erinnert, die verschiedene Arbeitssituationen durchläuft. Dabei kann eine elektronische Handakte [KaRa90, KaRa91] immer nur von einer Person bearbeitet werden, niemals von mehreren Personen gleichzeitig. Erforderlichenfalls kann aber eine Kopie erstellt werden, auf der dann eine andere Person arbeitet. Allerdings sind beim Zusammenfügen der verschiedenen Kopien Konflikte nicht ausschließbar. Diese müssen dann vom Bearbeiter der Join-Aktivität manuell aufgelöst werden. Auch ist schwer feststellbar, wer zu einem bestimmten Zeitpunkt den Vorgang bearbeitet.

Das Organisationsmetamodell von ProMInanD besteht, wie in [ReDa96] beschrieben, aus Organisationseinheiten, Stellen, Mitarbeitern, Arbeitsgruppen und Funktionen. Ein Mitarbeiter hat in einer Organisationseinheit eine Stelle inne, allerdings müssen nicht alle Stellen besetzt sein. Eine Organisationseinheit kann sowohl einer einzelnen direkt untergeordnet als auch mehreren Organisationseinheiten übergeordnet sein. Jede Organisationseinheit setzt sich aus mehreren Stellen zusammen und wird von genau einer Stelle, also nicht von einer Gruppe von Leitern, geleitet. Jede Stelle hat innerhalb einer Organisationseinheit verschiedene Aufgaben zu erfüllen. Um z.B. Projekte innerhalb einer Organisationseinheit modellieren zu können, können Mitarbeiter in Arbeitsgruppen zusammengefasst werden. Wie bei den Organisationseinheiten, gibt es auch hier wieder die Möglichkeiten, hierarchische Strukturen zu modellieren. Allerdings sind die Arbeitsgruppen kurzfristig änderbar und damit dynamischer in ihrer Zusammensetzung. Aufgaben und Verantwortlichkeiten, die ein Mitarbeiter zu erfüllen hat, werden durch Funktionen beschrieben, die entweder über die Stellen definiert werden oder aber über die Mitarbeiter selbst. Da ein Mitarbeiter mehrere Stellen in einem Unternehmen innehaben kann, muss er bei der Anmeldung an das PMS angeben, welche Stelle er aktuell einnimmt. Im Gegensatz zu den bereits vorgestellten Production Workflow Systemen sind bei ProMInanD den Stellen Funktionen zugeordnet, d.h. mehrere Stellen in unterschiedlichen Organisationseinheiten können die gleiche Funktion ausüben. Diese Zuordnung ist also unabhängig von der Personalstruktur.

Bei der Modellierung einer Workflow-Aktivität kann festgelegt werden, welche Rolle, also Stelle und/oder Funktion, der bearbeitende Mitarbeiter beim Start der Aktivität zu besetzen hat. Der Benutzer sieht nur die Aktivitäten, die er in der aktuell innehabenden Rolle auch bearbeiten darf. Innerhalb einer Aktivität kann spezifiziert werden, wie der Vorgang durch wen abzuarbeiten ist. Allerdings ist es auch möglich, das System während der Ausführung des Workflows entscheiden zu lassen, welcher Bearbeiter diese Aktivität nun endgültig zu bearbeiten hat. Dadurch ist auch modellierbar, dass ein Vorgang nur durch eine bestimmte

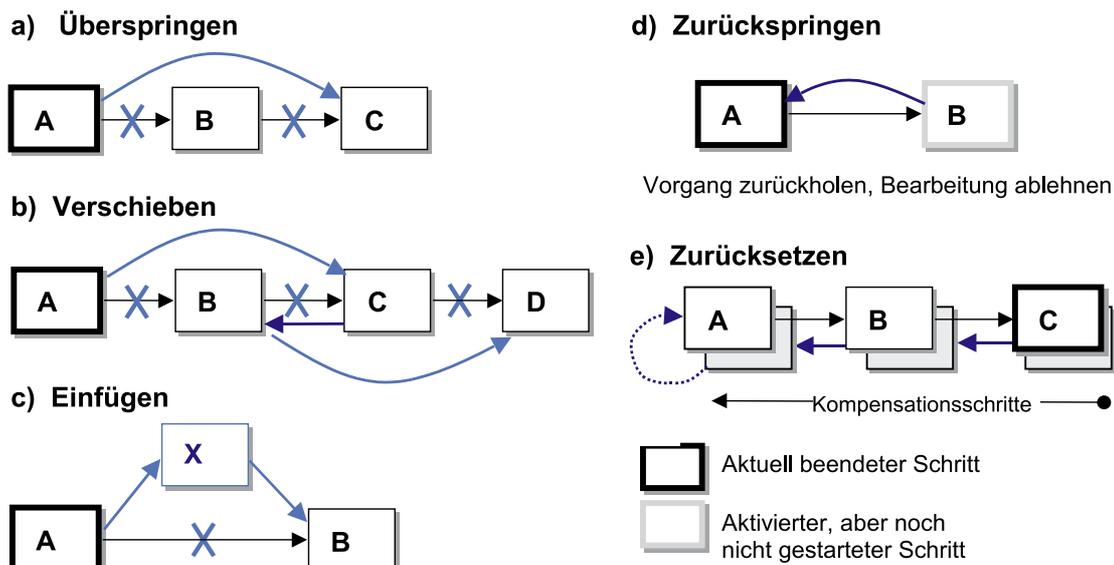


Abb. 2.7: Ad-hoc Änderungen in ProMInanD

Person bzgl. einer bereits erledigten Aktivität zu bearbeiten ist, d. h. abhängige Bearbeiterzuordnungen sind in ProMInanD modellierbar.

Wie eingangs beschrieben, ist der Workflow in ProMInanD nicht starr strukturiert und kann zur Laufzeit geändert werden. So ist es möglich, Aktivitäten zu überspringen, deren Reihenfolge bei der Abarbeitung zu ändern, neue Aktivitäten unmittelbar hinter einer soeben bearbeiteten Aktivität einzufügen, zu einer vorhergehenden Aktivität zurückzuspringen oder eine oder mehrere Aktivitäten zurückzusetzen. Schlägt eine Aktivität fehl, wird der Workflow vor den Start der letzten manuellen Aktivität zurückgesetzt, wobei dazwischen liegende Aktivitäten kompensiert werden [Karb94]. Aufgrund des ausdruckschwachen Prozessmetamodells von ProMInanD, stellt sich ein Workflow aus der Sicht des einzelnen Bearbeiters immer als eine Sequenz von Aktivitäten dar. Workflow-Änderungen entsprechen deshalb einfachen Änderungen der Anordnungsbeziehungen von Schritten bzw. der Hinzunahme neuer Schritte zu einer Sequenz, komplexe Änderungen (z. B. Einfügen einer einzelnen Aktivität parallel zu einer Menge existierender Schritte) sind dagegen nicht realisierbar. Alle spezifizierten Änderungsfunktionen sind immer von der Person ausführbar, die die aktuelle Aktivität bearbeitet. Aus diesem Grund sind keine speziellen Änderungs- oder Anpassungsrechte zu definieren. Zur weiteren Erläuterung siehe Abb. 2.7.

Ein Vorteil von ProMInanD ist sicherlich der Umgang mit Ad-hoc Änderungen, sofern sie vom aktuellen Bearbeiter eines Workflows durchgeführt werden sollen, und die Änderungen nicht von allzu großer Komplexität sind. Änderungen können, von wenigen Ausnahmen abgesehen, nur vom aktuellen Bearbeiter vorgenommen werden, so dass andere Prozessbeteiligte, die nicht aktuell an der Ausführung des Workflows beteiligt sind, auch nicht in dessen Kontrolle eingreifen können. Ein weiterer Nachteil ist, dass Aktivitäten eines Workflows, wenn sie parallel ausgeführt werden und beim Zusammenfügen Konflikte entstehen, diese nicht automatisch vom System erkannt werden, sondern manuell beseitigt werden müssen. Durch die starke Orientierung am Dokument bzw. Dokumentenfluss ist eine Adaption des Datenflusses, wie es im Kontext dynamischer Workflow-Änderungen erforderlich werden kann, sehr schwierig durchführbar.

2.2.3 Vergleich der beschriebenen PMS

Tabelle 2.2 gibt eine vergleichende Übersicht zu den verschiedenen Systeme und ihren Fähigkeiten.

	MQSeries Workflow	Staffware	WorkParty	ProMInanD
Elemente d. Org.modellierung				
- Organisationseinheit	x		x	x
- Rolle	x		x	
- Mitarbeiter	x	x	x	x
- Gruppe		x		
- Stelle		x	x	x
- Kompetenz			x	
- Arbeitsgruppe				x
Unterstützung von				
- hierarchischen Beziehungen zwischen OE	ja	nein	ja	ja
- Vertreterbeziehungen	ja	ja	ja	—
- Kompetenzfähigkeiten	eingeschränkt ¹	nein	ja	nein
- Stellenkonzept	nein	eingeschränkt	ja	ja
- spezialisierte Beziehungen zwischen Rollen	nein	nein	nein	—
Vertreterregelungen				
- auf Mitarbeiter-Ebene	ja	ja	nein	ja
- auf Stellenebene	nein	nein	ja (+ MA) ²	ja
- aufgabenbezogene Differen- zierung	nein	nein	nein	nein
Bearbeiterzuordnungen				
- abhängige BZ	ja	ja	ja	ja
- Mitarbeiter-bezogen ³	ja	ja	ja	ja
- Vier-Augen-Prinzip ⁴	ja	ja	ja	ja
- Stellen-bezogen ⁵	nein	nein	ja	nein
- OE-bezogen ⁶	nein	nein	nein	nein
- unabhängige BZ	ja	ja	ja	ja
- dynamische BZ	ja	ja	nein	ja
Rechte für dynamische WF-Anpassungen ⁷	nein	nein	nein	implizit
Ausdrucksmächtigkeit	gering	gering	hoch	mittel

x: vorhanden —: nicht bekannt

Tabelle 2.2: Vergleich der vorgestellten Prozess-Management-Systeme

¹über sogenannte Levels

²Eine Stelle kann als Vertretung einem Mitarbeiter zugeordnet werden.

³z. B. gleicher Bearbeiter wie bei vorausgehender Aktivität

⁴Bearbeiter ungleich Bearbeiter einer vorhergehenden Aktivität

⁵gleiche Stelle, die vorausgehende Aktivität bearbeitet hat

⁶Bearbeiter aus der gleichen OE wie Bearbeiter einer vorhergehenden Aktivität

⁷Beispielsweise könnte in einem Krankenhaus bei der Verlegung eines Patienten erforderlich werden, dass rücklaufende Befunde in Zukunft den Ärzten der neuen Station via Arbeitsliste angeboten werden.

2.3 Neue Ansätze für die Rechteverwaltung in WfMS

Zur Realisierung der Zugriffskontrolle werden in heutigen WfMS Beschränkungen oder Rechte für Benutzer und Rollen definiert. Das Modell der Beschränkungen wird z. B. durch *Separation of Duties* (kurz: SoD) beschrieben. Das Konzept der SoD besagt, dass eine Ausführungsfunktion und die zugehörige Kontrollfunktion nicht von ein und derselben Person besetzt sein soll, um einen möglichen Missbrauch durch diese Person ausschließen zu können. Im Kontext der Medikamentenbestellung im Krankenhaus etwas darf die Person, die Medikamente bestellt, nicht auch dieselbe Person sein, die die Bestellung in der Verwaltung freigibt. So wird ausgeschlossen, dass beispielsweise auf dem Schwarzmarkt teuer verkaufbare Medikamente vom Krankenhaus bestellt werden, ein Mitarbeiter diese illegal verkauft und den Erlös in die eigene Tasche wirtschaftet, ohne dass es irgend jemandem auffällt.

Herkömmliche rollenbasierten Zugriffskontrollen können das Konzept der SoD jedoch nicht adäquat umsetzen [BFA99], da sie zwar rollenbasierte Zugriffe unterstützen, nicht jedoch das Konzept der exklusiven Rechte. Exklusiv heißt, dass eine Rolle/Berechtigung eine andere Rolle/Berechtigung vom Zugriff ausschließt. Dadurch ist das Risiko des Missbrauchs der gewährten Berechtigungen sehr hoch. Aus dieser Misere heraus entwickelten Bertino et al. [BFA99] ein Zugriffsmodell, das die Vorteile RBAC und SoD miteinander kombiniert. Durch die Kombination von RBAC mit SoD ist kein Benutzer in der Lage, alle Aktivitäten, die zu einem Prozess gehören, allein zu bearbeiten, da jetzt das Konzept der exklusiven Rechte berücksichtigt wird.

Ein Beispiel zur Kombination von RBAC und SoD ist der Ablauf einer Steuerrückerstattung, auf den im weiteren Verlauf dieses Abschnitts Bezug genommen wird. Dieser ist zusätzlich in Abb. 2.8 skizziert.

- Schritt 1: Ein Rückerstattungsangestellter bereitet einen Scheck zwecks Steuerrückerstattung vor.
- Schritt 2: Ein Rückerstattungsmanager weist den Scheck zurück oder gibt ihn vorläufig frei. Dieser Schritt soll von zwei unabhängigen Rückerstattungsmanagern durchgeführt werden (Schritt 2a und 2b). Der Scheck wird ausgestellt, wenn er von beiden Managern vorläufig freigegeben wird, ansonsten wird er für ungültig erklärt.
- Schritt 3: Die Entscheidungen der beiden Rückerstattungsmanager werden gesammelt und der endgültige Entschluss gefällt. Der Manager, der die Entscheidungen sammelt, muss von den Ausführenden von Schritt 2, verschieden sein.
- Schritt 4: Ein Rückerstattungsangestellter stellt den Scheck, auf Basis der Entscheidung aus Schritt 3, aus oder verwirft ihn. Der betreffende Rückerstattungsangestellte muss von dem Rückerstattungsangestellten, der den Scheck vorbereitet hat, verschieden sein.

Der hier dargestellte Ablauf der Steuerrückerstattung enthält sowohl dynamische wie auch statische SoD. Jedem Schritt ist statisch eine der beiden Rollen (Angestellter oder Manager) zugeteilt. Die Bearbeitung verschiedener Schritte wird dynamisch verschiedenen Personen mit der gleichen Rolle zugewiesen, damit beispielsweise nicht dieselbe Person den Scheck freigeben kann, die ihn auch vorbereitet hat.

In der von Bertino et al. [BFA99] für dieses Beispiel verwendeten Personalstruktur gibt es eine globale Teilordnung. Interessant ist, dass durch die Angabe der Personalstruktur die zugehörigen Vertreterregelungen bereits über die hierarchische Zuordnungsbeziehungen der

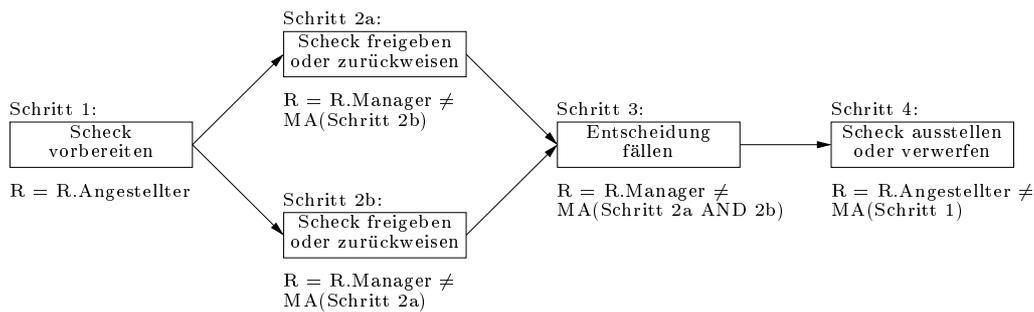


Abb. 2.8: Prozess der Steuerrückzahlung (aus [BFA99])

Rollen modelliert sind. Bezogen auf das Beispiel der Steuerrückerstattung etwa gilt Folgendes: Normalerweise bereitet der *Rückerstattungsangestellte* den Scheck vor. Existiert kein Benutzer, der diese Rolle einnimmt, kann dieser Schritt von jedem Benutzer übernommen werden, der die Rolle *Rückerstattungsmanager* oder *Generalbevollmächtigter* besetzt, da beiden die Stelle *Rückerstattungsangestellter* hierarchisch untergeordnet ist. Abb. 2.9 skizziert diese globale Teilordnung und drückt indirekt auch die Vertreterregelungen aus.

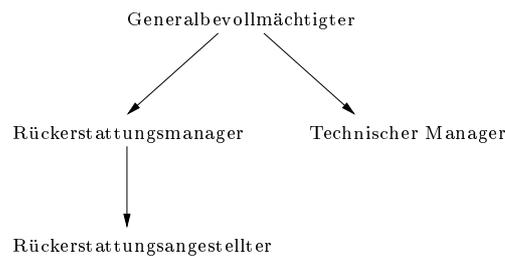


Abb. 2.9: Hierarchische Personalstruktur zum Beispiel der Steuerrückerstattung aus Abb. 2.8 (aus [BFA99])

Bertino et al. [BFA99] haben in ihrer Arbeit eine Kombination von verschiedenen bereits einzeln existierenden Zugriffskontrollmethoden beschrieben. Sie versuchen die Vorteile der verschiedenen Methoden miteinander zu koppeln und so deren Schwächen zu beseitigen. Der Vorteil dieser Koppelung liegt in der gewünschten Einschränkung der Rechte einzelner Personen, so dass der Missbrauch besetzter Positionen in einer Organisation weitgehend ausgeschlossen werden kann. Im vorgestellten Beispiel der Steuerrückerstattung etwa, müssten sich mehrere Personen zusammenschließen, um einen Betrug durchführen zu können. Allerdings ist der Aufwand der Umsetzung dieser Zugriffskontrolle sehr aufwendig, da sowohl die Rollen modelliert werden müssen, als auch die Beschränkungen, um die SoD umzusetzen. Wurden aber alle diese Festlegungen getroffen, ist der Ansatz von Bertino et al. ein verständliches und sicher auch gut „wirkendes“ Kontrollsystem.

Kapitel 3

Szenarien

In diesem Kapitel wird ein Anwendungsszenario vorgestellt, das die Notwendigkeit von (dynamischen) Prozessänderungen und zugehörigen Autorisierungskonzepten evident macht. Das gewählte Szenario bezieht sich auf reale Prozesse aus dem Klinikbereich, die von der Abteilung Datenbanken und Informationssysteme der Universität Ulm im Rahmen mehrerer Diplomarbeiten erhoben wurden [Meye96, Kony96a, Schu96a]. Vergleichbare Anforderungen bestehen aber auch in anderen Anwendungsdomänen, etwa im Zusammenhang mit der Abwicklung von Bestellvorgängen in einem Unternehmen [Ott96].

3.1 Das Grundszenario

Als Beispiel wird der (etwas vereinfachte) Ablauf einer radiologischen Untersuchung in einem Krankenhaus betrachtet.

Damit bei einem Patienten eine Diagnose bestätigt oder ausgeschlossen werden kann, ordnet der behandelnde Arzt eine radiologische Untersuchung an (z. B. Röntgen, CT) (Aktivität „Untersuchung anordnen“). Dazu muss mit der radiologischen Abteilung des Krankenhauses ein Termin vereinbart werden (Aktivität „Termin vereinbaren“). Bevor die Untersuchung durchgeführt wird, muss der Patient vom Arzt über eventuelle Risiken aufgeklärt werden (Aktivität „Patient aufklären“). Parallel dazu müssen auf der Krankenstation Vorbereitungen von einer Krankenschwester durchgeführt werden (Aktivität „Patient vorbereiten“). Am Tag der Untersuchung wird der Patient von der radiologischen Abteilung abgerufen (Aktivität „Patient abrufen“). Ist er nicht gehfähig, wird ein Krankentransport angefordert (Aktivität „Transport anfordern“), ansonsten begibt er sich zu Fuß zur radiologischen Abteilung. Dort wird die Untersuchung durchgeführt (Aktivität „Untersuchung durchführen“). Anschließend geht der Patient, sofern er gehfähig ist, zu seiner Krankenstation zurück. Ist er nicht gehfähig, wird ein Rücktransport angefordert (Aktivität „Rücktransport anfordern“). Der Radiologearzt erstellt einen Befund (Aktivität „Befund erstellen“), der an den Stationsarzt weitergeleitet und von diesem gelesen wird (Aktivität „Befund lesen“). Dieser Ablauf ist in Abb. 3.1 dargestellt.

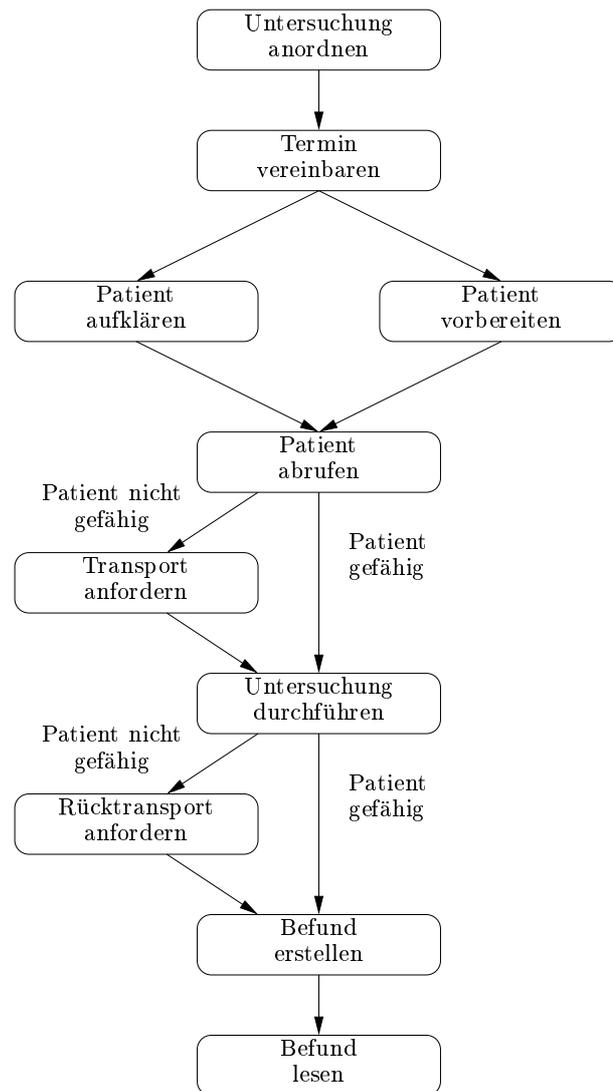


Abb. 3.1: Ablauf einer radiologischen Untersuchung

Aktivität	Typ	Bearbeiter
Untersuchung anordnen	Aktivität	Stationsarzt
Termin vereinbaren	Aktivität	Schwester oder Stationsarzt
Patient aufklären	Aktivität	Stationsarzt
Patient vorbereiten	Aktivität	Schwester
Patient abrufen	Aktivität	MTA der radiolog. Abteilung
Transport anfordern	Aktivität	Schwester
Untersuchung durchführen	Subworkflow	Radiologearzt
Rücktransport anfordern	Aktivität	Schwester
Befund erstellen	Subworkflow	Radiologearzt
Befund lesen	Aktivität	Stationsarzt

Tabelle 3.1: Aktivitäten mit zugehörigen Bearbeitern für den Prozess in Abb. 3.1

Die Aktivitäten und ihre zugehörigen Bearbeiter sind in Tabelle 3.1 für jede Aktivität einzeln aufgeführt.

3.2 Mögliche Änderungen im Workflow

Während ein konkreter Workflow, d. h. eine Instanz des beschriebenen Prozessmodells, ausgeführt wird, können situative Änderungen im Ablauf notwendig werden [DRK00]. Auf solche Ausnahmesituationen muss auf unterschiedliche Art und Weise reagiert werden können. Beispielsweise muss es möglich sein,

- einen Prozess abzubrechen
- einen Prozess zu unterbrechen
- Termine abzusagen oder zu ändern
- Aktivitäten zu überspringen
- einzelne Schritte zu verschieben
- neue Aktivitäten hinzuzunehmen.

Im Folgenden sollen am Beispiel des vorgestellten Szenarios einige mögliche Ausnahmen beschrieben werden.

Beispiel 1: Überspringen eines aktivierten Schrittes

Zustand:	Der Patient ist aufgeklärt und vorbereitet worden.
Ereignis:	Der Patient sucht die Untersuchungsstelle ohne Abruf auf, meldet sich also direkt bei der zuständigen Stelle.
Aktion:	Der (elektronische) Abruf des Patienten auf der Station wird damit hinfällig. Die Assistentin auf der Seite der Untersuchungsstelle überspringt deshalb den entsprechenden Schritt in ihrer Arbeitsliste.
Berechtigung:	Assistentin der betreffenden Untersuchungsstelle – sie ist auch die reguläre Bearbeiterin des auszulassenden Schrittes – oder deren fachlicher Vorgesetzter.

Beispiel 2: Überspringen einer Menge von Schritten

Zustand:	Die Untersuchung ist angeordnet.
Ereignis:	Der Zustand des Patienten verschlechtert sich plötzlich und die Untersuchung soll sofort durchgeführt werden.
Aktion:	Überspringen der dazwischen liegenden Schritte.
Berechtigung:	Arzt der behandelnden Einheit oder Stationsarzt (Im Prinzip muss jedem Arzt die Möglichkeit eingeräumt werden, diese Aktion durchzuführen, selbst wenn er laut System nicht der behandelnden Einheit angehört. In diesem Fall ist eine nachträgliche Kontrolle notwendig, um festzustellen ob der Arzt dazu eine Veranlassung hatte.)

Beispiel 3: Auslassen von Schritten und manuelle Festlegung von Anwenderdaten

Zustand:	Für einen Patienten wurde eine Untersuchung angeordnet. Die (elektronische) Terminvereinbarung ist im Gange.
Ereignis:	Der behandelnde Arzt möchte schnell einen Termin und ruft direkt bei der Untersuchungsstelle an. Nachdem ihm die Ambulanzkraft den Termin mitgeteilt hat, bricht er die elektronische Terminvereinbarung ab.
Aktion:	„Manuelles“ Eintragen des telefonisch erhaltenen Termins in den Kalender
Berechtigung:	Mitarbeiter der behandelnden Einheit

Beispiel 4¹: Abbruch eines Prozesses

Zustand:	Für die Untersuchung eines Patienten ist ein Termin vereinbart worden. Der Patient wird über die Untersuchung aufgeklärt.
Ereignis:	Der Patient willigt nicht in die Durchführung der Untersuchung ein.
Aktion:	Der laufende Prozess muss unter Angabe eines Grundes abgebrochen und der Termin storniert werden.
Berechtigung:	behandelnder Arzt

Beispiel 5¹: Abbruch eines Prozesses

Zustand:	Der Patient befindet sich in der Untersuchungsstelle, die Untersuchung soll starten.
Ereignis:	Der Patient zieht kurzfristig die Einwilligung zur Untersuchung zurück.
Aktion:	Der Prozess muss unter Angabe eines entsprechenden Grundes abgebrochen werden.
Berechtigung:	untersuchender Arzt

¹Bei etwas komplexeren Prozessen, die mehrere dutzende bis hunderte Arbeitsschritte umfassen, sind entsprechende Fälle prinzipiell nicht mehr vormodellierbar. Durch die Anzahl der zu modellierenden Abweichungen wird das Prozessmodell unübersichtlich.

Beispiel 6: Partielles Zurücksetzen eines Prozesses

Zustand:	Der Patient wurde in die Untersuchungsstelle gebracht und die Untersuchung soll starten.
Ereignis:	Der Patient ist nicht nüchtern oder sein Gesundheitszustand hat sich so verschlechtert, dass die Untersuchung nicht durchzuführen ist.
Aktion:	Der aktuelle Schritt („Untersuchung durchführen“) muss abgebrochen und ein neuer Termin vereinbart werden.
Berechtigung:	untersuchender Arzt oder Mitarbeiter der untersuchenden Einheit

Beispiel 7: Änderung von Ausführungsrechten zur Laufzeit

Zustand:	Der Patient wurde nach der Untersuchung auf seine Station zurücktransportiert. Der Befund wurde erstellt.
Ereignis:	Während der Befunderstellung wird der Patient auf eine andere Station verlegt. Üblicherweise wird die Aktivität „Befund lesen“ von einem Arzt (derselbe oder ein anderer Arzt) durchgeführt, auf der die betreffende Untersuchung angeordnet wurde. Angenommen der Patient befand sich zum Zeitpunkt der Untersuchungsanordnung auf der Notaufnahmestation und wurde nach der Durchführung der Untersuchung auf eine normale Station verlegt, dann muss der erstellte Befund von einem Arzt dieser neuen Station gelesen werden. Das bedeutet, dass die Ausführungsberechtigung für die in Zukunft liegende Aktivität „Befund lesen“ sich bei der Durchführung des Prozesses ändert.
Aktion:	Die Ausführungsberechtigung für die Aktivität „Befund lesen“ muss geändert werden.
Berechtigung:	behandelnder Arzt

Beispiel 8: Hinweis auf (drohende) Terminverletzung

Zustand:	Der Patient wurde (in der Vergangenheit) untersucht.
Ereignis:	Auf der Station ist noch kein Befund eingegangen.
Aktion:	Mitteilung an die prozessverantwortliche Person über den noch fehlenden Befund (hier: prozessverantwortlicher Arzt der aktuell behandelnden Station).
Benachrichtigung:	Ärzte der aktuell behandelnden Station und Arzt, der die Untersuchung durchgeführt hat.

Beispiel 9: Änderung von Terminen

Zustand:	Für eine Untersuchung wurde ein Termin vereinbart.
Ereignis:	Der Termin kollidiert mit anderen Terminen des Patienten.
Aktion:	Der Termin wird zurückgesetzt und ein neuer vereinbart.
Berechtigung:	behandelnder Arzt

Beispiel 10: Änderung der Prozessbeteiligten

Zustand:	Die Untersuchung wurde durchgeführt, der Befund aber noch nicht geschrieben.
Ereignis:	Der Patient verstirbt.
Aktion:	Der Prozess darf nicht mehr abgebrochen werden, d. h. der für die Untersuchung zuständige Arzt muss den Befund noch erstellen.
Berechtigung:	Arzt, der die Untersuchung durchgeführt hat

In dieser Arbeit erfolgt eine Beschränkung auf ein möglichst einfaches, überschaubares Anwendungsszenario. In der klinischen Praxis finden sich noch weitaus komplexere Ablaufbeispiele mit vergleichbaren Anforderungen, bei denen Prozesse langandauernd (bis zu mehreren Monaten) sein können und dabei dutzende bis hunderte Schritte umfassen [FSM96, Kony96b, KSR96a, KSR96b, Schu96b].

Die vorgestellten Abweichungsszenarien zeigen, dass medizinische Prozesse eine hohe Variabilität und Dynamik besitzen. Abweichungen vom Standardablauf kommen häufig vor und die Prozessteilnehmer sind auch in der Lage, entsprechende Abweichungen und die damit zusammenhängenden Aktionen durchzuführen. Die skizzierten Szenarien heben hervor, dass es ein Prozess-Management-System berechtigten Benutzern erlauben muss, in flexibler Form Abweichungen vom vormodellierten Workflow vorzunehmen. Geeignete Autorisierungskonzepte sind in diesem Zusammenhang unerlässlich.

Kapitel 4

Das ADEPT-Prozess-Management-System

4.1 Grundidee von ADEPT

Das ADEPT-Prozess-Management-System wird seit 1995 in der Abteilung Datenbanken und Informationssysteme der Universität Ulm entwickelt. Aus der Zusammenarbeit mit dem Universitätsklinikum Ulm sowie anderen Partnern sind in die Entwicklung der ADEPT-Technologie zahlreiche praktische Erfahrungen mit eingeflossen. Wie in [Weil97] und [Zeit99] beschrieben, war es das Ziel der Entwicklung von ADEPT, eine ausdrucksmächtige Beschreibungsmöglichkeit für betriebliche Abläufe zu schaffen, mit der eine graphisch anschauliche Darstellung von Prozessmodellen möglich ist.

In ADEPT sind Kontroll- und Datenfluss strikt voneinander getrennt. Weitere zur Modellierung notwendige Konzepte, wie die adäquate Abbildung von Organisationsstrukturen oder die Beschreibung temporaler Abhängigkeiten, werden ebenfalls unterstützt. Durch den Austausch von Informationen über eine Datenbank wird die Integration in bereits bestehende Systeme ermöglicht. Außerdem wird das Zusammenspiel mit komponentenbasierten Software-Technologien, ebenso wie die Durchführung dynamischer Ablaufänderungen unterstützt.

Auf die Möglichkeit zur automatischen Überprüfung von Korrektheitskriterien (zur Modellierungs- und Laufzeit) wurde besonderen Wert gelegt. Daher können mit ADEPT keine beliebigen Graphen erzeugt werden, statt dessen müssen die angebotenen Basiskonstrukte verwendet werden, die bereits zum Zeitpunkt der Modellierung auf Fehler oder Unstimmigkeiten hinweisen und so letztlich für einen reibungslosen Ablauf der Instanzen sorgen.

4.2 Prozess-und Organisations-Metamodell

4.2.1 Prozess-Metamodell

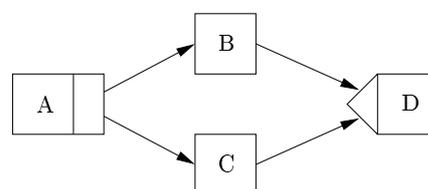
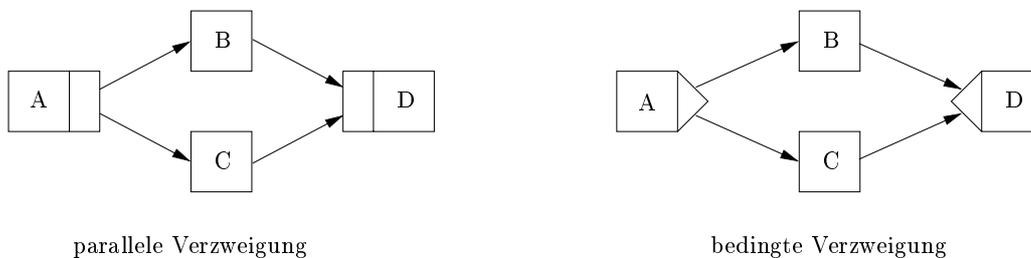
Da mit ADEPT alle in der Praxis auftretenden Abläufe modellierbar sein sollen [Weil97, Zeit99], werden bekannte Kontrollkonstrukte aus imperativen Programmiersprachen übernommen, da Arbeitsabläufe meistens in mehrere in sich abgeschlossene Teilaufgaben zerlegt werden. Die in ADEPT verwendeten Kontrollkonstrukte bauen auf einer regelmäßigen Blockstrukturierung auf und sorgen so für Übersichtlichkeit und einfache Änderbarkeit. Die Kontrollkonstrukte wie Sequenzen, Verzweigungen (unterschiedlicher Se-

mantik) oder Schleifen werden als regelmäßige Blöcke mit eindeutigen Start- und Endknoten dargestellt. Da sich diese Blöcke nicht überlappen, aber beliebig ineinander verschachtelt werden dürfen, sind nur Kontrollflussstrukturen mit regelmäßiger Schachtelung möglich [Reic00b]. Auch Prozesse werden als Block definiert, dieser besteht aus einem Start- und Endknoten und den dazwischen liegenden Aktivitätenknoten. Ein Workflow besteht somit aus einer Anzahl nacheinander abzuarbeitender Schritte oder Blöcke, auch Aktivitäten genannt, die durch Kanten miteinander verbunden sein müssen. Ein Beispiel eines einfachen Prozesses ist in Abb. 4.1 skizziert.



Abb. 4.1: Sequenz in ADEPT

Neben der Abbildungsmöglichkeit von einfachen Sequenzen können auch Schleifen und Verzweigungen unterschiedlicher Semantik modelliert werden. Sie haben jeweils einen eindeutigen Start- und Endknoten. Bei Schleifen ist im Endknoten eine prädikative Bedingung hinterlegt, nach dessen Überprüfung entschieden wird, ob die Schleife verlassen oder ein weiteres Mal durchlaufen wird. Bei Verzweigungen wird zwischen parallel auszuführenden Aktivitäten und Entscheidungen zwischen Alternativen unterschieden. Sie bestehen immer aus einem Verzweigungs-/Splitknoten, mindestens zwei Zweigen und einem Synchronisations-/Joinknoten. Die Entscheidungen zwischen Alternativen kann zwei verschiedene Formen annehmen. Entweder wird vor der Ausführung ein Zweig ausgewählt und ausgeführt, die anderen Zweige werden dann nicht weiter berücksichtigt oder es können alle Zweige ausgeführt werden, wobei frühestens nach Beendigung eines Zweiges der Kontrollfluss fortschreiten kann. Zur graphischen Veranschaulichung siehe Abb. 4.2.



parallele Verzweigung mit finaler Auswahl

Abb. 4.2: Verzweigungen in ADEPT

Bisher wurde lediglich die Kontrollflussmodellierung betrachtet. Aber es muss auch der Datenfluss und seine Modellierung zwischen den Aktivitäten berücksichtigt werden. Der Datenfluss zwischen den Aktivitäten wird in ADEPT über globale Prozessvariablen

[Reic00b], auch Datenelemente genannt, modelliert. Jedes dieser Datenelemente verwaltet ein Datenobjekt, wobei dieses Datenelement sowohl eine einfachere als auch komplexere Struktur besitzen kann. Die Ein- und Ausgabedaten der verschiedenen Aktivitäten werden über Datenflusskanten mit den entsprechenden Datenelementen verbunden. Ausgabedaten einer beendeten Aktivität werden in das über die entsprechende Schreibkante referenzierte Datenelement geschrieben. Danach kann der Wert von nachfolgenden Aktivitäten als Eingabeparameter gelesen werden. Eine korrekte Datenversorgung ist nur dann gewährleistet, wenn vor einem Lesezugriff auf ein Datenelement zuerst ein Schreibzugriff auf genau dieses Datenelement stattgefunden hat. Die Eingaben einer Prozessinstanz werden als vom Startknoten geschriebene Parameter der entsprechenden Datenelemente modelliert, Ausgabewerte erscheinen im Modell als Lesezugriffe des Endknotens.

Grundsätzlich gilt, dass nur solche Datenelemente für eine lesende Aktivität sichtbar sind, deren Schreiber im Kontrollfluss vor dem Leser liegen. Zur näheren Erläuterung ist mit Abb. 4.3 ein Beispiel zur Modellierung von Datenflüssen gegeben. Greift Aktivität F auf das Datenelement d_1 zu, wird F als Eingabewert immer die Eingabeparameter der betreffenden Prozessinstanz erhalten. Dabei ist egal, ob Aktivität C bereits vor dem Start von F schreibend auf d_1 zugegriffen hat oder nicht, da die Daten, die in einem parallelen Zweig geschrieben wurden, für andere Zweige nicht sichtbar sind. Daten aus einem parallelen Zweig werden nur sichtbar, wenn die parallelen Zweige über Synchronisationskanten verbunden sind und so die Aktivität aus dem einen Zweig zum Vorgänger einer Aktivität aus dem anderen Zweig wird. Dies ist im vorliegenden Beispiel etwa der Fall, damit Aktivität G auf das Datenelement d_2 zugreifen kann.

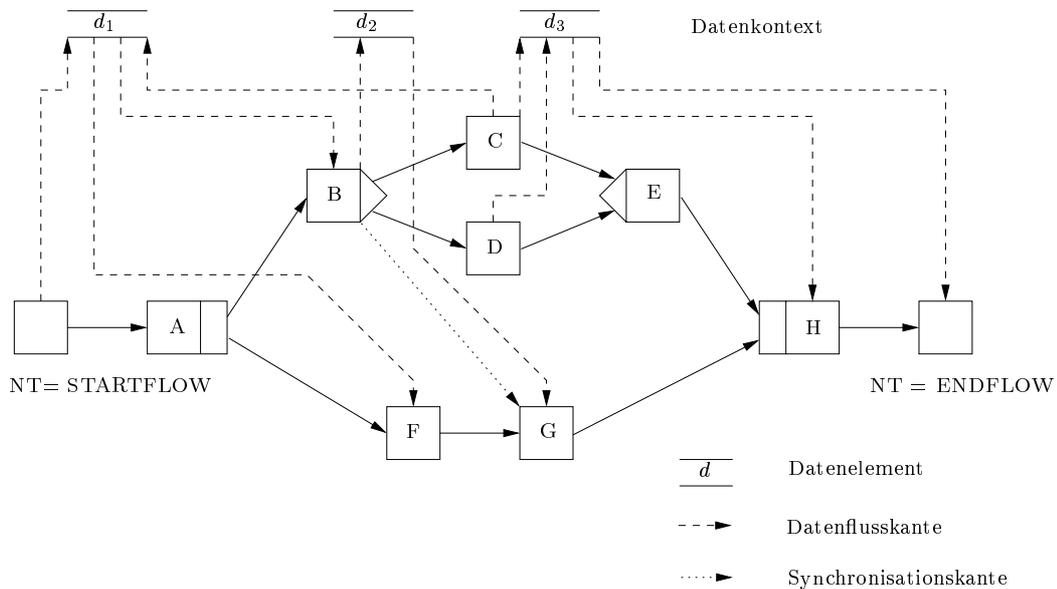


Abb. 4.3: Datenflussmodellierung in ADEPT

4.2.2 Organisations-Metamodell

Das Organisations-Metamodell beschreibt organisatorische Entitäten (z. B. „Organisationsgruppe“, „Organisationseinheit“, „Arbeitsgruppe“, „Rolle“, „Stelle“, „Fähigkeit“, „Mitarbeiter“, „Vertreterregelung“, „Aufgabenkategorie“) und deren Beziehungen zueinander. Die auf seiner Grundlage erzeugbaren Organisationsmodelle werden unter anderem dazu benötigt,

um Ausführungs- und Zugriffsrechte für Prozessvorlagen und -instanzen zu definieren. Der Aufbau des Organisations-Metamodells ist aus dem ER-Diagramm in Abb. 4.4 ersichtlich.

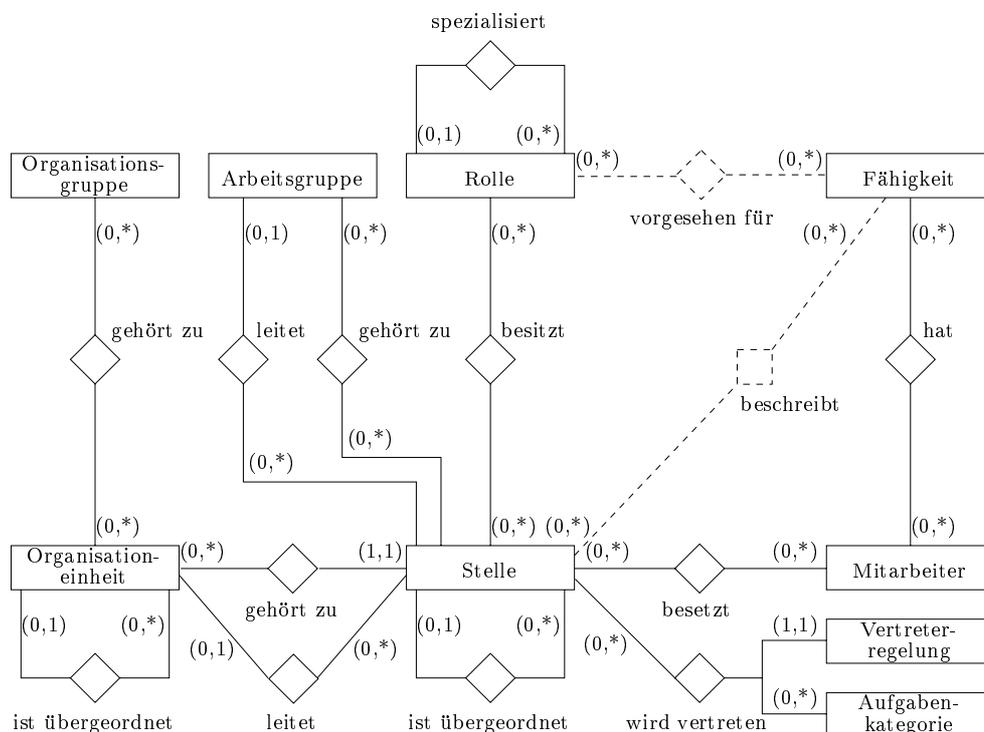


Abb. 4.4: Organisations-Metamodell von ADEPT

Zwischen Mitarbeitern und Stellen besteht normalerweise eine 1:1 Beziehung. Um Halbtagestellen und Mitarbeiter, die evtl. auch abwechselnd mehrere Stellen besetzen, abbilden zu können, ist auch eine n:m Beziehung möglich. Besetzen Mitarbeiter mehrere Stellen gleichzeitig, müssen diese bei der Anmeldung an das PMS angeben, in welcher Stelle sie gerade aktiv sind. Dadurch ist einem angemeldeten Benutzer stets eine eindeutige Stelle zugeordnet. An der Entität Stellen werden, neben der Mitarbeiterzuordnung, auch die Beziehung zu Rollen (*besitzt*) definiert. Ferner werden Vorgesetzten-Beziehungen (*ist übergeordnet*), die Zugehörigkeit zu (*gehört zu*) und die Leitung von Organisationseinheiten (*leitet*) an Stellen festgemacht. Da diese Beziehungen auf der Grundlage von Stellen festgelegt werden, bleiben diese Zurordnungen auch erhalten, wenn sich personelle Änderungen, wie etwa beim Ausscheiden eines Mitarbeiters aus dem Unternehmen, ergeben.

Fähigkeiten werden Mitarbeitern direkt zugeordnet und nicht den Stellen. Ein Mitarbeiter muss zwar ein Mindestmaß an bestimmten Fähigkeiten besitzen, um eine bestimmte Stelle besetzen zu können, darüberhinaus kann er aber durchaus weitere Fähigkeiten aufweisen.

Vertreterregelungen können sich sowohl auf Stellen als auch auf Mitarbeiter (oder Rollen) beziehen, da sie die Form einer Bearbeiterzuordnung (s. Kap. 4.2.3) haben. Zusätzlich ist die Vertreterregelung von den zu vertretenden Aufgaben eines Mitarbeiters abhängig, so können für einen Mitarbeiter mehrere Vertreter aufgabenspezifisch bestimmt werden.

Eine Organisationseinheit kann in beliebig vielen Organisationsgruppen Mitglied sein, wodurch Bearbeiterzuordnungen wie „Rolle = Arzt UND Organisationsgruppe = Chirurgie-Station“ erst möglich werden. Natürlich können die Organisationsgruppen auch zur Definition von Zugriffs- und Änderungsrechten verwendet werden. Stellen können zu Arbeitsgruppen zusammengefasst werden, so dass z. B. Projektgruppen definiert und verwaltet werden können. Auch sind dynamische Änderungen in der Zusammensetzung einer solchen Gruppe möglich.

Ähnlich wie bei der Leitung einer Organisationseinheit, wird die Leitung einer Arbeitsgruppe ebenfalls über die Stellen definiert.

4.2.3 Bearbeiterformeln

Für die Definition von Ausführungs-, Zugriffs- und Änderungsrechten muss auf das konkrete Organisationsmodell Bezug genommen werden. Konkret muss für Systemoperationen (z. B. Erzeugen einer Prozessinstanz, Hinzufügen eines neuen Schrittes zu einer bestimmten Prozessinstanz) festlegbar sein, welche Personen zur Durchführung berechtigt sind. Als Grundlage hierfür werden Bearbeiterformeln eingeführt, mit denen sich eine Teilmenge der Personen eines gegebenen Organisationsmodells festlegen bzw. ermitteln lässt. Diese Bearbeiterformeln sollen möglichst robust sein gegenüber personellen Veränderungen.

Die Bearbeiterformeln (*BF*) setzen sich aus Bearbeitermengen (*BM*) zusammen, die wiederum aus Bearbeiterausdrücken (*BA*) bestehen. Grundlage von Bearbeiterausdrücken sind Bearbeiterzuordnungen (*BZ*). Im Folgenden sollen die einzelnen Bestandteile kurz und in vereinfachter (normierter) Form dargestellt werden.

Bearbeiterzuordnungen werden in abhängige, unabhängige und dynamische Bearbeiterzuordnungen unterschieden. Unabhängige Bearbeiterzuordnungen bzw. die möglichen Bearbeiter unabhängiger Bearbeiterzuordnungen ergeben sich in der Regel aus einem einfachen Ausdruck und sind wie in Tab. 4.1 aufgebaut und haben die zugehörige Bedeutung.

BZ	Bedeutung
$R = X$	Benutzer hat Rolle X
$R+ = X$	Benutzer hat angegebene oder eine spezieller Rolle
$MA = X$	Benutzer ist angegebener Mitarbeiter X
$F = X$	Benutzer hat angegebene Fähigkeit X
$OG = X$	Benutzer ist Mitglied einer Organisationseinheit, die zur Organisationsgruppe X gehört
$AG = X$	Benutzer ist Mitglied der Arbeitsgruppe X
$S = X$	Benutzer besetzt die Stelle X

Tabelle 4.1: Beispiel für verschiedene unabhängige Bearbeiterzuordnungen

Abhängige Bearbeiterzuordnungen beziehen sich auf die Stelle des Bearbeiters einer referenzierten Aktivität. Von dieser Stelle ausgehend wird ggf. die Organisationseinheit oder die Arbeitsgruppe ermittelt, zu der die spezifizierte Stelle gehört. Die abhängigen Bearbeiterzuordnungen sind mit der entsprechenden Bedeutung wie in Tab. 4.2 aufgebaut.

Bei $n=0$ (d. h. n ist nicht vorhanden) qualifizieren sich die Stellen der entsprechenden Organisationseinheit bzw. Arbeitsgruppe. Bei ± 1 , ± 2 wird die „ist-übergeordnet“-Beziehung entsprechend oft nach oben/unten verfolgt.

Ein einfaches Beispiel zur Erklärung einer abhängigen Bearbeiterzuordnung ist in Abb. 4.5 skizziert.

Desweiteren gibt es noch eine spezielle Form der Bearbeiterzuordnungen. Im Allgemeinen sollte es möglich sein, dass eine Bearbeiterformel auf Prozessvariablen bzw. Datenelemente Bezug nimmt. Dadurch wird ermöglicht, dass Bearbeiter oder Organisationseinheiten auch noch dynamisch zur Laufzeit durch eine Anwendung bzw. deren Anwender festgelegt werden.

BZ	Bedeutung
$S = X$	Benutzer besetzt die Stelle X
$S+n = X$	Benutzer ist (n-facher) Vorgesetzter der Stelle X
$S+ = X$	Benutzer ist Vorgesetzter (beliebiger Ebene) der Stelle X
$S-n = X$	Benutzer ist (n-facher) Untergebener der Stelle X
$S- = X$	Benutzer ist Untergebener (beliebiger Ebene) der Stelle X
$OE = X$	Benutzer gehört der Organisationseinheit X an
$OE+n = X$	Benutzer ist Mitglied einer Organisationseinheit, die der Organisationseinheit X (n-fach höher) übergeordnet ist
$OE+ = X$	Benutzer ist Mitglied einer Organisationseinheit, die der Organisationseinheit X (auf beliebiger Ebene) übergeordnet ist (die OE X selbst ist dabei nicht eingeschlossen)
$OE-n = X$	Benutzer ist Mitglied der Organisationseinheit, die der Organisationseinheit X (n-fach niedriger) untergeordnet ist
$OE- = X$	Benutzer ist Mitglied einer Organisationseinheit, die X (auf beliebiger Ebene) untergeordnet ist
$OEI = X$	Leiter der Organisationseinheit X
$OEu = X$	die Mitarbeiter der Organisationseinheit X
$AG = X$	Benutzer, die der Arbeitsgruppe X angehören
$AGI = X$	Leiter der Arbeitsgruppe X (bei abhängigen Bearbeiterzuordnungen: die Leiter der Arbeitsgruppen, zu denen der Bearbeiter der referenzierten Aktivität gehört)
$AGu = X$	die Mitarbeiter der Arbeitsgruppe X

Tabelle 4.2: Beispiele für abhängige Bearbeiterzuordnungen

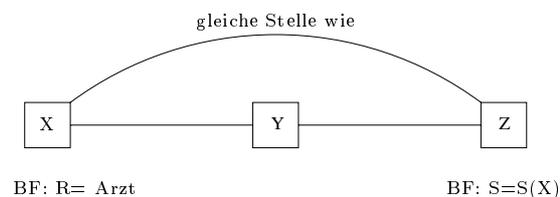


Abb. 4.5: Beispiel einer abhängigen Bearbeiterzuordnung

Durch eine dynamische Festlegung der Bearbeiter ist es möglich, z.B. Berechtigungen bei der Verlegung eines Patienten von einer Station auf eine andere ebenso weiterzugeben wie die optimale Weiterleitung von Befunden an die neue Station zu gewährleisten. Durch die Verwendung dynamischer Bearbeiterzuordnungen wird eine alte Bearbeiterzuordnung automatisch an die neue Form angepasst.

Ein Beispiel zur Realisierung dieser dynamischen Komponente ist in Abb. 4.6 skizziert. Die Aktivität „Untersuchung anordnen“ schreibt die behandelnde Organisationseinheit in die Variable x . Nach der Durchführung der Untersuchung wird der Patient auf eine andere Station verlegt. Damit der Befund an die neue Station gesendet wird, muss dem PMS mitgeteilt werden, dass sich die Organisationseinheit des Patienten geändert hat. Über ein API wird die neue Organisationseinheit in die Variable x geschrieben. In Zukunft werden bei der Auswertung eines Bearbeiterausdrucks dieser Instanz nur die Mitarbeiter der neuen Organisationseinheit berücksichtigt.

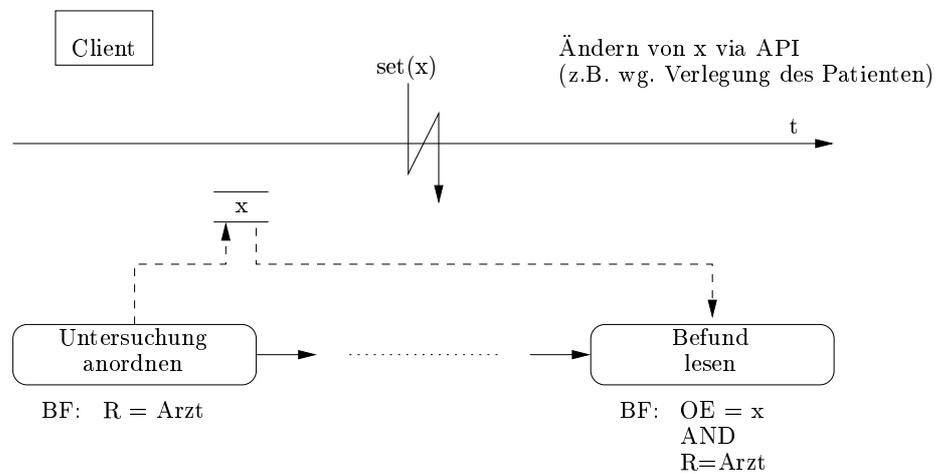


Abb. 4.6: Beispiel für eine dynamisch festgelegte Bearbeiterzuordnung. Durch $set(x)$ wird die Belegung der Variable x extern geändert.

Mehrere Bearbeiterzuordnungen miteinander kombiniert, ergeben einen *Bearbeiterausdruck*. Formel ist ein Bearbeiterausdruck folgendermaßen aufgebaut:

$$BA := BZ \{ op^1 BZ \}.$$

Tab. 4.3 zeigt einige kleine informelle Beispiele für Bearbeiterausdrücke:

Werden mehrere Bearbeiterausdrücke miteinander kombiniert, ergibt sich eine *Bearbeitermenge*. Der formale Aufbau einer Bearbeitermenge sieht wie folgt aus:

$$BM := BA \{ op^1 BA \}.$$

Aus der Kombination mehrerer Bearbeitermengen resultiert die *Bearbeiterformel*. Ihr formeller Aufbau ist der folgende:

$$BF := BM \{ op^1 BM \}.$$

Im Folgenden ist ein Beispiel für eine Bearbeiterformel mit der Bedeutung „Benutzer besitzt entweder die Rolle *Arzt* oder ist *Pflegekraft* mit der speziellen Berechtigung *Level 3* und ist

¹ $op \in \{ AND, ANDNOT, OR, \dots \}$

BA	Bedeutung
Rolle = Arzt AND OE = Chirurgie	Benutzer ist <i>Arzt</i> und gehört der Organisationseinheit <i>Chirurgie</i> an
MA = Meier	Benutzer ist der Mitarbeiter <i>Meier</i>
AG = Arbeitsgruppe XY	Benutzer ist Mitglied der Arbeitsgruppe X
R = Pflegekraft AND F = Blutabnahme	Benutzer ist <i>Pflegekraft</i> und besitzt die Fähigkeit zur <i>Blutabnahme</i>
S = xyz	Benutzer besetzt die Stelle <i>xyz</i>
OG = Kliniken	Benutzer gehört irgendeiner Klinik an
(OE = Frauenklinik AND R = Oberarzt) ANDNOT MA = Müller	Benutzer ist <i>Oberarzt</i> der <i>Frauenklinik</i> und heißt nicht <i>Müller</i>

Tabelle 4.3: Beispiel für Bearbeiterausdrücke

auf keinen Fall der Mitarbeiter mit Namen *Meier*“. Dieser Bearbeiterformel sind direkt die verschiedenen Komponenten einer Bearbeiterformel zugeordnet.

$$\underbrace{\underbrace{(R = \text{Arzt})}_{BZ}}_{BA} \text{ OR } \underbrace{\underbrace{(R = \text{Pflege})}_{BZ} \text{ AND } \underbrace{(F = \text{Level3})}_{BZ}}_{BA}}_{BM} \text{ ANDNOT } \underbrace{\underbrace{(MA = \text{Meier})}_{BZ}}_{BA}}_{BM}$$

4.3 Prozessausführung in ADEPT

Die Aufgabe eines Prozess-Management-Systems ist es, den Benutzer in der Ausführung seiner Arbeitsabläufe optimal zu unterstützen. Dazu werden die Abläufe analysiert und in Form von Ablaufdiagrammen abgebildet. Wünschenswert wäre es, wenn alle Abläufe von vornherein genau bekannt sind und diese auch nie geändert werden. In diesem Fall ist auf keine nicht planbaren Ausnahmen Rücksicht zu nehmen und die verschiedenen Berechtigungen wären klar definierbar. In der Praxis ist es aber der Fall, dass in den Arbeitsabläufen auch einmalige Änderungen auftreten können.

Im Folgenden soll sowohl auf die standardmäßige Ausführung einer Prozessinstanz als auch auf Abweichungen vom Standardablauf eingegangen werden.

4.3.1 Standardablauf

Neue Prozessinstanzen können in ADEPT auf der Grundlage der im ADEPT-PMS hinterlegten Prozessvorlagen bzw. -modelle durch berechnigte Benutzer erzeugt werden. Die Ausführung einer Prozessinstanz wird dann über ihre komplette Lebensdauer hinweg, ggf. also auch über Tage, Wochen oder Monate vom ADEPT-PMS begleitet. Das System koordiniert die fristgerechte Durchführung der Aktivitäten der Prozessinstanz gemäß der definierten Ablauflogik, bietet anstehende Tätigkeiten potentiellen Bearbeitern priorisiert über Arbeitslisten an, ruft automatisch die mit einer Aktivität verknüpfte Anwendungskomponente beim Start der Aktivität auf, und ist für die Versorgung und Übernahme der Eingabedaten geruener Anwendungskomponenten zuständig. Sämtliche der im Kontext einer bestimmten Prozessausführung relevanten Ereignisse (z. B. Start bzw. Ende einer Aktivitätenbearbeitung) werden vom ADEPT-PMS in der zugehörigen *Ausführungshistorie* protokolliert. Dabei wer-

den Informationen wie Start-/Ende-Zeiten, Bearbeiter oder Rückgabewerte gespeichert. Insgesamt bietet das ADEPT-PMS sehr mächtige Möglichkeiten zur Steuerung von Prozessinstanzen. Abhängig von der gewählten Modellierung, können die Aktivitäten eines Prozesses sequentiell, parallel, bedingt oder iterativ durchgeführt werden. Weitere Details finden sich in [Reic00b].

4.3.2 Abweichungen vom Standardablauf

Eigentlich sollten die systemseitig unterstützten Prozesse immer den in der Realität ablaufenden Vorgängen entsprechen. In der Praxis ist es jedoch nahezu unmöglich, alle Prozessinstanzen starr nach demselben Schema abzuwickeln. Stattdessen sollte es, wie in [Reic00b] beschrieben, Möglichkeiten geben, im PMS in sehr flexibler Weise vom vormodellierten Prozess abzuweichen. Dabei muss zwischen zwei verschiedenen Arten von Abweichungen unterschieden werden – planbaren Abweichungen und Ad-hoc Abweichungen. *Planbare Abweichungen* sind Abweichungen vom Standardablauf, deren Auftreten bereits zur Modellierzeit bekannt ist und die deshalb im Prozessmodell – zumindest in gewissem Umfang – Berücksichtigung finden können. *Ad-hoc Abweichungen* sind Abweichungen, die bei der Modellbildung der Prozessvorlage nicht absehbar und damit auch nicht modellierbar sind. Da viele Prozesse eine hohe Variabilität besitzen, ist die Vormodellierung aller Abweichungen schon aus kombinatorischen Gründen nicht immer möglich bzw. sinnvoll [Meye96].

4.3.2.1 Planbare Abweichungen

Planbare Abweichungen sind, wie oben beschrieben, zur Modellierungszeit bekannt und können daher im Prozessmodell berücksichtigt werden. Beispielsweise kann es im Kontext eines Untersuchungsablaufs (vgl. Kap 3.2) in Notfallsituationen erforderlich sein, den Patienten sofort der Untersuchungseinheit zuzuführen, ohne die ansonsten übliche Terminvereinbarung und Vorbereitung des Patienten zu treffen. D. h. es müssen evtl. Schritte übersprungen werden. ADEPT bietet mächtige Modellkonstrukte, um solche vorhersehbaren Abweichungen im Prozessmodell a priori abbilden zu können (A-priori-Flexibilität). Konkret ist es möglich, folgende Abweichungen vorzumodellieren:

- Vorwärtssprünge mit/ohne Nachholen der übersprungenen Schritte (sog. *Shortcuts*²)

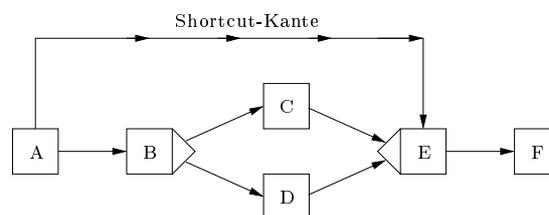


Abb. 4.7: Beispiel für einen Shortcut (aus [Reic00b])

- automatische Fehlerrücksprünge beim Scheitern einer Aktivitätenbearbeitung (*Rücksprung über Fehlerkanten*)(s. Abb. 4.8)
- benutzerinitiierte Rücksprünge, mit dem Ziel, die Kontrolle über den Ablauf einer Prozessinstanz wiederzuerlangen (*Regain Control*)

²Im Beispiel wird nach Beendigung der Aktivität A sowohl Aktivität B als auch Aktivität E aktiviert, d. h. würde direkt die Aktivität E zum Starten selektiert, würden die Aktivitäten B, C und D ausgelassen werden.

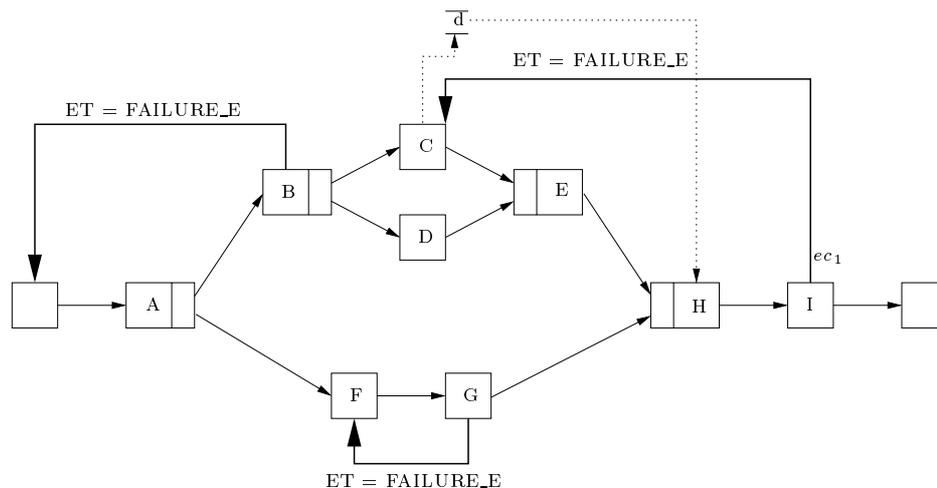


Abb. 4.8: Beispiel für einen Rücksprung über eine Fehlerkante (aus [Reic00b])

Werden derartige Abweichungen bereits zur Modellierungszeit berücksichtigt, werden zum einen der Aufwand der teureren Modifikationen auf WF-Instanzebene und damit verbundene Konsistenzprüfungen von vornherein reduziert, zum anderen sorgt eine solche Vormodellierung für ein höheres Maß an Flexibilität bei der Bearbeitung von Prozessen.

Um die „normale“ Ausführungsreihenfolge von ausnahmebedingten Abweichungen unterscheiden zu können, müssen diese außerplanmäßigen Ablaufpfade dem System kenntlich gemacht werden. Dazu werden in ADEPT *Knotenprioritäten* und *Priorisierungskanten* verwendet. Zur Modellierung planbarer Rücksprünge gibt es in ADEPT zudem die Möglichkeit, diese über *Fehlerkanten* zu spezifizieren. Auf die Beschreibung dieser drei Konstrukte soll im Folgenden eingegangen werden.

Neben der Modellierung von Ausnahmeschritten, muss dem Endanwender auch angezeigt werden, ob es sich bei einer angebotenen Aktivität um einen Ausnahmeschritt oder eine „normale“ Aktivität handelt. Die Kennzeichnung der Ausführungsarten von Aktivitäten kann auf unterschiedliche Weise vorgenommen werden. Zum Beispiel können die Arbeitslisteneinträge für Ausnahmeschritte in einer anderen Farbe erscheinen als die normalen Schritte. Auch können die Ausnahmeschritte in einer zweiten Arbeitsliste stehen oder sie werden erst auf direkte Auswahl überhaupt in der Arbeitsliste sichtbar. Weitere Darstellungsmöglichkeiten finden sich in [Saeu96]. Durch die Kennzeichnung der Ausführungsarten der Aktivitäten kann der Benutzer die Konsequenzen seiner Interaktionen mit dem System besser verstehen und nachvollziehen.

Knotenpriorisierungen und Priorisierungskanten

Die im Abschnitt 4.2.1 vorgestellten Modellierungskonstrukte von ADEPT legen die Aktivitäten eines Arbeitsablaufes sowie deren „normale“ Ausführungsreihenfolgen und -bedingungen fest. Um eine WF-Aktivität als regulären Arbeitsschritt oder Ausnahmeschritt kenntlich zu machen, werden, wie in [Reic00b] beschrieben, zunächst Knotenprioritäten eingeführt. Jedem Aktivitätenknoten wird bei der Modellierung eine Ausführungspriorität zugewiesen. Handelt es sich um einen regulären Arbeitsschritt, erhält er die Priorität *REGULAR*, ist es ein Ausnahmeschritt, erhält er die Priorität *EXCEPTIONAL*. Als Voreinstellung wird jedem Knoten die Priorität *REGULAR* zugewiesen. Die Aktivierung und Ausführung einer Ausnahmeaktivität mit Priorität *EXCEPTIONAL* erfolgt nach denselben Regeln wie im Fall regulärer Arbeitsschritte.

Abb. 4.9 skizziert ein Beispiel für die unterschiedliche Priorisierung und Ausführung von Aktivitäten. In dem dargestellten Ausführungsgraphen sind die Aktivitäten B und D gleichzeitig aktiviert, wobei B als regulärer Schritt und D als Ausnahmeschritt angeboten werden. Die präferierte Ausführungsreihenfolge ist demnach A, B, C, E. Nach Beendigung von Aktivität A kann aber anstelle der Aktivitäten B und C auch Aktivität D ausgeführt werden. Dies ist z. B. dann sinnvoll, wenn bestimmte Ausnahmesituationen die Ausführung dieser Aktivität zwingend erforderlich machen und die beiden anderen Aktivitäten – B und C – übersprungen werden sollen.

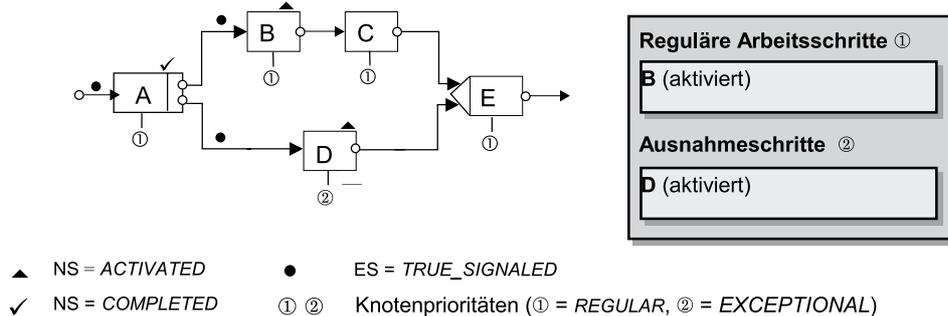


Abb. 4.9: Verwendung von Knotenprioritäten in ADEPT (aus [Reic00b])

Neben der Kennzeichnung von Aktivitätsprioritäten, sollte die Möglichkeit bestehen, diese während der Ausführung zu verändern. In gewissen Zuständen der Prozessinstanz etwa soll eine Aktivität als Ausnahmeschritt gekennzeichnet, zu einem späteren Zeitpunkt aber als regulärer Schritt ausführbar sein. Soll z. B. die Bearbeitung der Aktivität X in Ausnahmefällen vorzeitig erfolgen können, im Normalfall aber erst nach Abschluss bestimmter anderer Schritte, so stellt die Ausführung der Aktivität X zu gewissen Zeitpunkten der Prozessausführung eine Abweichung vom Standardablauf dar, während sie zu einem späteren Zeitpunkt regulären Charakter besitzt und die Aktivität dementsprechend als normaler Arbeitsschritt angeboten werden sollte. Mithilfe von statischen Knotenpriorisierungen allein lässt sich dieser Sachverhalt nicht modellieren. Dazu werden in ADEPT Priorisierungskanten verwendet. Sie dienen der Veränderung der Ausführungspriorität von Aktivitäteninstanzen während der Laufzeit. Die Priorisierungskante ist mit einer Kantenpriorität verknüpft, die die Werte *REGULAR* oder *EXCEPTIONAL* annehmen kann. Wird eine Priorisierungskante zu einer Ausnahmeaktivität während der Ausführung eines Prozesses aktiviert bzw. mit *TRUE* signalisiert, wird die Ausführung ihres Zielknotens entsprechend abgeändert. D. h. besitzt ein aktivierter Knoten die Priorität *EXCEPTIONAL* und wird eine in diesen Knoten einmündende Priorisierungskante mit der Priorität *REGULAR* neu signalisiert, so wird die Ausführungspriorität des Knotens auf *REGULAR* gesetzt. Die Ausführung dieses Knotens stellt dann keine Abweichung vom Standardablauf mehr dar und die Aktivität kann wie ein normaler Arbeitsschritt bearbeitet werden.

Abb. 4.10 zeigt ein entsprechendes Beispiel. Aktivität D wird zunächst als Ausnahmeschritt dargestellt (Abb. 4.10 a) und zu einem späteren Zeitpunkt – nach Setzen der Priorisierungskante $C \rightarrow D$ auf *TRUE_SIGNED* – als regulärer Schritt angeboten (Abb. 4.10 b).

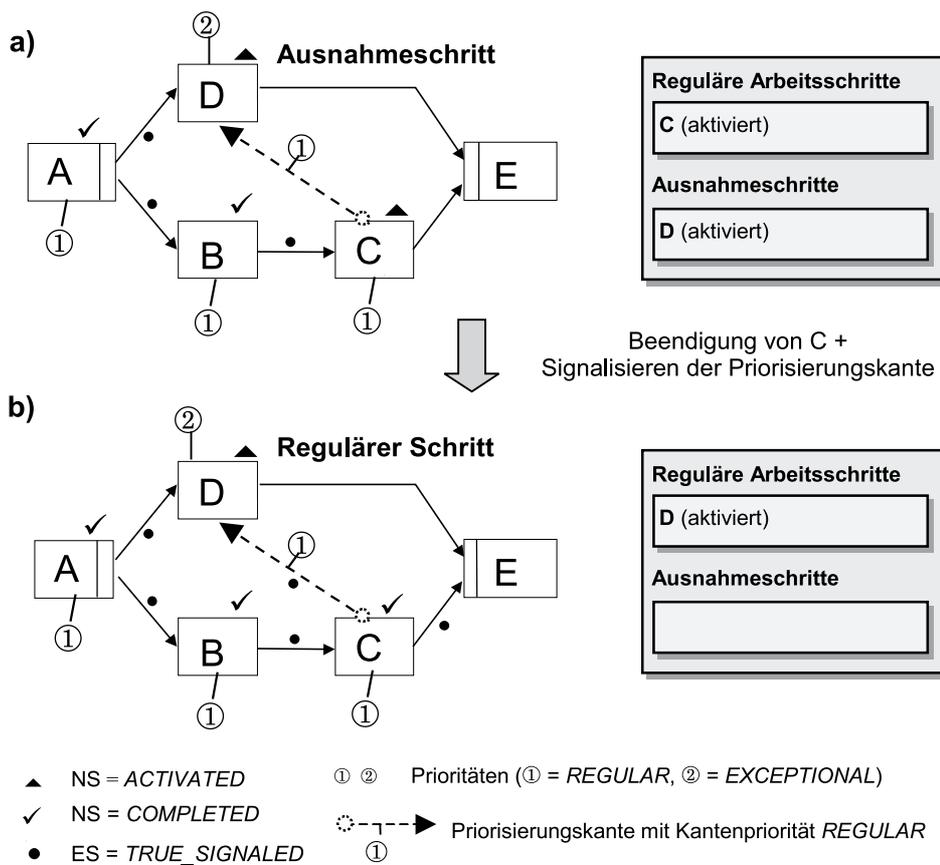


Abb. 4.10: Änderung der Ausführungspriorität von Aktivitätenknoten (aus [Reic00b])

Überspringen von Aktivitäten mit/ohne Nachholen (Shortcuts)

Wie in [Reic00b] beschrieben, ist es in bestimmten Fällen situationsbedingt notwendig, Aktivitäten früher als normal durchzuführen. Damit derartige Abweichungen optimal unterstützt werden können, muss autorisierten Anwendern die Möglichkeit gegeben werden, einzelne Aktivitäten oder Folgen von Aktivitäten zu überspringen und sie entweder nicht mehr oder zu einem späteren Zeitpunkt nachzuholen. In ADEPT lassen sich solche Vorwärtssprünge durch die Verwendung von *Shortcuts* vormodellieren. Voraussetzung ist, dass die Prozesszustände, in denen eine solche Abweichung erfolgen darf und der Zielknoten des Vorwärtssprungs, a priori bekannt sind. Aus der Sicht des Modellierers wird einfach eine gerichtete Verbindung zwischen dem Quell- und dem Zielknoten des Vorwärtssprungs im KF-Graphen hergestellt.

Abb. 4.11 zeigt sowohl die Sicht des Modellierers (Abb. 4.11 a) als auch deren Umsetzung in ADEPT (Abb. 4.11 b). Die Shortcut-Kante von A nach E beschreibt einen vorgeplanten Vorwärtssprung, wobei die übersprungenen Aktivitäten B, C und D nicht nachgeholt werden sollen. Ist die Aktivität A erfolgreich abgeschlossen, bekommt der Anwender sowohl die Aktivität B als auch den Ausnahmeschritt „Springe zu E“ zur Bearbeitung in seiner Arbeitsliste angeboten. Wird die Aktivität „Springe zu E“ vorzeitig gestartet, müssen die übersprungenen Schritte gesondert behandelt werden. Sie müssen, je nach ihrem Ausführungsstatus, kompensiert, abgebrochen oder deaktiviert werden. Dazu muss für jede

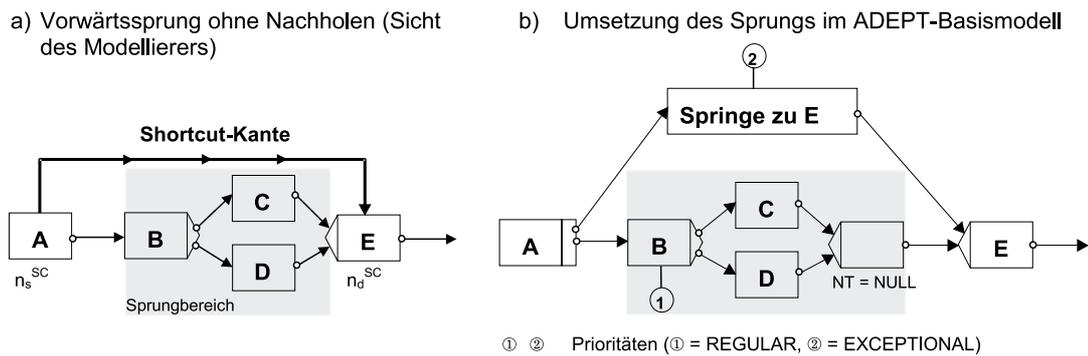


Abb. 4.11: Modellierung von Vorwärtssprüngen ohne Nachholen und deren Umsetzung in ADEPT (aus [Reic00b])

Aktivität festgelegt sein, ob sie kompensierbar oder abbrechbar ist.

Im oben angeführten Beispiel werden die übersprungenen Schritte nicht nachgeholt, es handelt sich also um einen *Vorwärtssprung ohne Nachholen*. Wurde eine Aktivität als Ausnahmeschritt über einen Shortcut zur Bearbeitung ausgewählt, wird nach der Bearbeitung dieser Aktivität am Sprungziel mit der Bearbeitung der Prozessinstanz fortgefahren. Ein Vorwärtssprung ohne Nachholen wird in ADEPT durch die Verwendung einer parallelen Verzweigung mit finaler Auswahl umgesetzt. Dies ist in Abb. 4.11 b) dargestellt. Der Kontrollfluss-Graph besteht aus zwei Teilgraphen, wobei der eine den normalen Prozessablauf wiedergibt und der andere lediglich die Aktivität „Springe zu Endknoten“ enthält. Diese Aktivität wird dann in der Arbeitsliste des Bearbeiters mit den entsprechenden Autorisierungen (s. Kap. 7) als Ausnahmeschritt hervorgehoben. Wird dieser Eintrag von einem Bearbeiter ausgewählt, wird er ohne weitere Aktion direkt wieder beendet, der untere Teilgraph abgebrochen und zurückgesetzt, und mit der Kontrolle des Knotens E wird fortgefahren. Wird dagegen der untere Teilzweig zur Bearbeitung ausgewählt und Knoten E kommt auf normalem Wege zur Ausführung, wird der obere Teilgraph deaktiviert und aus den zugehörigen Arbeitslisten entfernt.

Bei der Umsetzung der Modellierung in ADEPT finden die in Abschnitt 4.3.2.1 beschriebenen Knotenprioritäten Anwendung.

Bisher wurden Vorwärtssprünge ohne Nachholen betrachtet, es müssen aber auch *Vorwärtssprünge mit Nachholen* der übersprungenen Schritte möglich sein. In diesem Fall soll die Bearbeitung der übersprungenen Schritte nicht abgebrochen, kompensiert oder deaktiviert werden, sondern sie sollen parallel zu der vorgezogenen Aktivität weiter bearbeitet bzw. nachgeholt werden. D. h. Effekte bereits beendeter Aktivitäten bleiben erhalten, bereits gestartete Aktivitäten werden fortgesetzt und noch nicht aktivierte Aktivitäten sollen weiterhin ausführbar bleiben, wenn der Vorwärtssprung durchgeführt wird. Dazu muss vom Modellierer ein zusätzlicher Schritt angegeben werden, bis zu dem alle nachzuholenden Aktivitäten abgeschlossen sein müssen, d. h. Vorwärtssprünge mit Nachholen werden in ADEPT durch die Verwendung einer parallelen Verzweigung umgesetzt.

In Abb. 4.12 ist ein Beispiel für Vorwärtssprünge mit Nachholen der übersprungenen Aktivitäten skizziert. Nach erfolgreicher Beendigung der Aktivität A wird sowohl Aktivität B als auch Aktivität E zur Bearbeitung angeboten, wobei Aktivität E als Ausnahmeschritt in der Arbeitsliste erscheint. Beide Zweige der parallelen Verzweigung müssen bis zur Aktivierung der Aktivität H beendet sein. Wird Aktivität E nicht als Ausnahmeschritt gestartet, sondern

in der normalen Ausführungsreihenfolge nach Beendigung der Aktivitäten C und D, wird über die Priorisierungskante die Knotenpriorität von E auf einen regulären Ausführungsstatus gesetzt und die Ausführung des Prozesses läuft normal weiter. Wird Aktivität E aber als Ausnahmeschritt gestartet, muss der andere Teilzweig bestehend aus den Aktivitäten B, C und D weiter ausgeführt werden. Nach erfolgreicher Beendigung von C und D hat die Priorisierungskante dann allerdings keine Auswirkungen mehr auf die Ausführungspriorität der Aktivität E, da diese bereits gestartet bzw. ausgeführt wird oder wurde.

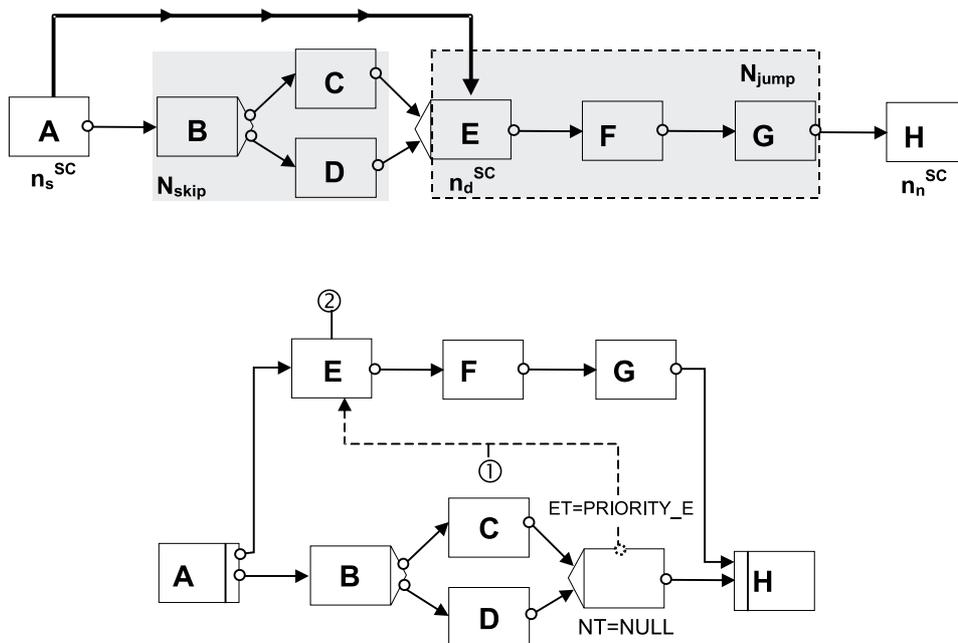


Abb. 4.12: Modellierung von Vorwärtssprüngen mit Nachholen und Umsetzung in ADEPT (aus [Reic00b])

Rücksprung über Fehlerkanten

Neben der Behandlung von Vorwärtssprüngen im Standardablauf eines Prozesses, müssen auch Verhaltensweisen für den umgekehrten Fall vorhanden sein, um z. B. den Fehlschlag einer Aktivitätenbearbeitung auszugleichen. Für diesen Fall müssen Rückwärtssprünge im Kontrollfluss erlaubt sein. ADEPT unterstützt das Zurücksetzen laufender oder bereits abgeschlossener Arbeitsschritte in unterschiedlichen Zusammenhängen:

- Bei Parallelverzweigungen mit finaler Auswahl werden im Anschluss an die Selektion des „Siegerezweiges“ nicht beendete Teilzweige abgebrochen und zurückgesetzt.
- Führen Ausnahmen zum Scheitern einer Aktivitätenbearbeitung, kann eine Prozessinstanz (teilweise) zurückgesetzt und erneut ausgeführt werden.
- Speziell autorisierte Anwender des Systems können in die Kontrolle der Prozessinstanz eingreifen, indem sie seine Ausführung unterbrechen und (partiell) zurücksetzen. Dadurch werden zuvor getroffene Verzweigungsentscheidungen revidiert oder Fehler vorausgegangener Arbeitsschritte korrigiert.

Diese Beispiele zeigen auf, dass ein das Rücksetzen verursachende Ereignis im vornherein bekannt sein kann, aber nicht muss. Das PMS sollte sowohl automatische Rücksprünge auf

der Grundlage der ihm vorliegenden Informationen durchführen können, als auch durch den Benutzer initiierte Rücksprünge zulassen.

Soll eine Prozessinstanz (partiell) zurückgesetzt werden, muss die Bearbeitung unterbrochen und in einen Zustand zurückgesetzt werden, der vor dem Start einer bestimmten Aktivitäteninstanz Gültigkeit besaß. Dabei müssen die ursprünglichen Knoten- und Kantenmarkierungen und die Werte seiner Datenelemente wiederhergestellt werden. Außerdem müssen auch externe Effekte laufender oder bereits abgeschlossener Aktivitäten zurückgesetzt werden. Das Rücksetzen der Daten und Markierungen muss derart vorgenommen werden, dass die Bearbeitung direkt am spezifizierten Rücksprungzielknoten wieder aufgenommen werden kann. Das Zurücksetzen sowohl der Prozessinstanz als auch der Daten geschieht auf Grundlage der Ausführungshistorie der Instanz und der Historie der einzelnen Datenelemente (für Details s. [Reic00b]).

Für die Umsetzung der Vormodellierung ausnahmebedingter Rücksprünge bietet ADEPT sogenannte *Fehlerrücksprungkanten* (kurz: Fehlerkanten) an. Jeder Aktivität werden durch seine Vorlage eine Menge von Fehlercodes zugeordnet, die a priori bekannte Ausnahmen, die bei der Bearbeitung von Instanzen einer Aktivität auftreten und deren Scheitern verursachen können, beschreiben. Tritt eine derartige Ausnahme auf, wird von der Aktivität der zugehörige Fehlercode gesetzt und die Bearbeitung der Aktivität abgebrochen. ADEPT kann auf diesen Fehlercode auf unterschiedliche Weise reagieren:

- Wiederholen der Bearbeitung
- Ausführen einer alternativen Tätigkeit
- Überspringen der Aktivität
- Partielles Zurücksetzen der Prozessinstanz in einen früheren Bearbeitungszustand mit anschließender Wiederaufnahme der Bearbeitung
- Kontrolliertes Abbrechen der Prozessinstanz
- Zurücksetzen der Prozessinstanz nach dynamischer Änderung des Ausführungsgraphen

Restriktionen für die Verwendung von Fehlerkanten werden in [Reic00b] vorgestellt. Beispiele für die Modellierung von Fehlerkanten sind in Abb. 4.13 skizziert.

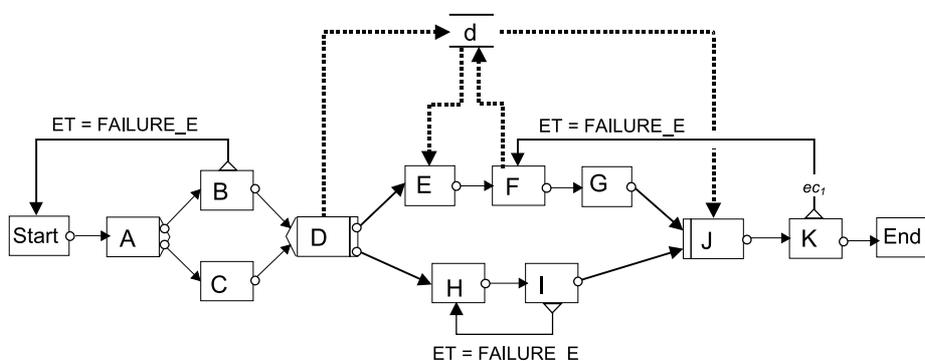


Abb. 4.13: Modellierung von Rücksprüngen durch Fehlerkanten

Benutzerinitiierte Rücksprünge

Neben den Vorwärtssprüngen mit und ohne Nachholen übersprungener Schritte und den (automatischen) Rücksprüngen über Fehlerkanten gibt es noch eine weitere Möglichkeit, die Bearbeitung von Prozessen zurückzusetzen. Will der Benutzer die Kontrolle über den Ablauf einer Prozessinstanz zurückgewinnen, kann er selbst den Rücksprung initiieren. In diesem Fall spricht man von *benutzerinitiierten Rücksprüngen* oder auch *RegainControl*. Sie setzen sich aus den bereits oben beschriebenen Verfahren zusammen. Genauer soll auf die benutzerinitiierten Rücksprünge hier jedoch nicht eingegangen werden.

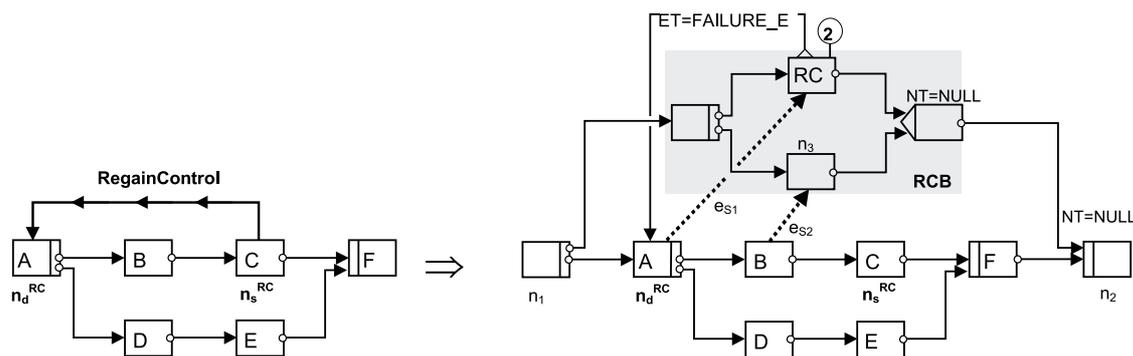


Abb. 4.14: Umsetzung von RegainControl-Rücksprüngen in ADEPT (aus [Reic00b])

Ein Beispiel für einen benutzerinitiierten Rücksprung und deren Umsetzung in ADEPT gibt Abb. 4.14 wieder. Über die RegainControl-Kante von C nach A wird ein planbarer Rücksprung beschrieben. Diese Kante ist nach Beendigung der Aktivität A solange aktiviert und ausführbar, bis Aktivität E gestartet wird. Wird der Rücksprung durchgeführt, wird die Prozessinstanz in den Zustand zurückgesetzt, der vor Ausführung von Aktivität A Gültigkeit besaß.

4.3.2.2 Ad-hoc Abweichungen

Nicht immer sind Abweichungen vom Standardablauf eines Prozesses von vornherein bekannt und können somit auch nicht vormodelliert werden. In diesem Fall müssen während der Laufzeit eines Prozesses noch Änderungen im Ablauf vorgenommen werden. Man spricht hier von *Ad-hoc Änderungen*. Diese Änderungen sollen sowohl auf einer Prozessvorlage durchgeführt werden können als auch bei einer bereits laufenden Prozessinstanz. Wurden die Änderungen auf der Vorlage durchgeführt, muss geprüft werden, ob und in wieweit die Änderungen auch auf bereits laufende Instanzen propagiert werden können. Änderungen, die auf eine laufende Instanz angewendet werden sollen, werden i.d.R. von an dieser Prozessinstanz beteiligten Personen vorgenommen, wohingegen Änderungen an der Vorlage meist durch den Prozessmodellierer vorgenommen werden. Egal, auf welcher Ebene die Änderungen vorgenommen werden, muss darauf geachtet werden, dass Korrektheit und Konsistenz eines Prozesses beibehalten bleiben. Die Korrektheit wird dadurch erreicht, dass die regelmäßige Blockstrukturierung erhalten bleibt. Konsistenz heißt u.a., dass der Ausführungsgraph einer Prozessinstanz zulässige Zustandmarkierungen hat, d. h. durch die Änderungen entsteht keine Graphmarkierung, die nicht auch entstehen könnte, wären die Änderungen bereits bei der Modellierung berücksichtigt worden.

Zu den Änderungen, die von ADEPT_{flex} unterstützt werden, gehören die folgenden Operationen:

- Hinzufügen, Löschen oder Verschieben einzelner Aktivitäten
- Bildung und Auflösung von Blockstrukturen
- Hinzufügen und Entfernen von Synchronisations-, Priorisierungs- und Fehlerkanten
- Hinzufügen und Entfernen von Datenelementen und Datenflusskanten
- Modifikation von WF-Attributen, wie z. B. Knotenprioritäten

Diese Änderungsoperationen sind auch in Kombination miteinander anwendbar und verwirklichen damit komplexere Änderungsoperationen. Die Änderungen in einer laufenden Prozessinstanz können so vorgenommen werden, dass sie temporär berücksichtigt und später automatisch wieder gelöscht werden. Sie können aber auch für den Rest der Laufzeit der Instanz erhalten bleiben.

Im Folgenden soll anhand des Beispiels des (dynamischen) Einfügens und Löschens von Aktivitäten und Kontrollblöcken die Vorgehens- und Funktionsweise erklärt werden.

(Dynamisches) Einfügen und Löschen von Aktivitäten und Kontrollblöcken

Grundsätzlich wird, wie in [Reic00b] beschrieben, für jede Prozessinstanz logisch eine eigene Kopie der Prozessvorlage, auch Ausführungsgraph genannt, verwaltet. Dies ermöglicht die Änderung einer einzelnen Instanz, ohne Auswirkungen auf andere Instanzen zu haben, die auf der gleichen Prozessvorlage beruhen. Der grundlegende Ablauf der Einfügeoperation ist folgender: Aus dem Vorlagenrepository wird für den einzufügenden Arbeitsschritt eine Aktivitätenvorlage ausgewählt und die Ausführung der Prozessinstanz unterbrochen. Nun werden die Änderungen genau definiert, d. h. die Einfügeposition, das Anknüpfen der Aktivitätenparameter an den Datenkontext usw. Sind diese Schritte durchgeführt, wird der Zustand der Instanz so angepasst, dass mit der Prozess-Kontrolle auf Basis des modifizierten Ausführungsgraphen fortgefahren werden kann.

Unterschieden werden drei Arten von Einfügeoperationen:

- **Seriell**es Einfügen: die Aktivität wird direkt zwischen zwei aufeinanderfolgenden Knoten im KF-Graphen eingefügt.
- **Paralleles** Einfügen: die Aktivität wird parallel zu einem Kontrollblock des WF-Graphen eingefügt. Hier entsteht also eine Verzweigung.
- Einfügen neuer Teilzweige: die Aktivität wird als neuer Teilzweig einer bereits bestehenden Verzweigung eingefügt.

Soll eine Prozessinstanz verändert werden, darf dieses nur geschehen, wenn zum einen der aktuelle Status dies zulässt und zum anderen die Änderung auch in der Prozessvorlage eingebracht werden könnte. Ein weiterer Punkt, der bei Änderungen an einer Prozessinstanz berücksichtigt werden muss, ist der, dass Änderungen nur an solchen Stellen im Ausführungsgraphen eingebracht werden dürfen, die noch nicht betreten wurden bzw. deren Ausführung noch nicht feststeht.

Serielles Einfügen

Das serielle Einfügen von Aktivitäten zwischen zwei aufeinanderfolgenden Aktivitäten stellt die einfachste Einfügeoperation dar.

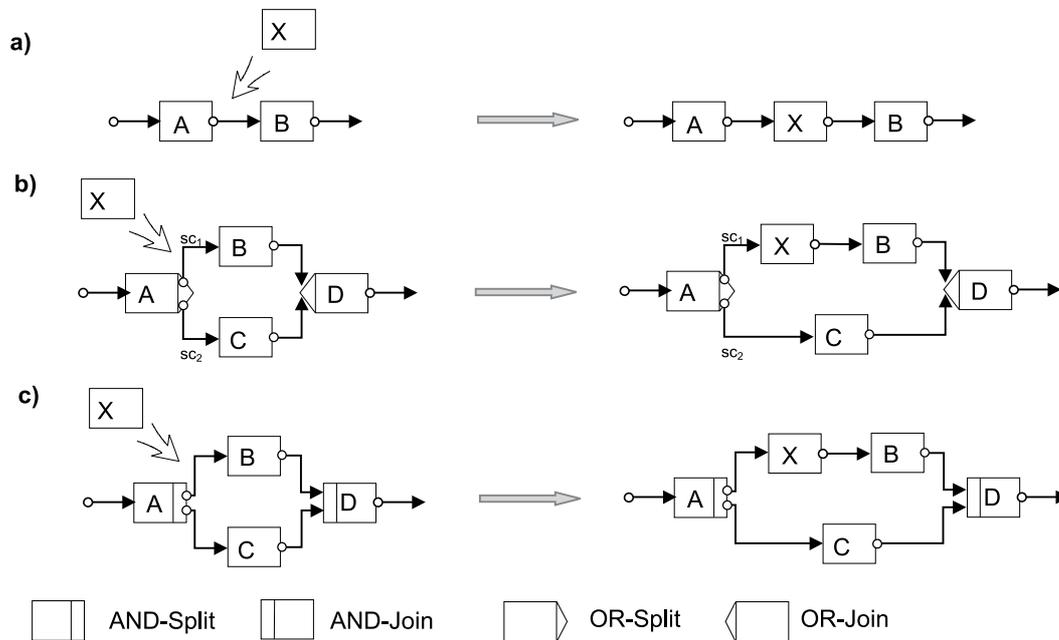


Abb. 4.15: Beispiel für serielles Einfügen einer Aktivität (aus [Reic00b])

Abb. 4.15 gibt drei Beispiele für das serielle Einfügen einer Aktivität zwischen zwei direkt aufeinanderfolgende Knoten. Aus den Beispielen b) und c) ist ersichtlich, dass die einzufügende Aktivität auch problemlos in eine Verzweigung eingebaut werden kann.

Beim seriellen Einfügen wird wie folgt vorgegangen:

- Auswahl der Aktivitätensvorlage
- Auslesen des Kantenattributs der Kante, in die eine Aktivität eingefügt werden soll
- Löschen der Kante zwischen den Aktivitäten, zwischen denen die neue Aktivität eingefügt werden soll
- Einfügen einer Kante zwischen der Aktivität, hinter der eingefügt wird und der neuen Aktivität und einer Kante zwischen der neuen und der nachfolgenden Aktivität
- Setzen der neuen Kantenattribute nach dem Einfügen einer Aktivität

Die formelle Beschreibung und die zugehörigen Restriktionen des *Seriellen Einfügens* finden sich in [Reic00b].

Paralleles Einfügen

Das parallele Einfügen dient dazu eine Aktivität parallel zu einem gegebenen Kontrollblock des Ausführungsgraphen einzufügen.

Abb. 4.16 zeigt den Ablauf und das Vorgehen der parallelen Einfügeoperation. Aktivität X

soll parallel zu dem Kontrollblock, der aus den Aktivitäten B, C, D, E und F gebildet wird, eingefügt werden. Dazu wird vor der Aktivität B ein AND-Split-Knoten und hinter Aktivität F ein AND-Join-Knoten eingefügt, wobei beide Knoten keine Aktivitäten beschreiben, sie sind sog. Nullaktivitäten. Der AND-Split-Knoten übernimmt dabei das Eingangsverhalten und die Eingangskontrollkanten von Aktivität B und der AND-Join-Knoten übernimmt entsprechend das Ausgangsverhalten und die ausgehenden Kontrollkanten von Aktivität F. Dann wird die neue Aktivität zwischen den beiden Nullaktivitäten parallel zu dem anderen Kontrollblock eingefügt. Im Anschluss wird der entstandene Graph, sofern dieses möglich ist, wieder vereinfacht.

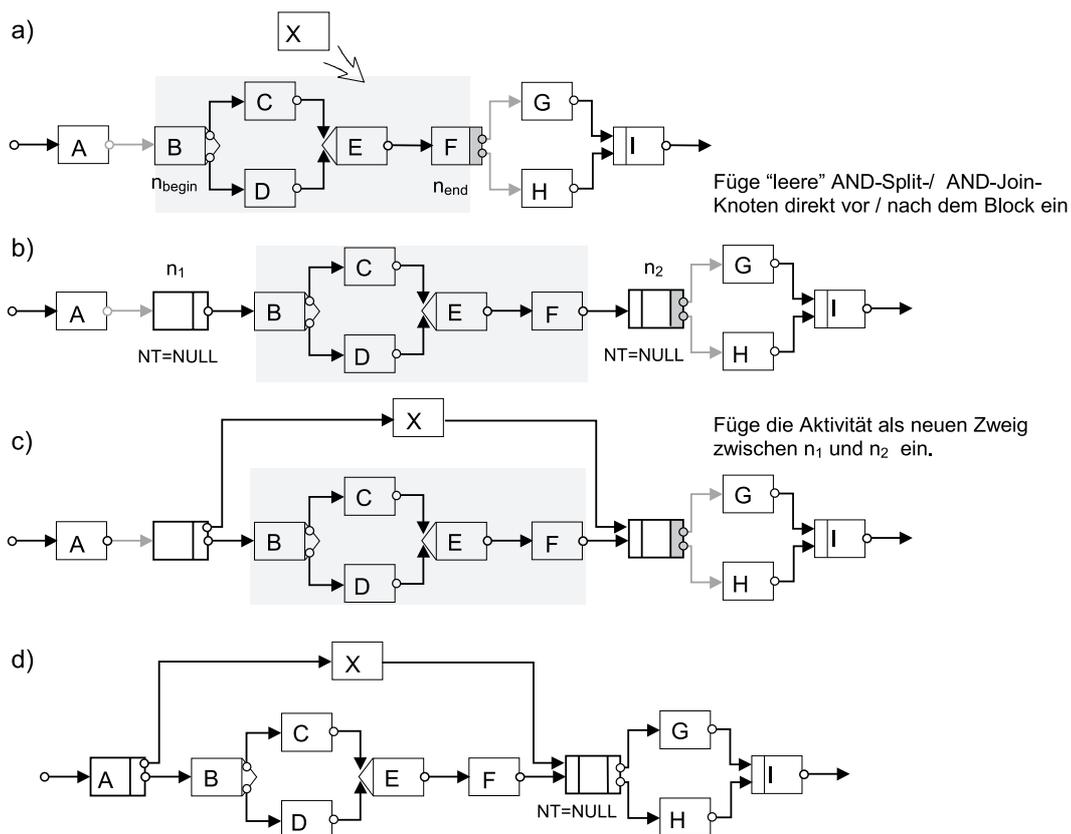


Abb. 4.16: Beispiel für paralleles Einfügen (aus [Reic00b])

Auf die formelle Beschreibung und die Restriktionen des *Parallelen Einfügens* ist hier bewusst verzichtet worden. Sie sind in [Reic00b] nachzulesen.

Einfügen neuer Teilzweige

Soll bei einer bestehenden Verzweigung ein neuer Teilzweig eingefügt werden, ist dieses mit dem bisher vorgestellten seriellen und parallelen Einfügen nicht durchführbar.

Der Unterschied zum seriellen und parallelen Einfügen liegt darin, dass bei der Platzierung der neuen Aktivität der Split- und der zugehörige Join-Knoten angegeben wird. Zusätzlich müssen auch zugehörige Bedingungen angegeben werden.

Auf eine genauere Beschreibung und die zu beachtenden Restriktionen soll hier nicht weiter

eingegangen werden. Sie finden sich in [Reic00b].

Bisher wurde das Einfügen von Aktivitäten in die Vorlage einer Prozessinstanz betrachtet. Die beschriebenen Operationen sind aber genauso auf eine laufende Prozessinstanz anzuwenden. Die dabei zu berücksichtigenden Restriktionen, wie die Berechtigung zu derartigen Änderungen, werden in Abschnitt 7.3 beschrieben.

Löschen

Soll eine Aktivität aus dem Ausführungsgraphen einer Prozessinstanz entfernt werden, müssen sowohl die Aktivität und ihre Kanten aus dem Graphen entfernt werden als auch sämtliche von ihr ausgehenden Lese- und Schreibzugriffe im Datenkontext. Auch hier ist das oberste Ziel wieder der Erhalt der Korrektheit und der Konsistenz des Prozesses und seiner Daten.

In ADEPT_{flex} ist es möglich, normale Aktivitäten, Nullaktivitäten und ganze Kontrollblöcke aus dem Ausführungsgraphen zu entfernen. Es ist nicht möglich, die Start- bzw. Endknoten eines Workflows zu löschen, da diese Bestandteil eines jeden Prozessinstanzgraphen sind. Soll eine Aktivität oder ein Kontrollblock aus dem Ausführungsgraphen gelöscht werden, muss zuvor überprüft werden, ob der Löschvorgang überhaupt zu einem zulässigen Resultat führt. Letzteres ist hier vorausgesetzt.

In Abb. 4.17 sind Beispiele für das Löschen von Aktivitäten skizziert. Ist der zu löschende Knoten X keine Split- oder Join-Aktivität und besitzt er auch sonst keine ein- oder ausgehenden Spezialkanten (Fehlerkanten, Synchronisationskanten usw.), kann die Aktivität X und ihre zugehörigen Kontrollkanten aus dem Ausführungsgraphen entfernt werden, anschließend muss allerdings eine neue Kante zwischen der Vorgänger- und Nachfolgeraktivität von X eingefügt werden (Abb. 4.17 a). Ist X der einzige Knoten eines Teilzweiges einer Parallelverzweigung, kann das Einfügen der neuen Kontrollkante entfallen, der Zweig wird vollständig aus dem Graphen entfernt. Besitzt der neu entstandene Ausführungsgraph an dieser Stelle dann nur noch einen Teilzweig, kann der Graph weiter vereinfacht und die Aus- und Eingangssemantik ihres Split- bzw. Join-Knotens abgeändert werden. (Abb. 4.17 b). Handelt es sich bei dem zu löschenden Knoten um einen Split- oder Join-Knoten mit ein- oder ausgehenden Kanten, muss anstelle der Aktivität eine Nullaktivität eingebaut werden (Abb. 4.17 c).

Auf die formelle Beschreibung der Löschoperation soll hier nicht weiter eingegangen werden, sie findet sich in [Reic00b].

Zusammenfassung weiterer Änderungsoperationen

Neben den oben beschriebenen Einfüge- und Löschooperationen, gibt es noch Operationen für das Verschieben von Aktivitäten und Kontrollblöcken, das Hinzufügen und Löschen von Spezialkanten, die Bildung und Auflösung physischer Blockstrukturen, die Änderung von DF-Modellen und die Änderung von Knoten- und Kantenattributen. Diese sollen im Folgenden teilweise kurz und informell beschrieben werden. Die formelle Beschreibung findet sich in [Reic00b].

Beim *Verschieben von Aktivitäten bzw. Kontrollblöcken* werden die betreffenden Teile aus ihrem bisherigen Umfeld im Ausführungsgraphen herausgelöst und an der gewünschten Stelle neu eingefügt. Dabei werden der Knoten und seine Kanten nicht aus dem Graphen gelöscht, sondern die Kontextkanten des Knotens werden „umgehängt“.

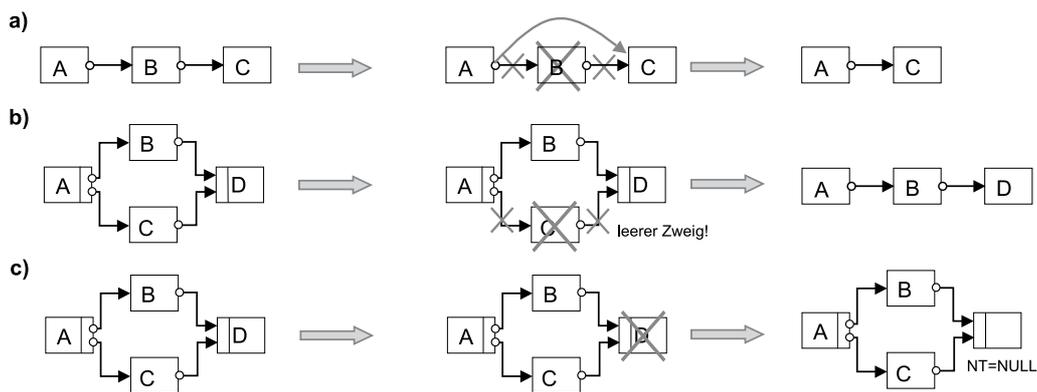


Abb. 4.17: Beispiele für das Löschen von Aktivitäten aus dem Ausführungsgraphen einer Prozessinstanz (aus [Reic00b])

Ein Beispiel für das Verschieben einer Aktivität ist in Abb. 4.18 skizziert. Dabei wird Aktivität B so verschoben, dass sie sowohl zu einem früheren als auch zum ursprünglich modellierten Zeitpunkt in der Prozessinstanz ausgeführt werden kann. Bei einer derartigen Änderungsoperation muss der Ausführungsstatus der Instanz berücksichtigt werden.

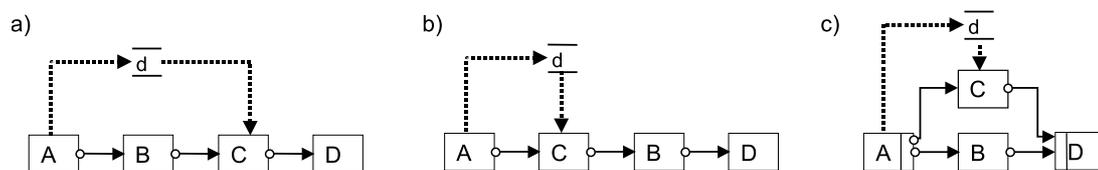


Abb. 4.18: Beispiel für das Verschieben einer Aktivität im Ausführungsgraphen einer Prozessinstanz (aus [Reic00b])

Da ein- und ausgehende Spezialkanten einer Aktivität beim Verschieben derselben nicht gelöscht werden, bleiben sie erhalten. Eine Aktivität mit Spezialkanten darf daher nicht in einen Bereich des Graphen verschoben werden, der vor bzw. nach dem Zielknoten der Kante liegt. Ansonsten würden unerwünschte Zyklen entstehen. Ein Beispiel für das Verschieben einer Aktivität mit Synchronisationskante ist in Abb. 4.19 gegeben. Hier darf die Aktivität G nicht in einen Bereich verschoben werden, der dem Knoten D nachfolgt.

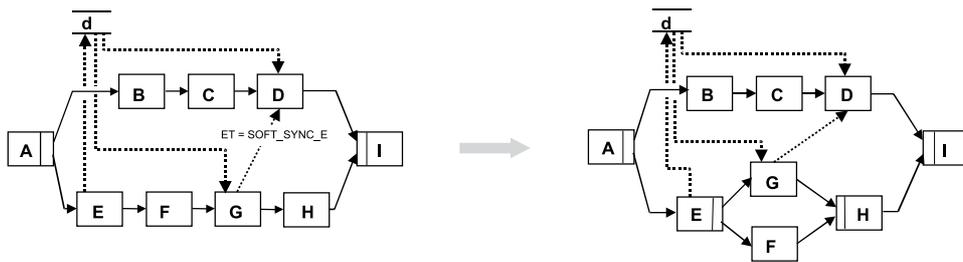


Abb. 4.19: Beispiel für das Verschieben einer Aktivität mit ausgehender Synchronisationskante (aus [Reic00b])

Kapitel 5

Modellierungsrechte

Im Zusammenhang mit Prozess-Management-Systemen ist es notwendig, verschiedene Berechtigungen zu definieren, damit die Verwendung des Systems genau geregelt werden kann. Dazu gehören u.a. die Modellierungsrechte. Bei den Modellierungsrechten muss unterschieden werden zwischen der Organisationsmodellierung, der Prozessmodellierung und der Modellierung der verschiedenen Aktivitätenvorlagen aus denen sich die unterschiedlichen Prozessvorlagen zusammensetzen. Eine andere Bezeichnung für Vorlagen ist Modell. Diese Namen werden im Folgenden synonym verwendet. Neben der Erstellung und Freigabe von verschiedenen Modellen bzw. Modellausschnitten müssen sowohl Zugriffsberechtigungen erstellt werden können als auch Anpassungsberechtigungen zum Ändern von Prozessabläufen. Allerdings soll in diesem Kapitel weder auf die Zugriffsberechtigungen zum Lesen von Daten noch auf die Änderungsberechtigungen eingegangen werden. Die Zugriffsberechtigungen werden gesondert in Kap. 6 behandelt, die Anpassungsberechtigungen in Kap. 7. Lediglich auf die Beschreibung der Modellierungsrechte soll im Folgenden eingegangen werden.

5.1 Organisationsmodellierung

Bevor mit einem PMS sinnvoll gearbeitet werden kann, muss die Unternehmens- bzw. die Personalstruktur modelliert werden. Dabei müssen verschiedene Modellierungsschritte berücksichtigt werden. Die wichtigsten sind:

- Anlegen von Benutzern
- Anlegen von Organisationsmodellen
- Freigeben von Organisationsmodellen
- Ändern von Organisationsmodellen
- Anfragen an das Organisationsmodell, z. B. alle Rollen eines Benutzers

Im Folgenden sollen diese Funktionen genauer beschrieben werden sowie deren berechtigte Benutzer.

5.1.1 Anlegen von Benutzern

Alle Mitarbeiter, die zu einem späteren Zeitpunkt durch das PMS bei ihrer Arbeit unterstützt werden sollen, müssen diesem im vornherein bekannt gemacht werden. Dazu werden

alle Mitarbeiter namentlich und mit ihren Fähigkeiten erfasst. Um nicht Arbeiten doppelt ausführen zu müssen, wäre es an dieser Stelle sinnvoll, wenn bereits existierende Datenbanken mit den Personaldaten, z. B. aus der Buchhaltung, übernommen werden können. Auf diese Weise können sich keine Flüchtigkeitsfehler, z. B. bei der Namensschreibung, einschleichen. Fehlende Eintragungen könnten dann einfach hinzugefügt werden.

Das Anlegen der Benutzer sollte von einer ausgewählten Person oder Gruppe von Personen übernommen werden, damit zum einen der Überblick über die Benutzer gewahrt bleibt und nicht jede beliebige Person Einblick in die Daten einer anderen Person erlangen und diese beliebig verändern kann. Außerdem dürfen nicht von jeder Person beliebige Benutzer angelegt werden, da evtl. Lizenzbeschränkungen ebenfalls eingehalten werden müssen. Als berechtigte Person oder Personengruppe ist sicherlich der Systemadministrator geeignet.

5.1.2 Anlegen von Organisationsmodellen

Das Anlegen von Organisationsmodellen ist sicherlich einer der Grundsteine eines jeden PMS. Hier wird sowohl die Aufbaustruktur eines Unternehmens mit all seinen Abteilungen und deren Beziehungen untereinander abgebildet, als auch die Personalstruktur. Auch werden auf diese Art Vertreterregelungen und Leitungspositionen erfasst.

Die Modellierung der Organisationsstruktur erfolgt nach dem zu einem PMS gehörenden Organisations-Metamodell. Das Organisationsmodell dient, bei den in Kap. 4.2.3 beschriebenen Bearbeiterformeln, als Auflösungsgrundlage der Berechtigungen.

Das Organisationsmodell soll und darf nicht von jeder beliebigen Person geändert werden. Es darf lediglich geändert werden, wenn auch im realen Unternehmen Änderungen vorstatten gehen, wie z. B. das Ausscheiden eines Mitarbeiters aus dem Unternehmen oder der Aufstieg eines Mitarbeiters im Unternehmen. Die Modellierung der Organisationsstruktur im Organisationsmodell darf nur von sog. Systemadministratoren vorgenommen werden. Dadurch ist gewährleistet, dass nur die Personen die Organisationsstruktur modellieren können, die einen besseren Überblick über die Firma besitzen und nicht direkt in die verschiedenen Prozesse involviert sind, deren Abwicklung später durch das PMS unterstützt werden sollen.

5.1.3 Freigeben von Organisationsmodellen

Nachdem das Organisationsmodell angelegt wurde, darf es nicht direkt verwendet werden, da noch Fehler enthalten sein könnten. Der leitende Systemadministrator muss das erstellte Organisationsmodell überprüfen und anschließend, sofern keine Beanstandungen vorliegen, zur Verwendung freigeben.

5.1.4 Ändern von Organisationsmodellen

Im Laufe des Lebenszyklus eines Unternehmens ist es ganz natürlich, dass das Unternehmen gewissen Schwankungen unterliegt. So scheiden z. B. Mitarbeiter aus einem Unternehmen aus, neue Mitarbeiter treten ein oder die Position eines Mitarbeiters innerhalb des Unternehmens ändert sich beispielsweise durch eine Beförderung. In all diesen Fällen muss auf die Änderungen in der Organisations- bzw. Personalstruktur eingegangen werden. Mitarbeiter, die aus dem Unternehmen ausscheiden, müssen aus dem Organisationsmodell entfernt und neue Mitarbeiter hinzugefügt werden. Ändert sich die Position eines Mitarbeiters, muss sich dieses in den Rollenbesetzungen ebenfalls niederschlagen. Auch können in einem Unternehmen neue Abteilungen entstehen oder andere Abteilungen geschlossen werden.

Diese Änderungen der Organisationsstruktur müssen ebenfalls im Organisationsmodell berücksichtigt werden. Dadurch wird ermöglicht, dass die in Kap. 4.2.3 beschriebenen Bearbeiterformeln, auch weiterhin korrekt aufgelöst werden können, und die zugehörigen Aufgaben nur solchen Mitarbeitern zugeteilt werden, die auch wirklich existieren und zur Bearbeitung berechtigt sind.

Wie bei den zuvor diskutierten Aspekten der Organisationsmodellierung, sollte auch das Ändern von Organisationsmodellen nicht durch beliebige Personen durchgeführt werden. Sinnvoll ist auch hier die Beschränkung der Änderung von Organisationsmodellen auf den Systemadministrator bzw. eine Gruppe von Systemadministratoren.

5.1.5 Anfragen an das Organisationsmodell

Im Zuge von wechselnden Personalstrukturen, ist es manchmal sinnvoll, sich einen Überblick über existierende Rollen und Mitarbeiter zu verschaffen. Beispielsweise kann es im Zusammenhang mit einer Beförderung eines Mitarbeiters von Interesse sein, vor oder nach der Änderung der Rollenzugehörigkeiten, die aktuellen Zugehörigkeiten vom System abzufragen. Damit wird dem Erhaltenbleiben falscher Zugehörigkeiten entgegengewirkt, da direkt der Status eines Mitarbeiters eingesehen werden kann.

Auch in diesem Falle gilt, dass nicht jeder beliebigen Person Einblick in den Status eines Mitarbeiters gewährt werden darf. Dieser Einblick sollte nur der Gruppe von Systemadministratoren vorbehalten sein.

5.2 Aktivitätenmodellierung

Unter Aktivitätenmodellierung wird die Beschreibung aller einzelnen Tätigkeiten verstanden. Dazu gehört auch die Modellierung der benötigten Datenzugriffe. Aktivitätenvorlagen dienen als Vorlage der Aktivitäteninstanzen auf der Ausführungsebene. Mehrere Aktivitäten bilden die Grundlage der abzubildenden Prozesse, sie werden jedoch erst in Kap. 5.3 beschrieben. Die wichtigsten Funktionen im Zusammenhang mit der Modellierung der Aktivitäten sind:

- Aktivitätenvorlagen erstellen
- Aktivitätenvorlagen freigeben
- Aktivitätenvorlagen ändern
- Aktivitätenvorlagen löschen

Auf die Beschreibung dieser Funktionen und deren Berechtigungen soll im folgenden eingegangen werden.

5.2.1 Aktivitätenvorlagen erstellen

Bei der Erstellung von Aktivitätenvorlagen ist es wichtig verschiedene Dinge festzulegen. Dazu gehören:

- Name der Aktivität zur eindeutigen Identifikation
- Beschreibung der Funktionalität der Aktivität

- Bearbeiterformel zur Festlegung der zur Ausführung dieser Aktivität berechtigten Person(en)
- Ein- und Ausgabedaten zur Versorgung der Aktivität mit den zur Bearbeitung benötigten Daten
- Zuordnung der Aktivität zu einer Aktivitätenkategorie (s. Kap. 7.3.1.3)
- eventuelle Verknüpfung der Aktivität mit einer Anwendungskomponente

Das Erstellen neuer Aktivitätenvorlagen ist also mit einer Menge von Definitionen verbunden. Darin liegt begründet, dass nur die Personen eine neue Aktivitätenvorlage erstellen dürfen, die einen entsprechenden Überblick sowohl über die Organisation als auch über die Möglichkeiten des Unternehmens besitzen. Aus diesem Grund sollten nur wenige Personen zur Erstellung von Aktivitätenvorlagen berechtigt sein. Sinnvoll wäre einen (oder mehrere) spezielle Prozess- oder Aktivitätenmodellierer dazu einzustellen. Dieser würde im Gespräch mit den Angestellten verschiedene Aktivitäten erarbeiten und diese dann in elektronische Form umsetzen. Steht kein Aktivitätenmodellierer zur Verfügung sollte der Systemadministrator diese Aufgabe übernehmen.

5.2.2 Aktivitätenvorlagen freigeben

Nachdem Aktivitätenvorlagen erstellt wurden, sind sie nicht automatisch einsatzbereit. Sie müssen erst von einer berechtigten Person freigegeben werden. Sicher ist es hier sinnvoll, eine leitende Person mit einzubeziehen, die bei der Vorbereitung der Modellierung beteiligt war. Ist sie mit der Modellierung der Aktivitätenvorlage einverstanden, kann sie auf der elektronischen Seite freigegeben wird. Dies sollte wiederum durch den Aktivitätenmodellierer geschehen, existiert er nicht, sollte der Systemadministrator die Aktivitätenvorlage freigeben.

5.2.3 Aktivitätenvorlagen ändern

Im Laufe des Lebenszyklus einer Aktivitätenvorlage können Veränderungen auftreten, z. B. derart, dass eine Anwendungskomponente durch eine andere ersetzt wird. In diesem Fall, kann sich der Aufruf der Anwendungskomponente mit den zugehörigen Eingabedaten verändern. Diese Änderung muss dann in der Aktivitätenvorlage vorgenommen werden. Da i.d.R. die Systemadministratoren über die Verwendung neuer Programme entscheiden, bzw. für deren Inbetriebnahme zuständig sind, sollten die entsprechenden Anpassungen in der Aktivitätenvorlage auch von ihnen bzw. den Aktivitätenmodellierern durchgeführt werden.

5.2.4 Aktivitätenvorlagen löschen

Wird eine Aktivität bei der Bearbeitung von Prozessinstanzen dauerhaft nicht mehr berücksichtigt, ist es sinnvoll, diese aus dem Vorlagenrepository zu entfernen. Dies bewahrt die Übersicht über das Repository, da nicht mehr Vorlagen gespeichert werden als unbedingt nötig. Die Vorlage darf allerdings nur aus dem Repository entfernt werden, wenn deren Existenz in keinerlei Kontext mehr benötigt wird. Um dies zu überprüfen, ist ein guter Überblick über alle vorhandenen Aktivitäten und Prozesse unumgänglich. Daher sollten Aktivitätenvorlagen nur von wenigen berechtigten Benutzer aus dem System entfernt werden können. Sinnvoll ist hier sicherlich der Aktivitätenmodellierer, sofern er vorhanden ist oder aber der Systemadministrator, da von ihm ebenfalls eine gute Kenntnis des Systems und seiner Prozesse vorausgesetzt werden kann.

5.3 Prozessmodellierung

Nach der Modellierung der Organisationsstruktur (Kap. 5.1) und der Aktivitätsvorlagen (s. Kap. 5.2) ist die Prozessmodellierung ein weiterer wichtiger Grundstein der Unterstützung von Arbeitsabläufen durch ein PMS. Prozessvorlagen ergeben sich aus der Kombination mehrerer Aktivitätsvorlagen. Mithilfe der Prozessmodellierung werden die zu unterstützenden Arbeitsabläufe im PMS dargestellt, die als Vorlage der Prozessinstanzen auf der Ausführungsebene dienen. Bei der Prozessmodellierung sind verschiedene Funktionen zu unterscheiden. Die wesentlichen sind:

- Anlegen von Prozessmodellen
- Freigeben von Prozessmodellen
- Anfragen zu Prozessmodellen
- Ändern von Prozessmodellen mit/ohne Propagation der Änderung auf laufende Instanzen
- Löschen von Prozessmodellen

Im Folgenden sollen sowohl diese Tätigkeiten kurz erläutert werden als auch deren berechnigte Benutzer.

5.3.1 Anlegen von Prozessmodellen

Bevor ein Prozessmodell als Vorlage für Prozessinstanzen auf Ausführungsebene dienen kann, muss es als Prozessvorlage modelliert werden. D. h. die verschiedenen benötigten Aktivitätsvorlagen aus dem Repository werden so miteinander kombiniert, dass sich ein logischer Ablauf, z. B. der Ablauf der radiologischen Untersuchung aus Kap. 3.1, ergibt. Die einzelnen Tätigkeiten der Prozessvorlage werden zur Ausführungszeit den jeweiligen berechtigten Personen zur Bearbeitung angeboten. Da nicht jede Person den Überblick und die Einsicht in die verschiedenen Aktivitätsvorlagen besitzt, sollten nur ausgewählte Personen dazu berechnigt sein, neue Prozessvorlagen anzulegen. Sinnvoll wäre ein Prozessmodellierer, gibt es ihn in einem Unternehmen nicht, sollten nur Systemadministratoren berechnigt sein, neue Prozessmodelle zu definieren. Allerdings sollte der eigentliche Ablauf zuvor mit Personen diskutiert werden, die auch an der Ausführung der Prozessinstanz beteiligt wären, so dass sie nicht unnötig durch das PMS eingeschränkt und optimal durch dieses unterstützt werden können.

5.3.2 Freigeben von Prozessmodellen

Nachdem Prozessvorlagen modelliert wurden, stehen sie nicht automatisch direkt zur Verfügung. Zum einen sollten sie noch einmal von einer Person überprüft werden, die später diese Prozessinstanz bearbeiten soll. Stehen keine Beanstandungen mehr an, wird sie sowohl von einer dieser Personen als auch vom Prozessmodellierer freigegeben und kann in Zukunft bearbeitet werden. Steht kein Prozessmodellierer zur Verfügung, sollte diese Aufgabe vom Systemadministrator übernommen werden.

5.3.3 Anfragen zu Prozessmodellen

Um sich einen Überblick über eine bestehende Prozessvorlage machen zu können, ist es sinnvoll, sich die Vorlage im Ganzen anzeigen zu lassen. Anschließend kann dann über eine Modifikation oder das Entfernen dieser Prozessvorlage genauer nachgedacht werden. Nicht jede Person eines Unternehmens sollte jedoch dazu berechtigt sein, sich komplette Prozessvorlagen anzusehen, da Prozesse oft bereichsübergreifend ablaufen. Es soll dann nicht von Interesse sein, was in anderen Bereichen innerhalb dieses Prozesses durchgeführt wird. Berechtigte Benutzer, die sich eine komplette Prozessvorlage ansehen dürfen, sollten daher nur Prozessmodellierer sein, wenn sie nicht existieren, sollten nur Systemadministratoren zum Einblick in Prozessvorlagen berechtigt sein.

5.3.4 Ändern von Prozessmodellen

Im Laufe des Lebenszyklus einer Prozessvorlage kann es durchaus zu Veränderungen kommen, wenn z. B. Teile eines Fertigungsprozesses nicht mehr im Unternehmen selbst gefertigt sondern komplett eingekauft werden. In einem derartigen Fall muss die entsprechende Fertigungsaktivität aus der Prozessvorlage entfernt werden. Hierbei gibt es zwei unterschiedliche Umstände zu berücksichtigen. Soll eine Änderung sofort in laufenden Prozessinstanzen dieser Prozessvorlage berücksichtigt werden, muss die Änderung auf die laufende Instanz propagiert werden. Soll sie jedoch erst in zukünftigen Prozessinstanzen berücksichtigt werden, muss die Änderung nicht propagiert werden. Im letzteren Fall muss bei einem neuen Start der Prozessinstanz dieser Prozessvorlage die Änderung berücksichtigt werden. Im ersteren Fall muss der Status der Prozessinstanzen dieser Prozessvorlage untersucht werden. Ist die Bearbeitung einer Instanz bereits über den geänderten Punkt hinausgeschritten, darf die Änderung nicht mehr in diese Prozessinstanz eingefügt werden. Ist der geänderte Bereich in der Prozessinstanz noch nicht erreicht, kann die Prozessinstanz ohne Probleme geändert werden. Ist die Bearbeitung jedoch gerade in den Bereich gelangt, der von der Änderung berührt wird, muss genau geprüft werden, in wieweit die Änderung noch durchgeführt werden kann. Ist die erste Aktivität des geänderten Bereichs bereits gestartet, darf die Änderung nicht mehr berücksichtigt werden. Ist die erste Aktivität lediglich selektiert, darf die Änderung noch eingebracht werden.

Auf die Berechtigungen zum Ändern von Prozessmodellen wird hier jedoch nicht näher eingegangen, Kap. 7 befasst sich ausschließlich mit Anpassungs- und Änderungsrechten.

5.3.5 Löschen von Prozessmodellen

Wird eine Prozessvorlage in keinem Bereich eines Unternehmens mehr benötigt, kann es durchaus sinnvoll sein, diese aus dem Prozessvorlagenrepository zu entfernen. Dadurch wird die Übersicht über alle existierenden Prozessvorlagen gewahrt.

Eine Prozessvorlage darf jedoch nicht leichtfertig aus dem Repository entfernt werden. Zuvor muss kritisch überprüft werden, ob diese wirklich nicht weiter benötigt wird. Durch diese kritische Überprüfung ist begründet, dass nicht jeder Benutzer Prozessvorlagen aus dem PMS entfernen darf. Das Löschen von Prozessvorlagen sollte nur dem Prozessmodellierer bzw. dem Systemadministrator erlaubt sein, wobei dieser nur nach Rücksprache mit den betroffenen Bereichen Vorlagen aus dem Repository entfernen darf.

Kapitel 6

Ausführungs- und Zugriffsrechte für Standardabläufe

Bereits für die normale Prozessausführung müssen Ausführungs- und Zugriffsrechte vorliegen, die z. B. regeln, wer welche Schritte eines Prozesses potentiell ausführen darf oder wer auf welche Daten einer Prozessinstanz zugreifen können soll. In diesem Abschnitt werden damit zusammenhängende Problemstellungen diskutiert und Lösungsansätze erarbeitet. Sie basieren auf dem in Kap. 4 vorgestellten Bearbeiterformeln. Keine Berücksichtigung finden hier allerdings dynamische Änderungen und dazu erforderliche Anpassungsrechte. Sie werden in Kap. 7 behandelt.

6.1 Prozessbeteiligte Personen

Auf der Ausführungsebene gibt es für jede Prozessinstanz bestimmte Personen, die für ihre Abarbeitung zuständig sind. Diese werden, je nach Prozess-Management-System, bereits bei der Modellierung festgelegt oder sie ergeben sich während der Laufzeit des Prozesses. Prozessbeteiligte müssen danach unterschieden werden, ob sie aktiv oder passiv an einem Prozess beteiligt sind, d. h. ob sie direkt in die Bearbeitung der Aktivitäten beteiligt sind oder nicht. Desweiteren muss festgelegt sein, welche Person(en) prozessverantwortlich sind. Dies wird z. B. für eine bestimmte Instanz notwendig, um in Ausnahmesituationen Mitteilungen bearbeiten zu können.

6.1.1 Aktive Prozessbeteiligte

Aktiv an einem Prozess beteiligt sind alle Personen, die in der Vergangenheit an der Bearbeitung von Aktivitäten eines Prozesses beteiligt waren, aktuell daran beteiligt sind oder in der Zukunft beteiligt sein werden.

Bearbeiter, die bisher aktiv an der Ausführung eines bestimmten Prozesses beteiligt waren, können aus seiner Ausführungshistorie und dem Workflow-Status ermittelt werden. Sie enthält u. a. Informationen

- zum Erzeuger und Starter der Prozessinstanz
- zum Starter gerade laufender Aktivitäten
- zu den Bearbeitern bereits abgeschlossener Aktivitäten.

Hierbei müssen der Starter und der Bearbeiter der ersten Aktivität einer Prozessinstanz nicht unbedingt verschiedene Personen sein.

Bearbeiter, die aktuell an der Ausführung eines Prozesses beteiligt sind, können aus der Ausführungshistorie und dem Workflow-Status bestimmt werden. Darunter fallen Personen, die eine aktivierte Aktivität in ihrer Arbeitsliste zur Bearbeitung angeboten bekommen oder die eine solche Aktivität bereits selektiert haben und sie evtl. schon bearbeiten. Darunter fallen alle Aktivitäten mit dem Status aktiviert, selektiert, gestartet oder suspendiert.

Bei der Betrachtung zukünftiger Prozessbeteiligter muss unterschieden werden, ob diese sicher oder nur potentiell an der Prozessausführung beteiligt sein werden. Eine Person wird als sicherer Bearbeiter angesehen, wenn bereits vor der Aktivierung einer Aktivität feststeht, wer diese zu bearbeiten hat. Zukünftige Bearbeiter werden anhand der zu einer Aktivität gehörenden Bearbeiterformel, der Prozessbeschreibung und der Ausführungshistorie bestimmt. Die zukünftigen Bearbeiter werden wie folgt festgelegt:

- Bestimmung eines Bearbeiters über eine abhängige Bearbeiterzuordnung (z. B. gleicher Bearbeiter wie der Bearbeiter einer bereits abgeschlossenen Aktivität)
- ein abgeschlossener Vorgängerschritt bestimmt den zukünftigen Bearbeiter
- im weiteren Verlauf des Workflows wird der Bearbeiter der Bearbeitermenge angehören (siehe Abb. 6.1 AND-Split).

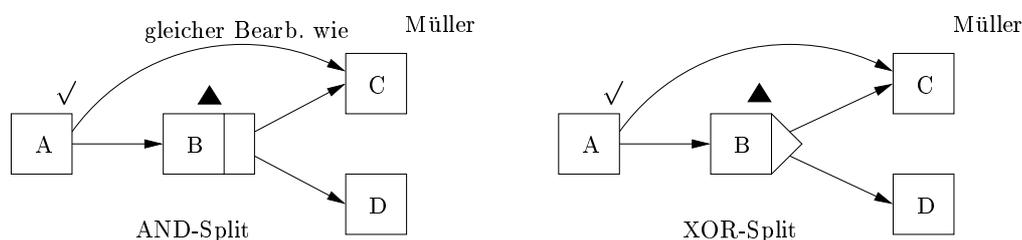


Abb. 6.1: Bestimmung zukünftiger Bearbeiter

In Abb. 6.1 sind zwei Beispiele für die Bestimmung zukünftiger Bearbeiter gegeben. Im ersten Teil der Abbildung (AND-Split) bestimmt Aktivität A, wer die übernächste Aktivität bearbeiten soll. In diesem Fall gehört „Müller“ sicher zu den zukünftigen Bearbeitern, da Aktivität C auf jeden Fall bearbeitet werden wird. Im zweiten Teil der Abbildung (XOR-Split) wird zwar von Aktivität A auch „Müller“ als Bearbeiter der Aktivität C festgelegt, aber in Aktivität B wird erst bestimmt, welche der Aktivitäten C oder D tatsächlich ausgeführt wird. Aktivität C muss also nicht definitiv ausgeführt werden. D. h. „Müller“ wäre in diesem Beispiel nur ein potentieller Bearbeiter.

6.1.2 Passive Prozessbeteiligte

Wie oben bereits erwähnt, gibt es neben aktiven auch passive Prozessbeteiligte. Passive Prozessbeteiligte sind nicht aktiv in einen Prozess involviert, sie werden lediglich von ihren Mitarbeitern oder Untergebenen über den Prozess informiert. Auf einer Station im Krankenhaus, muss sich beispielsweise der Chefarzt über die Patienten informieren können, muss aber die Behandlung nicht unbedingt selbst durchführen.

6.1.3 Prozessverantwortliche

Kann die Bearbeiterzuordnung einer Aktivität nicht aufgelöst werden, ist es nötig einen „Defaultbearbeiter“ zu haben, an den die Aktivität weitergeleitet wird, so dass er diese delegieren kann. Das gleiche gilt für den Fall, dass ein Prozess abgebrochen wurde. In diesem Fall muss der Prozess oder eine Prozessinstanz aus dem Prozess-Management-System wieder entfernt bzw. „sauber“ beendet werden können. Dieses lässt sich leicht umsetzen, wenn in diesen Fällen der Prozess an den Prozessverantwortlichen gegeben wird. Für den Prozessverantwortlichen gibt es verschiedene Formen der Festlegung. Ist die Bearbeitung eines Prozesses nur an eine Organisationseinheit gebunden, ist der Prozessverantwortliche leicht bestimmbar, z. B. der Leiter der Organisationseinheit. Durchläuft der Prozess allerdings mehrere Organisationseinheiten, ist der Prozessverantwortliche schwieriger zu bestimmen. Hier liegt begründet, warum der Prozessverantwortliche sowohl eine einzelne Person als auch eine Gruppe von Personen sein kann.

Ist wie bei MQSeries Workflow eine konkrete Person als Prozessverantwortlicher in einem gegebenen Prozessmodell festgelegt, ist dies nicht immer günstig. Treten z. B. Änderungen im Organisationsmodell auf (die betreffende Person scheidet aus dem Unternehmen aus), müssen auch in den Prozessvorlagen entsprechende Anpassungen vorgenommen werden. Dadurch entsteht ein hoher Wartungs- und Pflegeaufwand des Systems.

Der Prozessverantwortliche kann sowohl statisch als auch dynamisch über Bearbeiterzuordnungsausdrücke festgelegt werden. Ein Nachteil der statischen Festlegung des Bearbeiterzuordnungsausdrucks für den Prozessverantwortlichen, der fest mit einem Prozess verbunden ist, ist unabhängig davon, in welchem Zustand sich der Prozess befindet und evtl. eine andere Person an dieser Stelle besser geeignet wäre. Als Beispiel wäre ein Prozess zu nennen, der mehrere Bereiche durchläuft und die ganze Zeit ein und denselben Prozessverantwortlichen hat, dieser aber nur in einem Bereich geeignet, in den anderen eher ungeeignet ist. Wird hingegen für eine gegebene Prozessvorlage ein dynamischer Bearbeiterzuordnungsausdruck festgelegt, aus dem sich der oder die Prozessverantwortliche/n ergeben, ist die Bestimmung des Prozessverantwortlichen wesentlich flexibler und erhöht die Ausdrucksfähigkeit in der Definition des Prozessverantwortlichen. Eine weitere Möglichkeit prozessverantwortliche Personen zu bestimmen, ist die Festlegung eines Prozessverantwortlichen für jede einzelne Aktivität eines Prozesses und jeden ihrer Zustände über Bearbeiterzuordnungsausdrücke. Als Bearbeiterzuordnungsausdruck sind bei dieser Art der Bestimmung des Prozessverantwortlichen auch Vereinigungen über Mengen von Prozessverantwortlichen darstellbar, so dass einzelne über Differenzbildung aus der Menge der Prozessverantwortlichen leicht wieder zu entfernen sind. So wird im Bearbeiterzuordnungsausdruck eine Menge von Personen bestimmt, die im aktuellen Zustand eines Prozesses verantwortlich sind. Auf diese Art kann der Prozessverantwortliche spezieller bestimmt werden und auch die Problematik der Zuständigkeit innerhalb eines Prozesses, der durch verschiedene Bereiche läuft, ist beseitigt.

6.2 Ausführungsrechte

Mithilfe der Ausführungsrechte wird bestimmt, wer das Recht hat, Nicht-Standard-Operationen im Prozess-Management-System auszuführen, da es schon aus juristischen Gründen nicht vertretbar ist, wenn jeder alles tun darf, z. B. darf eine Pflegekraft keine Untersuchung durchführen oder einen Befund schreiben. Ein Prozess würde dadurch nicht übersichtlicher und nachvollziehbarer. Daher sind die verschiedenen Operationen nur bestimmten Personen erlaubt. Im Folgenden sollen diese *Ausführungsrechte* beschrieben werden.

6.2.1 Rechte für die Manipulation von Prozessinstanzen

In ADEPT können für Prozessinstanzen die folgenden Operationen durchgeführt werden:

- Erzeugen
- Starten
- Unterbrechen
- Fortsetzen
- Beenden
- Abbrechen
- Archivieren
- Löschen
- Monitoring

Im Folgenden wird die Semantik dieser Operationen informell dargestellt und darauf eingegangen, wie Berechtigungen für die verschiedenen Operationen festgelegt werden können.

6.2.1.1 Prozessinstanzen erzeugen

Prozessinstanzen werden auf der Grundlage von Prozessvorlagen erzeugt. Diese Prozessvorlagen (*PV*) werden je nach ihrer Art in verschiedene Kategorien oder Gruppen hierarchisch eingeteilt. Hier ein kurzes Beispiel:

- Medizinische PV
- Medikations-PV
 - * PV Schmerztherapie
 - * PV Chemotherapie
 - * PV Wärme-Kälte-Therapie
 - * PV Bewegungstherapie
 - * ...
 - Untersuchungs-PV
 - * ...
 - ...

Wie sonst auch, werden für die Erzeugung von Prozessinstanzen Bearbeiterzuordnungsaustrücke festgelegt, die angeben, wer eine Prozessinstanz erzeugen darf. Dabei kann nun auf unterschiedliche Arten vorgegangen werden. Entweder werden die Bearbeiterzuordnungsaustrücke für jede einzelne PV auf der untersten Ebene festgelegt oder auf einer übergeordneten werden die entsprechenden Bearbeiterzuordnungsaustrücke spezifiziert und vererben die Rechte an niedrigere Ebenen weiter. Somit gelten die Benutzerzuordnungsaustrücke auch für die untergeordneten Prozessvorlagen. Werden für eine untergeordnete Prozessvorlage spezifischere Rechte benötigt, kann für diese ein spezieller Bearbeiterzuordnungsaustruck festgelegt werden. Ziel der Festlegung von Bearbeiterzuordnungsaustrücken sollte immer die kompakte Festlegung von Rechten sein, damit diese einfach gewartet und auch geändert werden können. Nachdem eine Prozessinstanz erzeugt wurde, muss natürlich noch festgelegt werden, wer sehen darf, dass es diese Prozessinstanz gibt und diese auch starten darf.

6.2.1.2 Prozessinstanzen starten

Wurde eine Prozessinstanz erzeugt, muss diese auch gestartet werden, damit die Aktivität bei einem Bearbeiter in der Arbeitsliste erscheint und dann abgearbeitet werden kann. Beim Start einer Prozessinstanz muss danach unterschieden werden, ob die Instanz explizit gestartet werden muss oder ob sie direkt nach Abschluss ihrer Erzeugung bereits gestartet wurde. Im letztgenannten Fall muss nichts weiter getan werden, die Instanz wird dem zugehörigen Bearbeiter in seiner Arbeitsliste angeboten. Muss die Instanz wie bei ADEPT noch gestartet werden, kann dieses entweder durch das Workflow-System selbst vorgenommen werden oder es muss eine Person bestimmt werden, die den Vorgang manuell startet. Wird die Instanz vom System gestartet, kann diese direkt nach ihrer Erzeugung gestartet werden, was den expliziten Startvorgang überflüssig macht, oder dem System kann ein Termin gegeben werden, zu dem der Vorgang gestartet werden soll. Z. B. wird im Krankenhaus ein Termin für die Durchführung einer Therapie vereinbart, müssen ein Tag vorher die zugehörigen Medikamente bestellt werden. Dieser Vorgang soll dann ein Tag, bevor der Patient wieder kommt, vom entsprechenden Bearbeiter gestartet werden. Die Aktivität soll aber nicht die ganze Zeit, bis diese Aktivität durchgeführt werden soll, in der Arbeitsliste des Bearbeiters stehen, da sie dann evtl. übersehen werden kann. Durch eine spezielle Kennzeichnung neuer Einträge in der Arbeitsliste wird der Bearbeiter direkt auf die Durchführung dieser Aktivität „gestoßen“. Wird eine Prozessinstanz nicht vom System selbst gestartet, gibt es verschiedene Möglichkeiten von Personen, die die Instanz starten können. Die Person, die die Prozessinstanz erzeugt hat, kann identisch mit dem Starter der Instanz sein. Auch ist der Prozessverantwortliche sicher eine sinnvolle Alternative, da er über die eventuellen Hintergründe der Aktivität Bescheid weiß. Als weitere Möglichkeit können über einen Bearbeiterzuordnungsausdruck beliebige Personen als Starter der Instanz spezifiziert werden.

6.2.1.3 Prozessinstanzen unterbrechen/fortsetzen

Laufende Prozessinstanzen müssen unterbrochen werden können, da es z. B. in Krankenhäusern durchaus dazu kommen kann, dass ein Patient eine Therapie aus verschiedensten Gründen unterbricht. Zu einem späteren Zeitpunkt kann die Therapie aber auch wieder fortgesetzt werden. Dem Workflow-System muss dieses ebenfalls kenntlich gemacht werden. Dies sollte aber nicht durch jeden Benutzer des Systems möglich sein. Hier wäre es durchaus sinnvoll, wenn nur der Prozessverantwortliche einen Vorgang unterbrechen darf. Ist ein Vorgang unterbrochen, dürfen alle laufenden Prozesse noch beendet aber keine neuen Schritte mehr gestartet werden dürfen. Zu einem späteren Zeitpunkt darf wiederum nur der Prozessverantwortliche den Vorgang wieder fortsetzen. Es sollte dann, wenn möglich, an der Stelle fortgesetzt werden, an der der Prozess zuvor unterbrochen wurde.

6.2.1.4 Prozessinstanzen beenden

Wie bereits in 4.2.1 beschrieben, besitzt ein Prozess in ADEPT einen expliziten Start- und Endknoten, d. h. ein Prozess muss in ADEPT auch explizit beendet werden. Dieses kann entweder vom System selbst oder von einem spezifizierten Bearbeiter durchgeführt werden. Wird dem Endknoten eines Prozesses der Bearbeiterzuordnungsausdruck „R = System“ zugewiesen, beendet das System den Prozess sofort, wenn der Endknoten aktiviert wurde. Es können keine zusätzlichen Schritte eingefügt werden. Erhält der Endknoten aber einen Bearbeiterzuordnungsausdruck mit „R \neq System“, muss der Workflow explizit von einem Bearbeiter beendet werden. Solange der Prozess noch nicht manuell beendet wurde, können noch beliebig viele zusätzliche Aktivitäten zum Prozess hinzugefügt werden (natürlich nur

an das Ende, nicht vor eine Aktivität, die bereits den Status *COMPLETED* besitzt). Das hat den Vorteil, das auf Ad-hoc Änderungen im Prozess direkt und ohne großen Aufwand eingegangen werden kann. In Spezialfällen kann natürlich auch dem Prozessverantwortlichen das Beenden eines Prozesses zugewiesen werden. Abb. 6.2 zeigt, die Auswirkungen eines expliziten Endknotens und die Beendigung eines Prozesses durch einen Bearbeiter, nicht aber durch das System selbst.

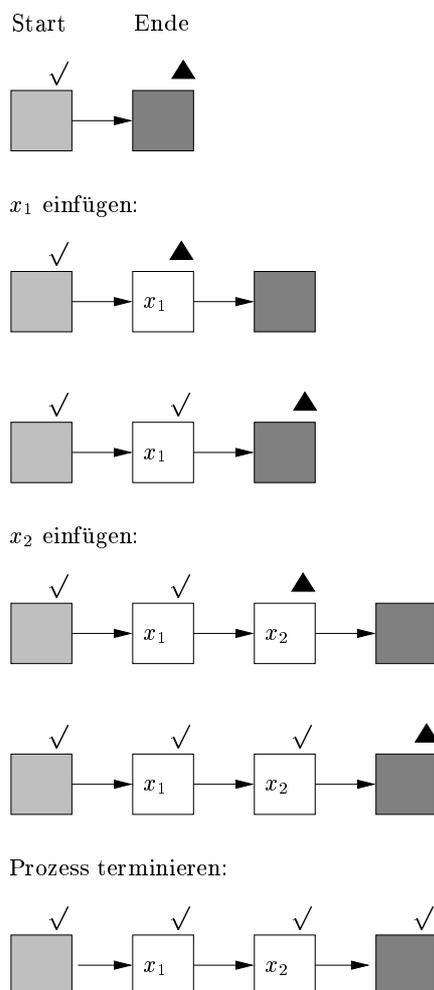


Abb. 6.2: Ad-hoc Änderungen an einem Prozess, der noch nicht beendet ist

6.2.1.5 Prozessinstanzen abbrechen

Es ist notwendig, Prozessinstanzen abbrechen zu können, da die Therapie eines Patienten abgebrochen werden kann, z. B. weil er sie nicht verträgt oder gar verstorben ist. Dem Workflow-System muss dieses mitgeteilt und der Prozess dann auch im System abgebrochen werden. Dies sollte aber nicht durch jeden beliebigen Bearbeiter geschehen dürfen, lediglich der Prozessverantwortliche sollte dazu befähigt sein, da er i. d. R. auch den Überblick über den bisherigen Verlauf des Prozesses besitzt. Neue Aktivitäten dürfen auf keinen Fall mehr gestartet werden, bereits gestartete Aktivitäten sollten wenn möglich direkt abgebrochen werden. Während einer Operation kann beispielsweise nicht einfach abgebrochen werden, wenn festgestellt wurde, dass der Schaden nicht operierbar ist, der Patient aber noch lebt. Der

Patient muss dann so wieder hergerichtet werden, dass er so, wie vor Beginn der Operation, weiterleben kann, d. h. er muss wenigstens wieder zugenäht werden.

6.2.1.6 Prozessinstanzen archivieren oder löschen

Wurde eine Prozessinstanz erfolgreich beendet, muss diese zum Zwecke der Aufbewahrung archiviert werden. Wurde eine Prozessinstanz nicht erfolgreich beendet und zwar so, dass keine Aktivitäten ausgeführt wurden, muss die Instanz aus dem System gelöscht werden. Diese beiden Zustände sind am Status der Prozessinstanz ablesbar. Das Löschen oder Archivieren von Prozessinstanzen sollte nicht von den üblichen Bearbeitern oder dem Prozessverantwortlichen durchgeführt werden. Diese Schritte sollten vom Systemadministrator oder einem Archivar durchgeführt werden.

6.2.1.7 Prozessinstanzen monitoren

Mit Monitoring bezeichnet man das Verfolgen eines Prozessverlaufes. D. h. der Zugriffsberechtigte erhält Einblick in den bisherigen Verlauf des Prozesses, wer hat was warum wann wie lange gemacht. Auf diese Berechtigung wird im Abschnitt 6.3 genauer eingegangen.

6.2.2 Ausführung von Aktivitäteninstanzen

Wie in [Reic00b] beschrieben, ist für die verschiedenen Kontrollkonstrukte des ADEPT-Basismodells eine präzise Definition der Ausführungssemantik unbedingt erforderlich. Sie legt fest, unter welchen Bedingungen eine Prozessinstanz aktiviert werden kann und welche Zustandsänderungen sich im Anschluss an ihre Ausführung ergeben. Auf deren informelle Beschreibung soll im Folgenden eingegangen werden.

In Abb. 6.3 sind vereinfacht alle Zustände und Zustandsübergänge, die bei der Ausführung einer elementaren Aktivitäteninstanz auftreten können, dargestellt. Initial ist eine Aktivität in dem Zustand *NOT_ACTIVATED*. In diesem Zustand ist die von der Aktivität festgelegte Aufgabe (noch) nicht bearbeitbar. Sind alle Vorbedingungen erfüllt, wird die Instanz aktiviert und befindet sich im Zustand *ACTIVATED*. Handelt es sich bei der Instanz um eine automatische Aktivität, wird sie direkt gestartet und erhält den Zustand *STARTED*. Handelt es sich bei der Instanz um eine interaktive Aktivität, erscheint diese in den Arbeitslisten aller in Frage kommenden Bearbeiter. Die Instanz bleibt solange im Zustand *ACTIVATED* bis einer der Bearbeiter sie zur Bearbeitung auswählt. Die Aktivität wechselt dann in den Zustand *SELECTED* und wird bei allen anderen Bearbeitern aus der Arbeitsliste entfernt. Solange sich die Aktivität im Zustand *ACTIVATED* oder *SELECTED* befindet, kann sie wieder deselektiert werden und wird wieder in die Arbeitslisten aller in Frage kommenden Bearbeiter eingefügt, sie wechselt dann wieder in den Zustand *ACTIVATED*. Wird eine Aktivität mit dem Zustand *SELECTED* oder *ACTIVATED* vom zugehörigen Bearbeiter gestartet und wechselt in den Zustand *STARTED* und initiiert die mit ihr verknüpften Aktionen. Kann eine interaktive Aktivität nicht innerhalb eines Schrittes fertiggestellt werden, wird sie unterbrochen und wechselt in den Zustand *SUSPENDED*. Wird sie zu einem späteren Zeitpunkt wieder fortgesetzt, wechselt sie zurück in den Zustand *STARTED*. Wurde die Aktivität korrekt beendet, geht sie in den Zustand *COMPLETED* über, wurde sie nicht korrekt beendet, wechselt sie in den Zustand *FAILED*. Hat eine Aktivität den Zustand *COMPLETED* erreicht, wird erst jetzt entschieden, ob die externen Auswirkungen der Aktivität dauerhaft beibehalten werden sollen und die Aktivität den Zustand *COMMITTED* erhält, oder die internen und externen Auswirkungen der Aktivität zurückgesetzt werden sollen. In diesem Fall würde der Zustand der Aktivität in *COMPENSATED* wechseln.

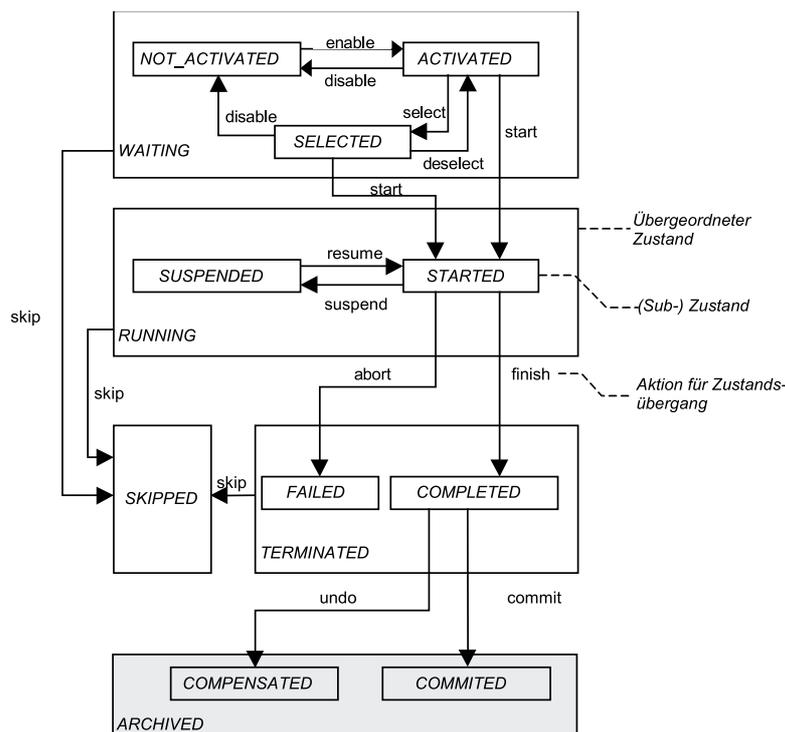


Abb. 6.3: Statechart einer elementaren Aktivitäteninstanz (vereinfacht) (aus [Reic00b])

Einen Sonderfall nimmt der Zustand *SKIPPED* ein. In diesen Zustand wechselt eine Aktivität, die zuvor den Zustand *NOT_ACTIVATED*, *ACTIVATED* oder *SELECTED* und deren Vorbedingungen für ihre Ausführung nicht mehr gegeben sind, z. B. wird dieser Zweig einer Verzweigung nicht mehr ausgeführt. Bei parallelen Verzweigungen mit finaler Auswahl kann sich ergeben, dass die Voraussetzung für die Ausführung der Aktivität im nachhinein nicht mehr gegeben sind. Dann wechselt sie aus den übrigen Zuständen ebenfalls in den Zustand *SKIPPED*. Eventuell sind hier aber noch begleitende Aktionen, wie Abbruch oder Kompensation der Instanz erforderlich.

Neben den verschiedenen Zuständen, die eine Aktivität einnehmen kann, werden auch die Kanten, über die die Aktivitäten im Graphen verbunden sind markiert. Sind die Vorbedingungen für die Ausführung einer Aktivität noch nicht erfüllt, die Aktivität also noch nicht aktiviert, befinde(n) sich die einmündende(n) Kante(n) im Zustand *NOT_SIGNALED*. Sind die Vorbedingungen für die Ausführung einer Aktivität erfüllt, werden die in die Aktivität einmündende(n) Kanten mit *TRUE_SIGNALED* gekennzeichnet. Die Kanten, die sich in einem alternativen Zweig des Ausführungsgraphen befinden und deren Aktivitäten nach Überprüfung der Verzweigungsbedingung nicht ausgeführt werden sollen, erhalten den Zustand *FALSE_SIGNALED*.

Die Kanten- und Knotenmarkierungen sind in Abb. 6.4 dargestellt. In Abb. 6.5 ist ein Beispiel für die Anfangsmarkierung eines Workflows gegeben und im Vergleich dazu die Markierung des Graphen nach fortgeschrittener Bearbeitung.

6.2.3 Vertreterregelungen

Bisher wurde immer davon ausgegangen, dass ein zuständiger Bearbeiter immer erreichbar ist, und er die Einträge seiner Arbeitsliste abarbeiten kann. Selbst wenn er die Bearbeitung

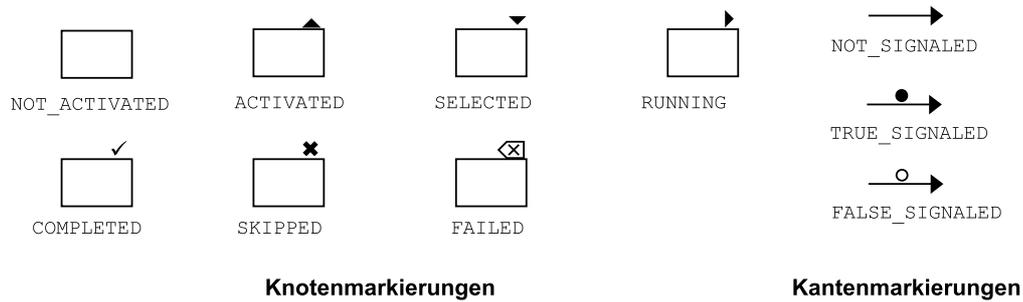


Abb. 6.4: Graphische Darstellung von Knoten- und Kantenmarkierungen (aus [Reic00b])

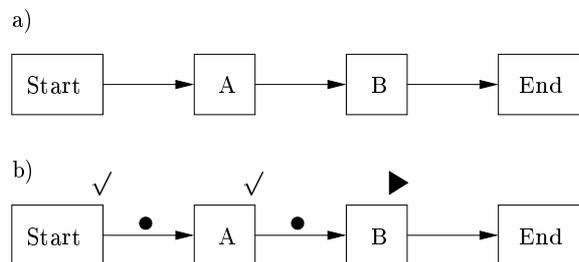


Abb. 6.5: a) Anfangsmarkierung eines Ausführungsgraphen und b) Markierung während der Bearbeitung eines Prozesses

einer Aufgabe unterbricht, ist er zu einem späteren Zeitpunkt in der Lage, die Aufgabe fertigzustellen. In der Wirklichkeit sieht es aber oft so aus, dass der Bearbeiter nicht immer in der Lage ist, eine Aufgabe zu übernehmen, er kann z. B. krank sein oder sich im Urlaub befinden. Für diese Zwecke ist es sinnvoll, einen Stellvertreter definiert zu haben, damit die Arbeit erledigt wird und nicht ins Stocken gerät. Auch gibt es Aufgaben, die sich an einen strikten Zeitplan halten müssen bzw. die in einem bestimmten Zeitrahmen abgearbeitet werden müssen, damit nachfolgende Tätigkeiten ebenfalls durchgeführt werden können.

In ADEPT haben Vertreterregelungen, wie bereits in 4.2.2 beschrieben, die Form von Bearbeiterzuordnungsausdrücken. Diese können sich sowohl auf Stellen als auch auf Mitarbeiter oder Rollen beziehen. Zusätzlich kann die Vertreterregelung noch mit der Aufgabenkategorie der zu vertretenden Stelle gekoppelt sein. Ist die Vertretung einer Stelle unabhängig von der Aufgabenkategorie, gibt es die Möglichkeit, die Kategorie *ALL* anzugeben, so dass kein Stellvertreter mit speziellen Kenntnissen ermittelt werden muss. Der Bearbeiterausdruck für eine Vertreterregelung einer Schwester auf der Station eines Krankenhauses ist im Folgenden gegeben.

Beispiel 1: Schwester „Müller“ von Station 2a soll in allen Aufgaben durch eine Pflegekraft der Station 2a vertreten werden. Sie besetzt eine Stelle mit der Stellenbeschreibung „Pflege_1_S2a“. Es ergibt sich folgende Vertreterregelung:

Stelle:	Pflege_1_S2a
Vertretung für:	„Müller“
Aufgabenkategorie: ALL	Vertreten durch: OE = 'Station 2a' ∧ R = 'Schwester'

Beispiel 2: Chefarzt „Meyer“ von Station 1 soll in den verschiedenen Bereichen wie Lehre, Forschung, Visite durch bestimmte Personen vertreten werden. Zum Chefarzt gehört die Stellenbeschreibung „Chefarzt_S1“. Es gibt sich die folgende Vertreterregelung:

Stelle:	Chefarzt_S1
Vertretung für:	„Meyer“
Aufgabenkategorie: - Management-Aufgaben - ärztliche Aufgaben - Forschung/Lehre	Vertreten durch: — OE = 'Station 1' \wedge R = '1. leitender Oberarzt' OE = 'Station 1' \wedge R = '1. leitender Oberarzt'

Die in ADEPT gewählte Modellierung deckt einfache Regelungen, wie sie in existierenden Systemen (s. Tabelle 2.2) geboten werden, ebenfalls ab. D. h. einer Stelle kann als Vertretung eine andere Stelle oder eine konkrete Person zugewiesen werden. Darüber hinaus aber gestattet es die vorgeschlagene Form der Vertretung, dass auch komplexere, in der Praxis jedoch auftretende Fälle, abgedeckt werden zu können. Insbesondere kann eine Vertretung für eine Stelle sich für unterschiedliche Aufgabenbereiche auch unterschiedlich darstellen. Die Beschreibung von Vertreterregelungen mittels allgemeiner Bearbeiterformeln bedeutet darüber hinaus implementierungstechnisch keinen wirklichen Zeitaufwand, da entsprechende Ausdrücke oder Formeln auch im Rahmen der normalen Bearbeiterauflösungen verarbeitet werden. Durch die Möglichkeit, Vertretungen nicht ausschließlich auf Mitarbeiter- oder Stelzebene festzulegen, resultieren auch pflegbarere/wartbarere Modelle.

6.2.4 Ausnahmebehandlungen

Bearbeiterzuordnungsdrücke sind im Normalfall so zusammengesetzt, dass sie anhand des Organisations-Modells des Unternehmens aufgelöst werden können. Es ergibt sich entweder eine einzelne Person, die die zugehörige Aufgabe erledigen soll oder es ergibt sich eine Menge von Bearbeitern, denen die Aufgabe zur Bearbeitung angeboten wird. Einer dieser Bearbeiter selektiert die Aufgabe und erledigt sie. Kann ein Bearbeiterzuordnungsdruck jedoch nicht nach dem Organisations-Modell aufgelöst werden, d. h. die Menge der potentiellen Bearbeiter ist leer, entsteht ein Problem. In diesem Fall sollte ein *Defaultbearbeiter* zur Verfügung stehen, der eine nicht zuweisbare Aufgabe zugeteilt bekommt. Als guter Defaultwert könnte hier der Prozessverantwortliche, wie in Kap. 6.1.3 bereits beschrieben, eingesetzt werden. Existiert kein Prozessverantwortlicher, muss der Systemadministrator den Prozess noch einmal modifizieren.

6.3 Zugriffsrechte

Zugriffsrechte beschreiben das Recht, auf Daten, die in einem Workflow entstehen, lesend und/oder schreibend zugreifen zu können. Folgende Daten sind zu unterscheiden:

- Workflow-Kontrolldaten (WF-Status, Ausführungs- und Änderungshistorie, Arbeitslisten)
- WF-relevante Daten und Anwendungsdaten (Datenelemente und externe Datenobjekte (Dokumente, Patientendaten, ...))
- WF-Modelle und Organisationsmodelle (der Benutzer sieht die Struktur von Prozessgraphen)

Können sich die Zugriffsrechte während der Bearbeitung eines Workflows ändern, ist es sinnvoll, die Zugriffsberechtigungen unter der Berücksichtigung einer Prozessvariablen zu vergeben. Dies ist z. B. bei Veränderungen der zuständigen Organisationseinheiten, wie in Beispiel 7 aus Kap. 3.2 beschrieben, von Vorteil, da angeforderte Berichte dann direkt an die neue OE weitergeleitet werden können. Auf den letzten Fall ist in Abschnitt 6.3.1 näher eingegangen.

6.3.1 Aktueller Workflow-Status

Der Status eines Workflows ergibt sich aus dem Zustand seiner einzelnen Aktivitäten (siehe Abb. 6.3). Demnach spiegelt der aktuelle Workflow-Status einen Ausschnitt der Prozesshistorie wider. Der Status einer Prozessinstanz wird entweder aus einer abstrakten oder konkreten Sicht (s. Abb. 6.6 und 6.7) heraus betrachtet.

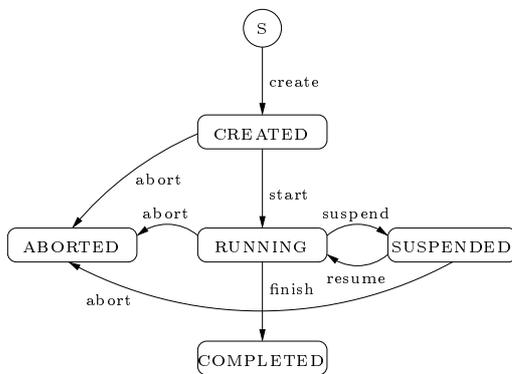


Abb. 6.6: Beispiel für eine abstrakte Sicht auf eine Prozessinstanz

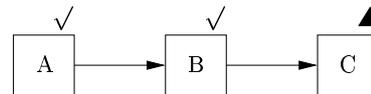


Abb. 6.7: Beispiel für eine konkrete Sicht auf eine Prozessinstanz

Wie bei der Prozesshistorie gilt auch hier, dass der Status des Workflows nie von einem Bearbeiter aktiv geändert werden darf. Das Prozess-Management-System selbst verändert den Status eines Workflows, indem es verschiedene Dinge protokolliert, die die Ausführung des Workflows betreffen, aber nicht zur Bearbeitung eines Prozesses gehören. Darunter fallen Start- und Endzeiten einer Prozessinstanz und ihrer Aktivitäten, der Zustand, in dem sich die Instanz gerade befindet, der Bearbeiter einer Aktivität, Ein- und Ausgabedaten der einzelnen Aktivitäten. Diese Daten werden in der Ausführungshistorie (s. Kap. 6.3.3) gespeichert. Nur ausgewählte Personen dürfen lesend auf bestimmte Daten des Workflow-Status zugreifen. Z. B. besitzt der Prozessverantwortliche die Berechtigung, sich über den Zustand seiner Prozesse bzw. deren Aktivitäten zu informieren. Er kann also die gesamte Ausführungshistorie seiner Instanzen einsehen. Ein Bearbeiter einer Aktivität darf dagegen lediglich über seine Arbeitsliste Einblick in von ihm selbst bearbeitete Tätigkeiten haben. Ein dabei auftretendes Problem ist: In wie weit darf ein Bearbeiter, der in der Vergangenheit an der Ausführung eines Prozesses beteiligt war, in der Gegenwart noch Einblick in die Historie dieser Prozessinstanz erhalten. Dies ist ein grundsätzliches Problem der Zugriffsberechtigung von Benutzern auf Prozessinstanzen innerhalb des Ablaufs eines Workflows. Bezogen auf ein Beispiel aus dem Krankenhaus heißt das: für einen Patienten gibt es mehrere Vorgänge, die teilweise auch parallel ablaufen, wie z. B. Laboruntersuchungen und eine Röntgenuntersuchung. Diese Untersuchungen wurden aber von verschiedenen Ärzten oder gar Abteilungen in Auftrag gegeben. Wer darf nun auf die verschiedenen Ergebnisse zugreifen? Das gleiche Problem besteht bei der Verlegung eines Patienten von einer auf eine andere Station. Die Befunde einer Untersuchung werden meistens an die Station geschickt, die die Untersuchungen angefordert hat und werden nicht automatisch an die neue Station weitergeleitet. Das führt in der Praxis

dazu, dass Untersuchungen doppelt angefordert und auch durchgeführt werden, lediglich mit dem Unterschied, dass eine andere Station diese Untersuchungen anfordert. Ein Beispiel für die verschiedenen Zugriffsberechtigungen ist im Folgenden gegeben.

Gegeben ist ein Benutzer „Dr. Müller“ in einer Organisationseinheit „Station 2a“. Auf „Station 2a“ startet „Dr. Müller“ eine Prozessinstanz, noch sind alle Verantwortlichkeiten genau geregelt: Dr. Müller hat Zugriff auf alle Prozessinstanzen, die von Benutzern der gleichen OE (hier: „Station 2a“) erzeugt werden, somit hat er auch Zugriff auf alle Prozessinstanzen, die der Benutzer (hier: „Dr. Müller“) selbst erzeugt hat. Dies kann in Bearbeiterzuordnungsdrücken leicht festgelegt werden. Alle Benutzer, die sich in der gleichen OE befinden wie der Erzeuger der Instanz wäre $OE = OE(\text{Prozesscreator}) (\wedge \dots)$. Wird der Patient verlegt, wird eine Möglichkeit benötigt, die Zugriffsberechtigungen auf „seine“ Prozessinstanzen dynamisch zu ändern. Dazu muss sich der Bearbeiterzuordnungsdruck auf vorhergehende Ereignisse und/oder Benutzer beziehen können, bzw. ein Vorgang muss auch an einen anderen Bereich abzugeben sein. Um dieses Problem zu lösen, sollten abhängige Bearbeiterzuordnungsdrücke und eine zusätzliche Variable verwendet werden, um organisatorische Elemente, wie z. B. die Organisationseinheit, abzubilden. Besitzt eine Prozessinstanz eine solche Variable z. B. die OE, kann diese leicht auch von extern geändert werden, wodurch auch die Zugriffsberechtigungen im Falle einer Verlegung automatisch neu vergeben würden (s. Abb. 4.6). Ein Beispiel für eine derartige Prozessinstanz mit zugeordneter Variable ist in Abb. 6.8 gegeben.

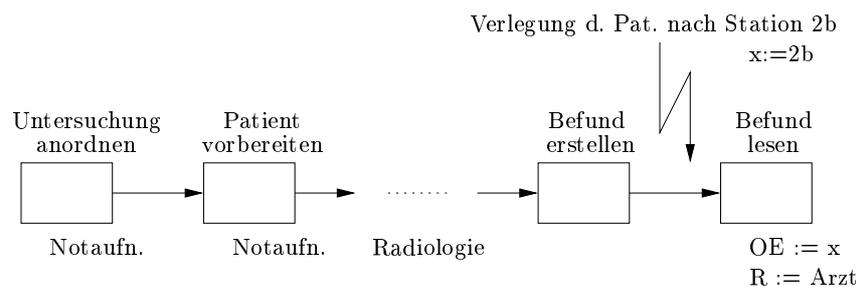


Abb. 6.8: Beispiel für den Ablauf einer Untersuchung mit Veränderung der OE

6.3.2 Zugriff auf Datenelemente

Datenelemente sind die Daten, die der Bearbeiter während der Ausführung einer Prozessinstanz benötigt, wie z. B. Patientendaten oder Referenzen auf Dokumente, die er erstellen oder überarbeiten soll. Die Berechtigung für Lese- und Schreibzugriffe auf die Datenelemente erfolgt indirekt über die Aktivitäten. Sollen allerdings weitere bzw. alle bisher angefallenen Daten eingesehen werden, reicht dieser Zugriff evtl. nicht aus. Für diesen Fall müssen geeignete Defaultwerte definiert werden.

Der Zugriff auf Daten sollte möglichst einfach und leicht zu verändern sein. Die einfachste Regelung ist über Lese- und Schreibzugriffe zu realisieren. In ADEPT wird der Workflow getrennt nach Kontrollfluss und Datenfluss modelliert. Der Datenaustausch geschieht über sogenannte Ein- und Ausgabecontainer. Ein Eingabecontainer kann nur ausgelesen werden, wenn eine vorhergehende Workflowinstanz in diesen Container ihre Ausgabedaten geschrieben hat. Durch die getrennte Modellierung des Datenflusses ist schnell ersichtlich, welche Daten der Bearbeiter einer Prozessinstanz zur Bewältigung seiner Aufgabe erhalten muss. Bei der Definition der Zugriffsrechte auf die Datenelemente gibt es zwei verschiedene Varianten:

1. Für jedes Datenelement wird explizit festgelegt, wer welche Zugriffsberechtigung auf

dieses besitzt.

2. Datenelemente werden in disjunkten Gruppen zusammengefasst und auf diesen Gruppen werden die Zugriffsberechtigungen erteilt (s. Abb. 6.9).

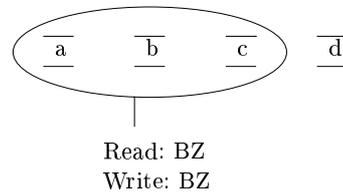


Abb. 6.9: Beispiel für Zugriffsberechtigungen in Gruppenform

Der Nachteil der ersten Variante ist, dass dieser Ansatz sehr aufwendig und wartungsintensiv ist. Auch kann passiven Bearbeitern kein Einblick in die Daten gewährt werden. Der Vorteil der zweiten Variante ist, dass auch spezielle Benutzer die Daten einsehen können, obwohl sie nicht aktiv in den Workflow eingebunden sind.

Neben diesen Berechtigungen muss auch berücksichtigt werden, welche Daten der Bearbeiter noch zusätzlich einsehen darf, z. B. über eine weitere Zugriffserlaubnis auf zusätzliche Dokumente.

Bei den in einem PMS auftretenden Daten müssen zwei Arten von Datenelementen unterschieden werden - Values und Referenzen. Unter Values sind wirkliche Werte strukturierter Daten zu verstehen. Der Zugriff auf diese Daten ist über die Aktivitäten selbst geregelt. Referenzen verweisen nur auf das tatsächliche Objekt, der Zugriff wird über ein Objektzugriffssystem geregelt. In ADEPT können auch nur Verweise auf die Daten gespeichert werden, um das zu bewegende Datenvolumen so gering wie möglich zu halten. Das hat den Vorteil, dass die Daten für die potentiellen Bearbeiter nicht mehrfach auf dem Server liegen. Der Bearbeiter, der die Aktivität letztendlich startet, holt sich die benötigten Daten dann erst auf seinen Rechner. So bleibt das PMS schnell und wird nicht durch unnötigen Datentransport verlangsamt.

6.3.3 Historie

Um sich über den bisherigen Verlauf einer Prozessinstanz zu informieren, ist ein Zugriff auf deren Ausführungs- und Änderungshistorie vonnöten. Die Historie zeichnet in chronologischer Reihenfolge die Statusinformationen für jede Aktivitäteninstanz eines Prozesses auf. Sie wird auch als Ausführungshistorie bezeichnet. In der Änderungshistorie werden solche Informationen protokolliert, die die Änderung eines Workflows betreffen. Auf die Änderungshistorie soll hier nicht weiter eingegangen werden. Ein kurzes Beispiel für die Ausführungshistorie einer Aktivität folgt nun:

Objekt	Aktivität A
Ereignis	Start
Zeit	3.2.00 12:00
User	userid=4711

Die Historie einer Instanz oder eines Workflows darf niemals von einem Benutzer aktiv manipuliert werden. Lediglich das PMS greift schreibend auf die Historie zu.

Die Ausführungshistorie darf nicht von jedem Bearbeiter eingesehen werden. Für ihre Einsicht wird wieder ein Bearbeiterzuordnungsausdruck festgelegt, der den Zugriff auf die Historien-daten regelt, ihre Wirkungsweise ist mit der eines Filters gleichzusetzen. Die Daten, die eine Benutzerstufe einsehen kann, ist je Eintrag attributbezogen abhängig von

- Ereignistypen
- Personengruppe (Benutzertypen)
- der Zeit
- der Aktivität

Ereignistypen heißt in diesem Zusammenhang, der Benutzer darf nur Historiendaten einsehen, die eine bestimmte Art von Ereignis widerspiegeln (z. B. Start, Ende, ...). Benutzertypen werden über Bearbeiterformeln festgelegt, sie geben an, wer bestimmte Daten einsehen darf. Zeitbezogene Daten sind Daten von Aktivitäten, die einen bestimmten Zeitstempel noch nicht überschritten haben bzw. erst nach einem bestimmten Zeitstempel aufgetreten sind. Darf ein Benutzer nur bestimmte Daten ausgewählter Aktivitäten einsehen, spricht man von Aktivitätenbezogenheit. Dies ist jedoch nicht unkritisch, da innerhalb einer Aktivität auch unvorhergesehene Daten entstehen können, die der normale Bearbeiter nicht einsehen darf. Zur genauen Definition der Zugriffsrechte auf die Historie ist es zweckmäßig, eine Art Filter für verschiedene Benutzergruppen zu definieren. Ein solcher Filter könnte z. B. anhand der in Tab. 6.1 dargestellten Stufen aufgebaut sein.

Stufe	Art der sichtbaren Daten	Person, die die Daten einsehen darf
1	Nichts	Personen, die nicht an der Ausführung des WFs beteiligt sind
2	An welcher Stelle befindet sich die Ausführung des WFs aktuell?	Personen, die mit dem Kunden Kontakt haben, z. B. Service Personen, die zwar an der Ausführung des WFs aber nicht an der aktuellen Aktivität beteiligt sind
3	Daten, die die von bestimmten Bearbeitern selbst bearbeiteten Aktivitäten betreffen	Bearbeiter der Vergangenheit
4	Eingabedaten der aktuellen Aktivität, alle Bearbeiter der bisherigen Aktivitäten für Nachfragen	Bearbeiter(Aktivität)
5	Wie weit ist der WF fortgeschritten? (z. B. um zu sehen, wann bestimmte Arbeiten anfallen)	sichere zukünftige Bearbeiter Vorgesetzte von potentiellen zukünftigen Bearbeitern
6	Alles (auch Gründe für verschiedene Ereignisse)	Prozessverantwortlicher (Aktivität/Instanz)

Tabelle 6.1: Beispiel für eine mögliche Filterdeklaration

Die Zuordnung der Mitarbeiter zu den verschiedenen Stufen der Filter erfolgt wie gehabt durch Bearbeiterzuordnungsausdrücke.

Bisher wurde das ganze bei einfachen Workflows betrachtet. Es müssen aber auch hierarchische Workflows betrachtet werden. Befinden sich in einem Workflow Subprozesse, ist dieses oftmals mit einem Wechsel in eine andere Organisationseinheit verbunden. Hier tritt die Frage auf, darf ein Bearbeiter die Historie des Subworkflows einsehen oder nicht. Generell sollte man diese Frage sicherlich mit nein beantworten, da es die übergeordnete Organisationseinheiten

nichts angeht, welche Schritte von wem in der anderen Organisationseinheit durchgeführt wurden. Mitarbeiter der übergeordneten Organisationseinheit dürfen lediglich sehen, wann der Subprozess gestartet und wann er beendet wurde.

6.3.4 Arbeitslisten

Arbeitslisten (s. Abb. 2.4, Seite 18) sind Listen von zur Bearbeitung anstehenden oder bereits in Bearbeitung befindlichen Tätigkeiten. Eine Arbeitsliste zeigt auch solche Aktivitäten an, die das PMS diesem und anderen möglichen Bearbeitern zur Bearbeitung anbietet. Sobald eine solche Aktivität in der Arbeitsliste zur Bearbeitung ausgewählt wird, wird sie bei allen anderen in Frage kommenden Bearbeitern aus der Arbeitsliste gelöscht.

Eine persönliche Arbeitsliste kann nur von der Person eingesehen werden, deren Tätigkeiten in ihr aufgelistet sind, d. h. nur diese Person kann lesend auf sie zugreifen. In der Regel darf nicht manipulierend auf sie zugegriffen werden, eine Ausnahme folgt weiter unten. Die Arbeitsliste wird also nicht aktiv vom Bearbeiter verändert, die Änderungen an ihr werden wie oben beschrieben automatisch vom PMS durchgeführt.

Ist ein Bearbeiter jedoch abwesend, darf der Betrieb nicht ins Stocken geraten. Für diesen Fall ist in der Regel eine Vertreterregelung gegeben, so dass die Aktivitäten der Arbeitsliste an den Vertreter weitergeleitet werden oder er die Arbeitsliste der zu vertretenden Person einsehen kann (s. Abschnitt 6.2.3). Ist keine Vertreterregelung gegeben, muss einer weiteren Person Einblick in die Arbeitsliste des abwesenden Bearbeiters gewährt werden. Sinnvoll wäre in diesem Zusammenhang sicherlich, dass der Prozessverantwortliche Zugriff auf diese „fremden“ Arbeitslisten erhält. Neben dem lesenden Zugriff des Prozessverantwortlichen, muss dieser in der Lage sein, die Arbeitsliste eines anderen Bearbeiters zu manipulieren. Er muss auch schreibenden Zugriff auf andere Arbeitslisten erhalten, um im Falle der Abwesenheit des Bearbeiters, Arbeitslisteneinträge an andere Bearbeiter delegieren zu können.

Kapitel 7

Anpassungs- und Änderungsrechte

7.1 Problemstellung

Während der Bearbeitung eines Prozesses kann es zu situationsbedingten Notfällen kommen, so dass der Prozess nicht anhand des Standardablaufes weiter bearbeitet werden kann. In diesem Fall muss der Benutzer auf Abweichungen vom Standardablauf eingehen können, ohne dass dieser vom PMS in seinem Handeln eingeschränkt wird. Dazu werden Abweichungen, die im Voraus bekannt sind, bereits bei der Modellierung berücksichtigt und modelliert. Auf Änderungen, die zum Zeitpunkt der Modellierung noch nicht absehbar sind, muss der Benutzer ad-hoc reagieren können.

In beiden Fällen müssen jedoch Berechtigungen definiert werden, die regeln, wer wie auf diese Abweichungen reagieren können soll. Bisher fehlt es in ADEPT jedoch am geeigneten Sicherheitskonzept. Im Folgenden werden Ansätze zur Festlegung und Verwaltung entsprechender Berechtigungen unterbreitet.

7.2 Planbare Abweichungen

In Abschnitt 4.3.2.1 wurden die von ADEPT gebotenen Möglichkeiten angesprochen, vorhersehbare (planbare) Abweichungen vom Standardablauf vorzumodellieren (A-priori-Flexibilität). Solch eine Vormodellierung ist auch vorteilhaft im Hinblick auf die Autorisierung entsprechender Abweichungen. Im Folgenden wird anhand der von ADEPT bereitgestellten Konstrukte zur Vormodellierung von ausnahmebedingten Abweichungen dargelegt, wie in diesem Zusammenhang entsprechende Ausführungsrechte festzulegen sind.

7.2.1 Überspringen von Aktivitäten ohne Nachholen

Das Überspringen von Aktivitäten ohne Nachholen wird über die in Kap. 4.3.2.1 vorgestellten Shortcuts modelliert. Die Menge der Personen, die potenziell zur Auswahl eines solchen Shortcuts berechtigt sein sollen, wird über einen entsprechenden Bearbeiterzuordnungsausdruck festgelegt. Dieser ist unmittelbar der betreffenden Shortcut-Kante zugeordnet.

Bei Auswahl des Shortcuts werden bereits gestartete oder beendete Aktivitäten, die übersprungen werden sollen, vom System zurückgesetzt und die von ihnen ausgelösten Effekte kompensiert. Dabei orientiert sich das Prozess-Management-System sowohl an der Ausführungshistorie als auch an den Historien der einzelnen Datenelemente.

In Abb. 7.1 ist ein Beispiel für das Überspringen von Aktivitäten ohne Nachholen mit entsprechender Bearbeiterzuordnung (aus Sicht des Modellierers) skizziert. Während ein Termin

für eine radiologische Untersuchung vereinbart wird, verschlechtert sich der Zustand des Patienten so, dass nicht weiter auf einen Termin gewartet werden kann. Der Patient muss direkt der Untersuchungseinheit zugeführt, die Terminvereinbarung abgebrochen und weitere dazwischen liegende Schritte übersprungen werden. Ist die Untersuchung gestartet, ist es nicht mehr sinnvoll, einen Termin zu vereinbaren, den Transport des Patienten zur Untersuchungsstelle anzufordern und ihn zur Untersuchungsstelle zu transportieren.

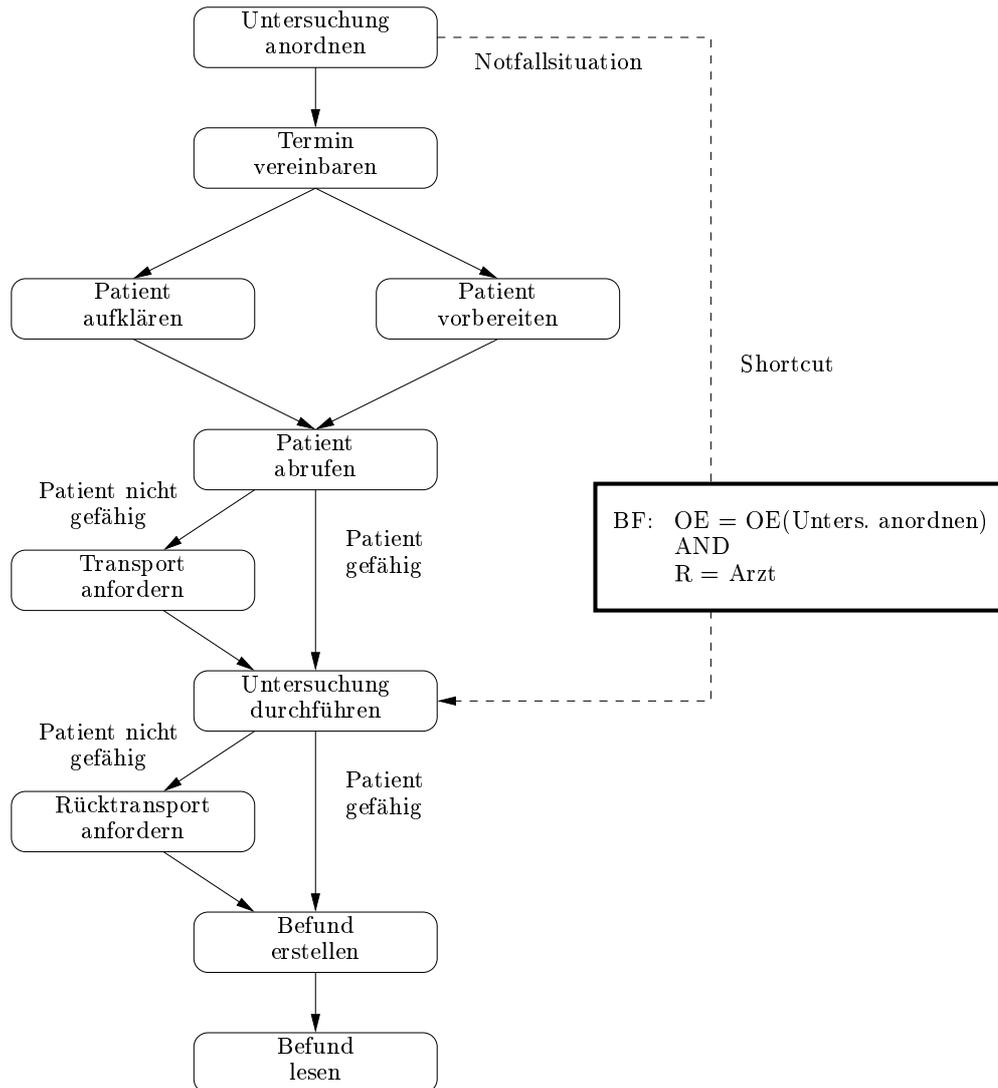


Abb. 7.1: Beispiel für das Überspringen von Aktivitäten ohne Nachholen aus Modellierersicht

Die Umsetzung des Shortcut-Konstrukts aus Abb. 7.1 in das ADEPT-Basismodell ist in Abb. 7.2 skizziert. Wie leicht ersichtlich ist, kann die mit der Shortcut-Kante verknüpfte Bearbeiterformel dem bei der Transformation entstandenen Sprungknoten zugewiesen werden. Es sind also zur Laufzeit keine gesonderten Kontrollmechanismen hinsichtlich den Autorisierungen entsprechender Sprünge erforderlich.

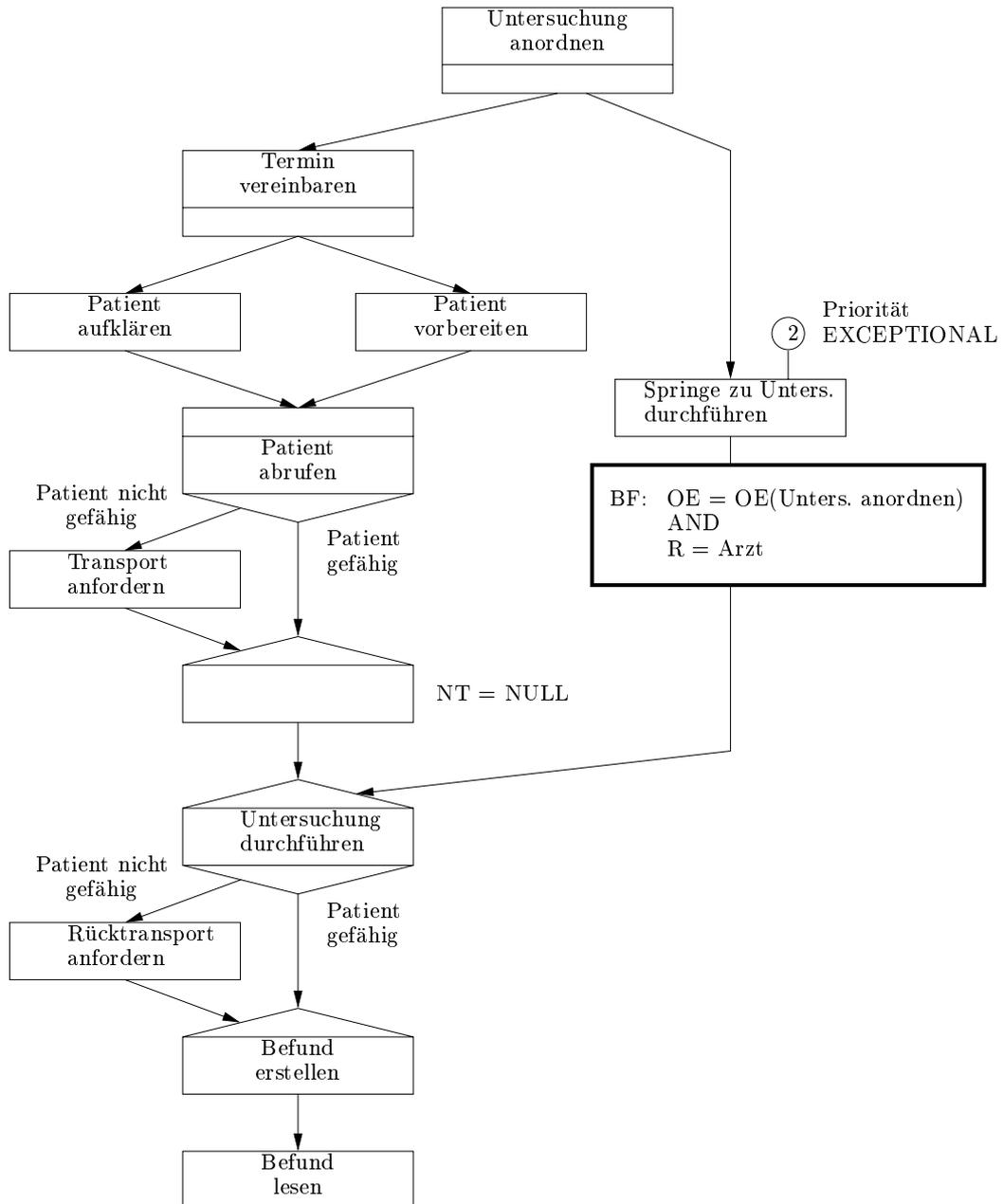


Abb. 7.2: Umsetzung des Shortcuts aus Abb. 7.1 im ADEPT-Basismodell

7.2.2 Überspringen mit Nachholen

Ein Beispiel für einen Shortcut mit Nachholen der übersprungenen Schritte (aus Sicht des Modellierers) ist in Abb. 7.3 gegeben. Hier kann der behandelnde Arzt den Schritt „Patient aufklären“ außerplanmäßig der Aktivität „Termin vereinbaren“ vorziehen, etwa wenn er sich gerade im Gespräch mit dem Patienten befindet.

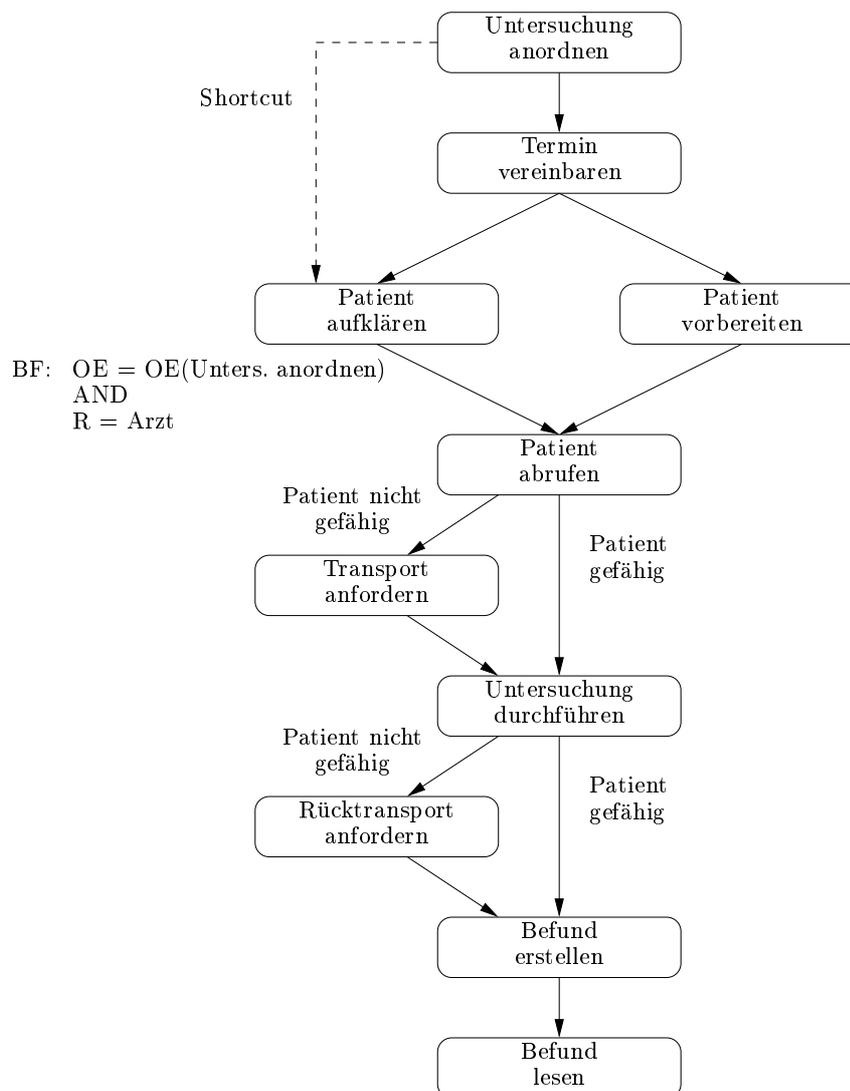


Abb. 7.3: Beispiel für das Überspringen von Aktivitäten mit Nachholen der Schritte (aus Modellierersicht)

In Abb. 7.4 ist die Umsetzung dieses Vorwärtssprunges im ADEPT-Basismodell skizziert. Aus der Abbildung ist ersichtlich, dass der gewünschte Effekt durch Verwendung von Prioritäten und Priorisierungskanten erreicht wird. Dabei sind keine gesonderten Rechte für die vorzeitige Bearbeitung des Schrittes „Patient aufklären“ festzulegen. Vielmehr kann diese durch jeden für die Bearbeitung dieser Tätigkeit qualifizierten Mitarbeiter erfolgen. Die zugehörige Bearbeitermenge ist durch den regulären Bearbeiterzuordnungsdruck der Aktivität definiert.

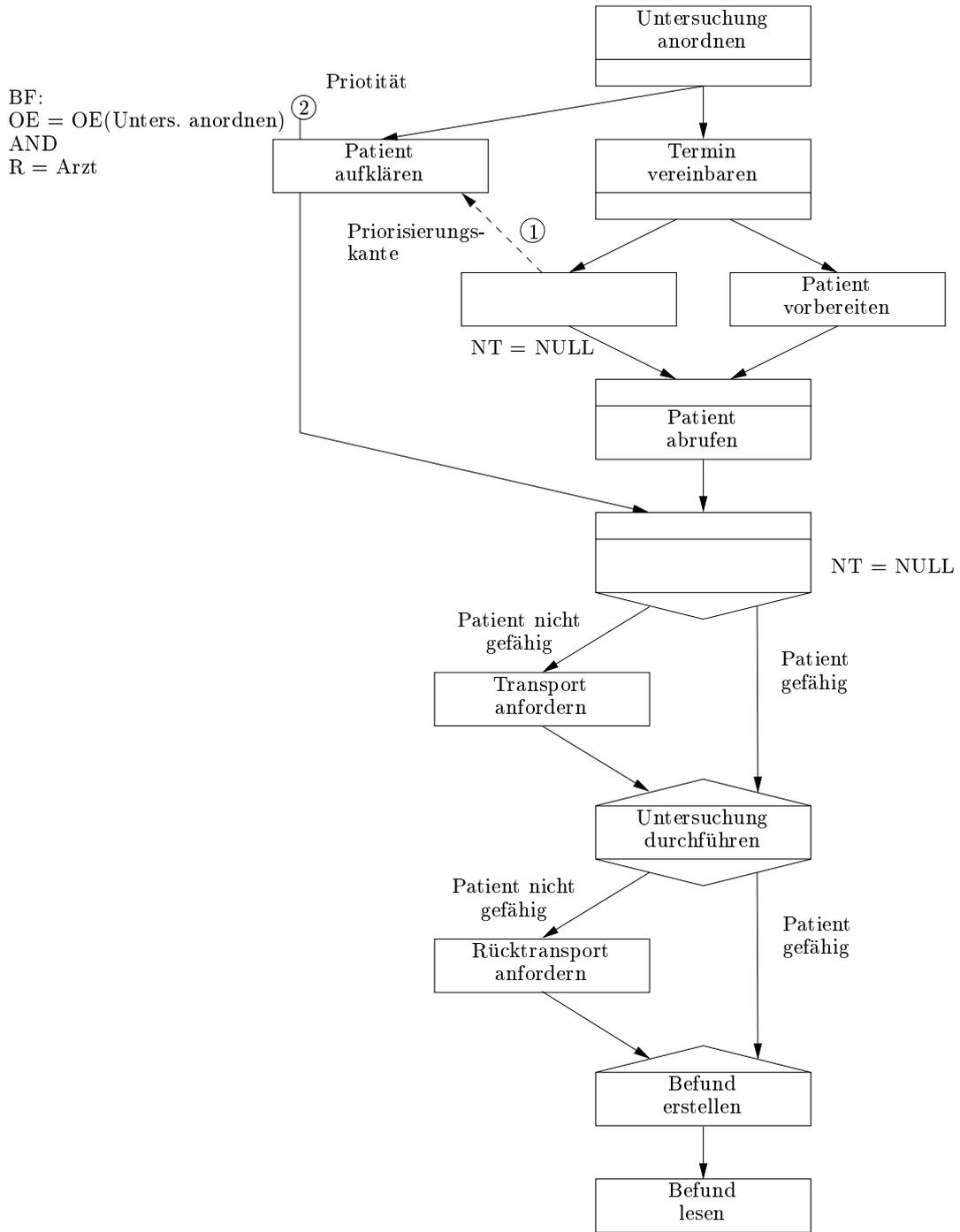


Abb. 7.4: Umsetzung des Shortcut-Konstruktes aus Abb. 7.3 im ADEPT-Basismodell

7.2.3 Rücksprung über Fehlerkanten

Beim Scheitern einer Aktivitätenbearbeitung kann die Prozessinstanz nicht wie gewohnt weiter bearbeitet werden. Eine Möglichkeit zur Behandlung eines solchen semantischen Fehlers besteht darin, den Ablauf partiell zurückzusetzen, so dass die Bearbeitung in einem früheren, konsistenten Zustand wieder aufsetzen kann. Abb. 7.5 zeigt die Folgen des Scheiterns der Bearbeitung der Aktivität „Untersuchung durchführen“. Dieser Fall kann z. B. eintreten, wenn der Patient nicht nüchtern zur Untersuchung erscheint. Die Untersuchung muss dann abgebrochen und ein neuer Termin vereinbart werden.

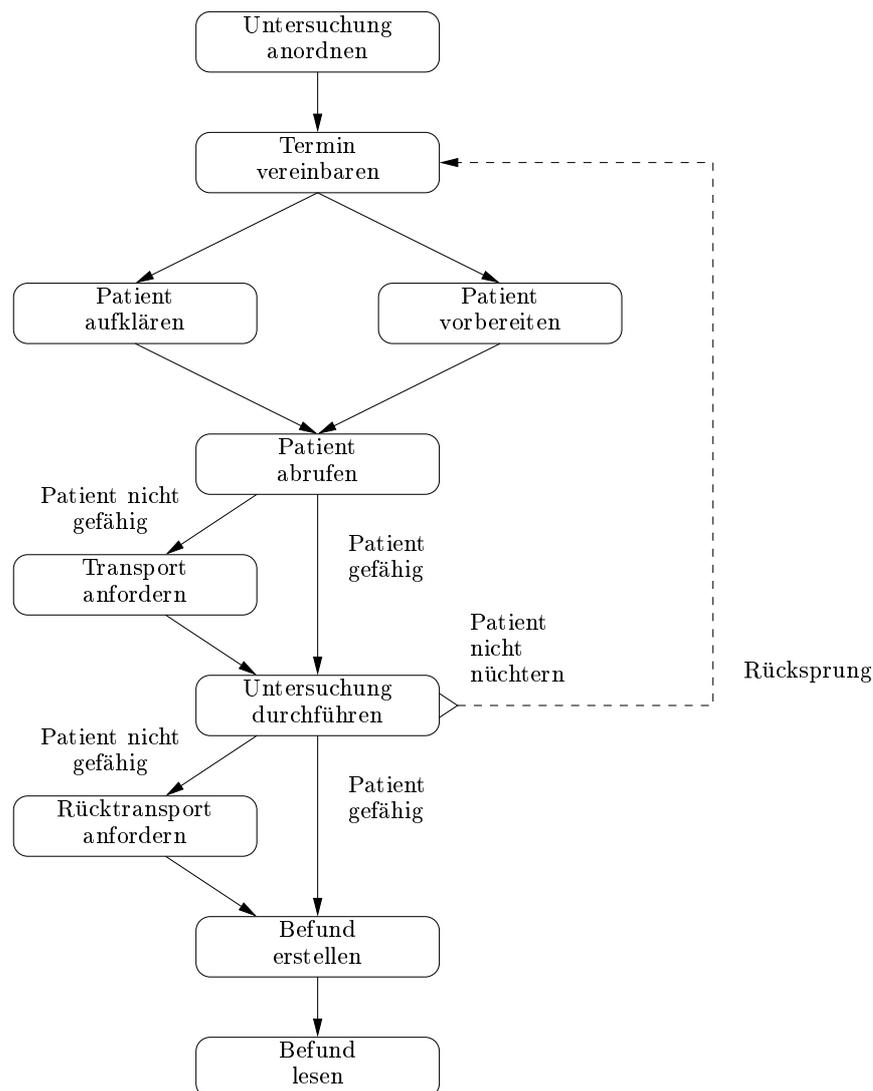


Abb. 7.5: Verhalten im Falle des Scheiterns der Untersuchungsdurchführung

Ob ein Prozessschritt erfolgreich bearbeitet werden kann oder nicht bzw. aus welchem Grund er scheitert, kann in der Regel nur vom Bearbeiter des entsprechenden Schrittes entschieden werden. D. h. er muss die Berechtigung besitzen, die Ausführung einer Aktivität abzubrechen und sie, unter Angabe eines entsprechenden Fehlercodes als gescheitert zu kennzeichnen. Das bedeutet, dass auf Modellierungsebene keine gesonderten Angaben hinsichtlich der Berechtigung erforderlich sind.

7.2.4 Benutzerinitiierte Rücksprünge

ADEPT bietet darüber hinaus die Möglichkeit, benutzerinitiierte Rücksprünge im Ablauf zu erlauben. Grundlage für ihre Vormodellierung bilden die angesprochenen Fehlerkanten sowie die erwähnten Knotenprioritäten.

Bezogen auf das in Abb. 4.14 auf Seite 48 dargestellte Beispiel eines benutzerinitiierten Rücksprungs sollen die Berechtigungen kurz beschrieben werden. Die Rechtfestlegung erfolgt hier ähnlich wie bei den in Abschnitt 7.2.1 vorgestellten Shortcuts, indem der Modellierer für RegainControl-Kanten eine entsprechende Bearbeiterzuordnungsformel festlegt. Diese wird dann bei der Transformation des Graphen in das ADEPT-Basismodell einem Rücksprungknoten (im Beispiel der Knoten RC) zugeordnet. Im Zusammenhang mit der Vormodellierung benutzerinitiiertter Rücksprünge ist somit kein gesondertes Rechtekonzept erforderlich.

7.2.5 Fazit

Durch die Vormodellierung von planbaren Ausnahmen können aufwendige Adaptionen zur Laufzeit, zumindest teilweise, vermieden werden. ADEPT bietet hierfür intuitive Modellierkonstrukte an, deren operationale Semantik durch das ADEPT-Basismodell präzise definiert ist. In diesem Abschnitt wurde gezeigt, welche Rechte im Zusammenhang mit planbaren Abweichungen erforderlich sind. Ihre Festlegung und Kontrolle erfordert keine speziellen Konstrukte, sondern kann mit Mitteln (z. B. Bearbeiterzuordnungen) erfolgen, wie sie im „Normalfall“ verwendet werden.

7.3 Ad-hoc Änderungen

Sind Ausnahmen nicht bereits zum Zeitpunkt der Modellierung bekannt, muss der berechtigte Benutzer sie zur Ausführungszeit behandeln können. Die Ad-hoc Anpassungsmöglichkeiten, die das ADEPT-PMS zur Unterstützung der Anwender bereitstellt, wurden bereits in Abschnitt 4.3.2.2 vorgestellt. Im Folgenden soll auf die in diesem Zusammenhang notwendigen Berechtigungen eingegangen werden. Diese werden anhand der Einfüge-, Lösch- und Verschiebeoperation von Aktivitäten näher beschrieben.

7.3.1 Rechtekonzept für das dynamische Einfügen von Aktivitäten

Im Folgenden wird gezeigt, welche Art von Berechtigungen für das dynamische Einfügen von Prozessschritten erforderlich sind und wie diese Rechte definiert werden können. Ein wichtiges Ziel dabei sind die eingangs dieser Arbeit geforderte Konsistenz und Pflegbarkeit der definierten Berechtigungen.

7.3.1.1 Erforderliche Rechte zur Kontrolle dynamischer Einfügeoperationen

Das dynamische Einfügen einer Aktivität in eine gegebene Prozessinstanz erfordert eine entsprechende Berechtigung der Person, die diese Einfügeoperation vornehmen will. In diesem Zusammenhang sind folgende Fragestellungen von Bedeutung:

1. Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) aus dem Repository darf ein bestimmter Benutzer generell einfügen?

Unabhängig von der Prozessvorlage bzw. -instanz, in die eine neue Aktivität eingefügt werden soll, stehen einem Benutzer im Allgemeinen nicht alle im Repository hinterlegten Aktivitätenvorlagen zur Auswahl. Ein Verwaltungsangestellter eines Krankenhauses etwa darf keine medizinische oder pflegerische Tätigkeit in einen Behandlungsprozess einfügen. Die Menge der Aktivitätenvorlagen, auf die ein bestimmter Benutzer beim Einfügen eines Schrittes Bezug nehmen darf, muss daher von vornherein eingeschränkt werden.

2. Welche Art von Aktivitäten (bzw. Aktivitätenvorlagen) dürfen in die Instanzen einer bestimmten Prozessvorlage eingefügt werden?

Im Allgemeinen dürfen zu einer Instanz einer bestimmten Prozessvorlage nicht Aktivitäten beliebigen Typs hinzugefügt werden. Ist z. B. ein Bestellvorgang von Medikamenten von einer Station im Krankenhaus in Auftrag gegeben worden, der unabhängig von der Behandlung eines konkreten Patienten ist, darf in die entsprechende Prozessinstanz keine patientenbezogene Aktivität eingefügt werden, (z. B. Pfl egetätigkeit „Waschen des Patienten“). Darf ein Benutzer eine bestimmte Aktivität in eine Instanz einer Prozessvorlage X einfügen, bedeutet dies also nicht, dass er dieselbe Aktivität auch in Instanzen anderer Vorlagen einfügen darf.

3. Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) in eine bestimmte Instanz einfügen?

Die Bedingungen 1 und 2 reichen allein noch nicht aus, um zu ermitteln, ob ein bestimmter Benutzer eine Aktivität in eine Prozessinstanz einfügen darf. Zu einer Prozessvorlage kann es Instanzen geben, für die ein bestimmter Benutzer eine Aktivität X einfügen darf, während er dies für andere Instanzen derselben Vorlage nicht darf. Beispielsweise können radiologische Untersuchungsabläufe von unterschiedlichen Stationen im Krankenhaus angefordert werden. Generell darf ein Stationsarzt einen zusätzlichen Vorbereitungsschritt in Instanzen der entsprechenden Prozessvorlage einfügen. Ob er dies für eine konkrete Instanz darf, hängt allerdings davon ab, ob er der behandelnden Einheit (d. h. Station) des Patienten angehört. Abb. 7.6 skizziert diesen Fall. Hier soll Aktivität X in die Prozessinstanz eingefügt werden.

4. In welche Bereiche eines Prozess-Ausführungsgraphen dürfen Aktivitäten eingefügt werden?

Die bisherigen Festlegungen regeln, wer welche Art von Aktivitäten in welche Prozessinstanzen einfügen darf. Nicht festgelegt ist durch sie, an welchen Positionen des Instanz-Ausführungsgraphen entsprechende Einfügungen erfolgen können sollen. Dieser Aspekt soll in der Folge zunächst ausgeklammert werden, am Ende dieses Kapitels aber wieder aufgegriffen werden.

Diese Fragestellungen werden in der Folge sukzessive behandelt, um zu zeigen, wie in ADEPT entsprechende Berechtigungen festgelegt werden können.

7.3.1.2 Vereinfachungen

Der Einfachheit halber wird für die folgenden Betrachtungen angenommen, dass die Festlegung der Rechte für eine bestimmte Prozessvorlage nicht für eingeschränkte Bereiche des

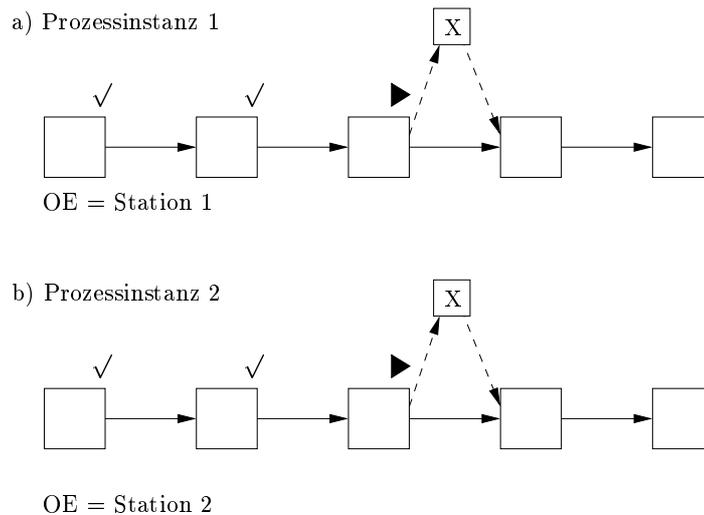


Abb. 7.6: Beispiel für zwei Prozessinstanzen derselben Prozessvorlage mit Zuordnung zu unterschiedlichen Krankenstationen. (a) Nur Mitarbeiter der „OE = Station 1“ dürfen Schritte einfügen. (b) Nur Mitarbeiter der „OE = Station 2“ dürfen Schritte einfügen.

zugehörigen Prozessgraphen erfolgen kann. D. h. eine bestimmte Aktivität kann von berechtigten Akteuren an beliebiger Stelle in der Prozessinstanz eingefügt werden, vorausgesetzt der aktuelle Status der Instanz läßt dies zu.

In die Prozessinstanz aus Abb. 7.7 darf ein berechtigter Benutzer z. B. eine neue Aktivität sowohl zwischen den Aktivitäten B und C, C und D als auch D und E einfügen. Er kann sie aber auch parallel zu B oder irgendeiner anderen Aktivität ausführen.

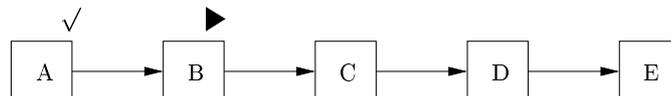


Abb. 7.7: Beispiel für eine einfache Prozessinstanz

Im Zusammenhang mit hierarchischen Prozessmodellen gelten Rechtfestlegungen immer in Bezug auf eine bestimmte Hierarchiestufe. Insbesondere können Rechte für das Einfügen einer Aktivität für Prozesse und zugehörige Sub-Prozesse differenziert festgelegt werden. Dadurch stellt die oben getroffene Vereinfachung keine wesentliche Einschränkung dar.

Für den Graphen in Abb. 7.7 bzw. die entsprechende Vorlage etwa kann aufgrund der getroffenen Vereinbarung nicht festgelegt werden, dass eine Aktivität X zwischen den Aktivitäten C und D nicht eingefügt werden kann. Bei Verwendung von Sub-Prozessen und differenzierter Rechtfestlegung auf den verschiedenen Prozess- bzw. Hierarchieebenen ist dies prinzipiell möglich. Es ergibt sich dann der Graph aus Abb. 7.8.

7.3.1.3 Kategorisierung von Prozess- und Aktivitätsvorlagen

Um Rechte für Einfüge-Operationen möglichst kompakt und pflegbar zu halten, bietet es sich an, für ihre Definition unterschiedliche Granularitätsstufen zu wählen. Zu diesem Zweck

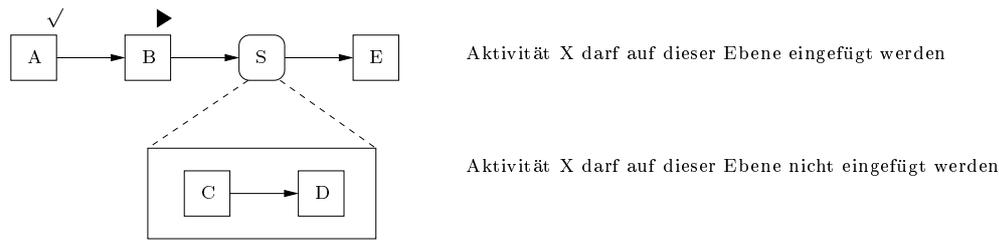


Abb. 7.8: Das Beispiel aus Abb. 7.7 mit der Festlegung, dass zwischen den Aktivitäten C und D keine zusätzliche Aktivität eingefügt werden darf.

sollte es möglich sein, die im Repository hinterlegten Prozess- und Aktivitätensvorlagen¹ in (disjunkte) Kategorien einteilen zu können. Die für eine einzelne Kategorie festgelegten Rechte können dann für alle Elemente dieser Kategorie angewendet werden.

In Umgebungen, wie z. B. Krankenhäusern, gibt es bis zu mehrere hundert Tätigkeiten, so dass die Kategorien aus Gründen der besseren Übersichtlichkeit hierarchisch angeordnet werden sollten. Ein Beispiel zeigt Abb. 7.9.

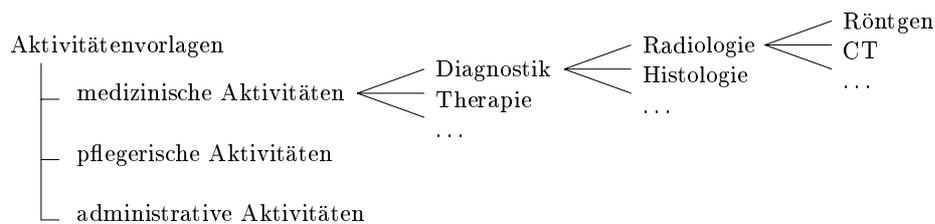


Abb. 7.9: Beispiel von hierarchisch angeordneten Aktivitätskategorien

7.3.1.4 Definition von Einfügerechten

Im Folgenden wird auf die Fragestellungen 1 – 3 aus Abschnitt 7.3.1.1 Bezug genommen.

Aspekt 1: Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) aus dem Repository darf ein bestimmter Benutzer generell einfügen?

Um festlegen zu können, von welchen Benutzern bestimmte Aktivitäten generell eingefügt werden dürfen, muss für jede Aktivitätsvorlage eine entsprechende Regelung getroffen werden. Dies ist in ADEPT auf der Grundlage der in Kap. 4.2.3 beschriebenen Bearbeiterzuordnungsausdrücke möglich. Ordnet man also z. B. einer Aktivitätsvorlage „Patient aufklären“ den Bearbeiterzuordnungsausdruck „R = Arzt“ zu, so sind alle Benutzer mit dieser Rolle prinzipiell berechtigt, neue Arbeitsschritte mit dieser Vorlage in Prozesse einzufügen.

Wie im vorangegangenen Abschnitt bemerkt, wäre ein Ansatz, bei dem Bearbeiterzuordnungsausdrücke für jede Aktivitätsvorlage explizit festgelegt werden müssen, sehr wartungsaufwendig. Aus diesem Grund müssen entsprechende Berechtigungen auch auf der Ebene von Aktivitätskategorien festgelegt werden können. Ein entsprechender Ansatz

¹Einer Aktivitätsvorlage ist entweder eine aufrufbare Anwendungskomponente zugeordnet (elementare Aktivität) oder aber einer Prozessvorlage (komplexe Aktivität).

hierzu wird im Folgenden anhand von Beispielen vorgestellt.

Gegeben seien die im Repository hinterlegten Aktivitätenkategorien und -vorlagen sowie deren hierarchische Beziehungen (vgl. Abb. 7.9). Der Einfachheit halber wird in der Folge vom *Aktivitätenbaum* gesprochen. Seine Nicht-Blätter entsprechen den Aktivitätenkategorien und seine Blätter den Aktivitätsvorlagen. Eine Aktivitätenkategorie umfasst in der Regel mehrere Sub-Aktivitätenkategorien und/oder Aktivitätsvorlagen.

Die Festlegung der Rechte, von wem welche Aktivitätsvorlagen generell in Prozesse eingefügt werden dürfen, kann dann nach folgendem Prinzip erfolgen: Einem Nicht-Blattknoten kann ein Bearbeiterzuordnungsausdruck zugewiesen werden, der die Rechte für diesen Knoten und alle Kindknoten festlegt, es sei denn für einen Kindknoten wird diese Bearbeiterformel explizit überschrieben. In der Regel genügt es also, Rechte für einzelne Kategorien festzulegen, diese sind dann für alle untergeordneten Kategorien und Vorlagen gültig.

Ein einfaches Beispiel zeigt Abb. 7.10. Die Berechtigungen zum Einfügen von Aktivitäten sind hier auf der Ebene der Aktivitätenkategorien $A_1 - A_3$ festgelegt. Sie gelten somit auch auf den untergeordneten Ebenen, wie den zugehörigen Aktivitätsvorlagen oder weiteren Aktivitätenkategorien. Im Beispiel gilt die Berechtigung „R = Arzt“ für die Aktivitätenkategorie A_1 und alle ihr untergeordneten Aktivitätenkategorien und -vorlagen. Ein Benutzer der Rolle *Arzt* darf somit beim Einfügen von Prozessschritten auf die Aktivitätsvorlagen a_{11} , a_{12} , a_{13} , ... Bezug nehmen. Es muss also nicht für jede dieser Vorlagen eine entsprechende Berechtigung definiert werden. Durch diese kompakte Definition der Rechte ist der Aufwand, die Berechtigungen zu ändern oder an andere Umgebungsvariablen anzupassen, relativ gering und die Pflegebarkeit der Rechte erhöht sich erheblich.

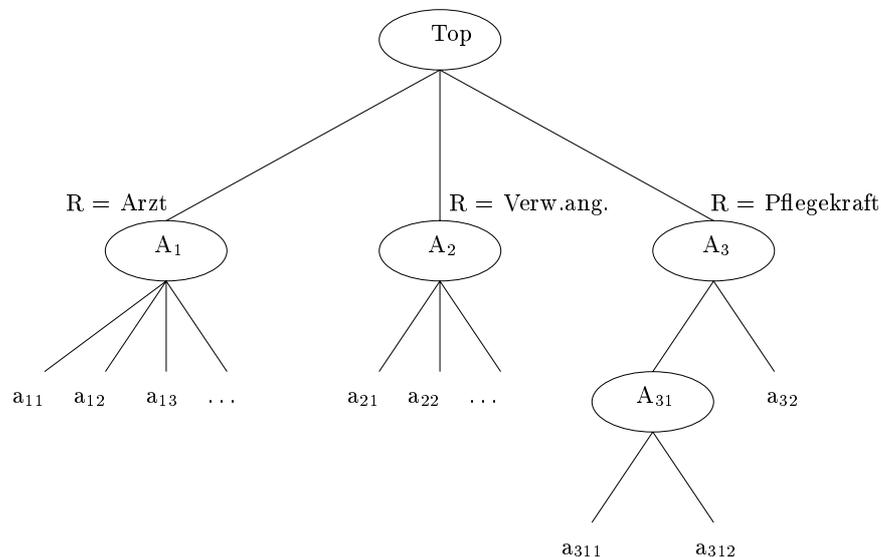


Abb. 7.10: Beispiel für einen einfachen Aktivitätsbaum mit den zugehörigen Berechtigungen

In Abb. 7.10 sind die Berechtigungen nur auf einer Ebene, den Aktivitätenkategorien der obersten Stufe, definiert. Es ist aber auch möglich für eine bestimmte Aktivitätenkategorie oder -vorlage eine speziellere Berechtigung als auf den übergeordneten Ebenen festzulegen. In den Abb. 7.11 und 7.12 sind zwei Beispiele für das Überschreiben von Rechten auf einer untergeordneten Ebene dargestellt.

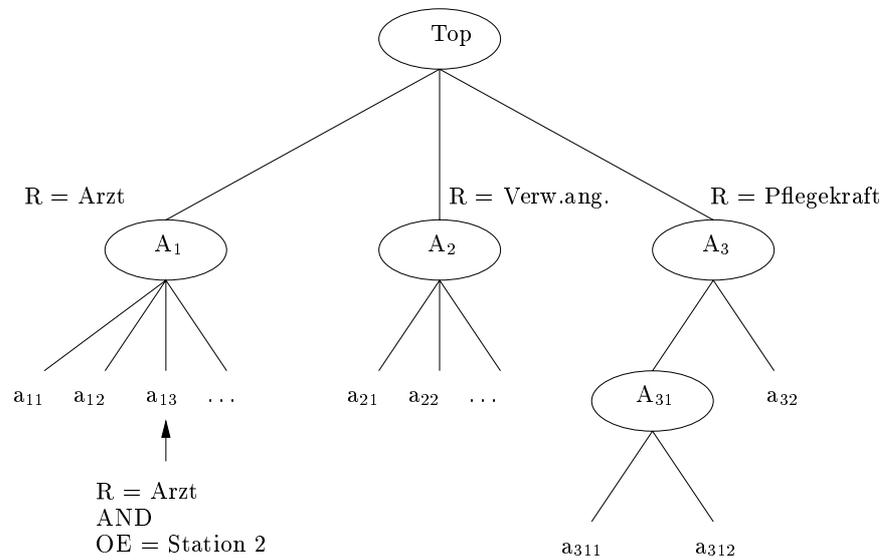


Abb. 7.11: Beispiel für das Überschreiben von Rechten auf untergeordneter Ebene durch einen spezielleren Bearbeiterzuordnungsausdruck

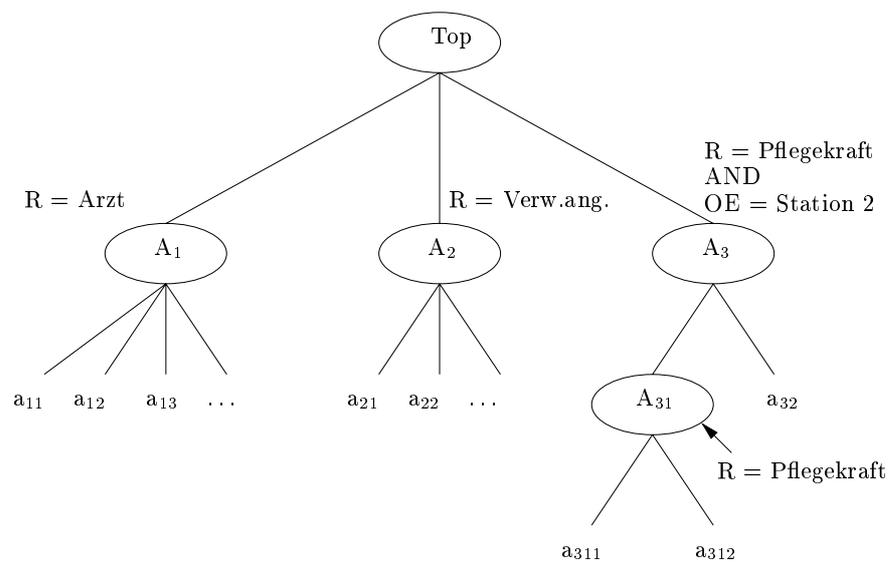


Abb. 7.12: Beispiel für das Überschreiben von Rechten durch eine allgemeinere Bearbeiterformel auf untergeordneter Ebene

In Abb. 7.11 gilt die für die Aktivitätenkategorie A_1 festgelegte Berechtigung der Rolle *Arzt* auch für die untergeordneten Aktivitätenvorlagen a_{11} und a_{12} , nicht jedoch für a_{13} . Für die letztgenannte Aktivitätenvorlage wurde die Berechtigung überschrieben. Diese Aktivität darf nur der Benutzer einfügen, der sowohl die Rolle *Arzt* einnimmt als auch der Organisationseinheit *Station 2* angehört, d.h. die Menge der berechtigten Akteure wird gegenüber der Festlegung für die Kategorie A_1 weiter eingeschränkt.

In Abb. 7.12 gilt die für die Aktivitätenkategorie A_3 festgelegte Berechtigung „*R = Pflegekraft AND OE = Station 2*“ nur für die Aktivitätenvorlage a_{32} . Sie gilt nicht für die untergeordnete Aktivitätenkategorie A_{31} und deren Aktivitätenvorlagen a_{311} und a_{312} , da für die Aktivitätenkategorie A_{31} die allgemeinere Berechtigung „*R = Pflegekraft*“ festgelegt wurde. D. h. Benutzer, die die Rolle *Pflegekraft* einnehmen und der Organisationseinheit *Station 2* angehören, dürfen lediglich die Aktivitätenvorlage a_{32} in eine laufende Instanz einfügen. Benutzer, die die Rolle *Pflegekraft* einnehmen, dürfen die Aktivitätenvorlagen a_{311} und a_{312} in eine Instanz einfügen. Hier wird also die Menge der berechtigten Akteure auf untergeordneten Ebenen vergrößert.

Soll nun für einen konkreten Benutzer festgestellt werden, auf welche Aktivitätenvorlagen er beim Einfügen von Schritten Bezug nehmen darf, so muss der Aktivitätenbaum (vollständig) durchlaufen werden. Bei effizienter Implementierung wird z. B. an einem Nicht-Blattknoten vermerkt, ob ein Nachfolgeknoten die Bearbeiterformel überschreibt oder nicht. Ist ersteres der Fall, müssen für untergeordnete Kategorien oder Vorlagen keine weiteren Prüfungen erfolgen.

Aspekt 2: Welche Art von Aktivitäten (bzw. Aktivitätenvorlagen) dürfen in die Instanzen einer bestimmten Prozessvorlage eingefügt werden?

Vorangehend wurden Berechtigungskonzepte vorgestellt, die regeln, welche Art von Aktivitäten bzw. Aktivitätenvorlagen ein bestimmter Benutzer generell einfügen darf. Entsprechende Festlegungen erfolgen (noch) ohne Bezugnahme auf konkrete Prozessstypen bzw. -vorlagen. Wie eingangs in Abschnitt 7.3.1.4 diskutiert, reicht dies jedoch noch nicht aus, um festzulegen, welche Aktivitäten in welche Prozessinstanzen eingefügt werden dürfen.

Abhängig vom Prozessstypen muss es auch möglich sein, die Art von Aktivitäten, die in zugehörigen Instanzen eingefügt werden dürfen, zu begrenzen. Aus diesem Grund muss für (einzelne) Prozessvorlagen festgelegt werden können, auf welche Aktivitätenvorlagen beim Einfügen von Schritten Bezug genommen werden darf. Dazu muss auf diesem Wege eine Obermenge von Aktivitätenvorlagen aus dem Repository vereinbart werden können.

Bei der Festlegung entsprechender Berechtigungen muss wieder darauf geachtet werden, dass die resultierenden Berechtigungen mit vertretbarem Aufwand pflegbar sind und sich gegenüber Änderungen im Repository (z. B. durch Hinzunahme/Entfernen von Vorlagen) robust zeigen. Dies legt wieder nahe, Prozess- und Aktivitätenkategorien bei der Rechtedefinition einzubeziehen, sowie sich dabei die hierarchischen Beziehungen zwischen Kategorien zu nutze zu machen.

Anhand eines einfachen Beispiels soll dieser Vorgang nachvollzogen werden. Die Grundidee dabei ist, Prozesskategorien und/oder -vorlagen, die im sogenannten *Prozessbaum* organisiert sind, entsprechende Aktivitätenvorlagen, auf die dann beim Einfügen von Prozessschritten Bezug genommen werden darf, zuzuordnen. Anstelle von einzelnen Aktivitätenvorlagen können auch Aktivitätenkategorien den Prozesskategorien bzw. -vorlagen zugewiesen werden. In diesem Fall gelten die entsprechenden Festlegungen dann für alle Aktivitätenvorlagen, die dieser

Aktivitätenkategorie bzw. den ihr untergeordneten Kategorien zugeordnet sind.

Ein Beispiel für die Zuordnung von Aktivitätenkategorien und -vorlagen zu Knoten des Prozessbaums zeigt Abb. 7.13. Die Festlegung von Berechtigungen erfolgt hier additiv. Gilt eine bestimmte Zuordnung für eine Prozesskategorie P , so gilt sie auch für alle untergeordneten Prozesskategorien und -vorlagen von P . Zusätzlich können auf untergeordneter Ebene den betreffenden Knoten des Prozessbaumes, falls gewünscht, weitere Aktivitätenkategorien oder -vorlagen zugewiesen werden. In Abb. 7.13 bedeutet das:

- $a_{21} \rightarrow Top'$: Diese Festlegung gilt für alle Prozesskategorien und -vorlagen, die der Prozesskategorie Top' (Wurzel des Prozessbaumes) untergeordnet sind. Folglich können in Prozessinstanzen mit beliebiger Prozessvorlage neue Aktivitäten mit der Vorlage a_{21} eingefügt werden.
- $A_2 \rightarrow P_1$: Diese Festlegung gilt für die Prozesskategorie P_1 und damit für alle Prozesskategorien und -vorlagen, die P_1 untergeordnet sind. Alle Aktivitätsvorlagen der Kategorie A_2 , sowie Aktivitätsvorlagen untergeordneter Kategorien (a_{21}, \dots) können in Prozessinstanzen eingefügt werden, wenn ihre Prozessvorlage einer der Prozesskategorien P_1, P_{11} oder P_{12} angehört. Im Beispiel sind dies die Vorlagen p_{111} und p_{112} .

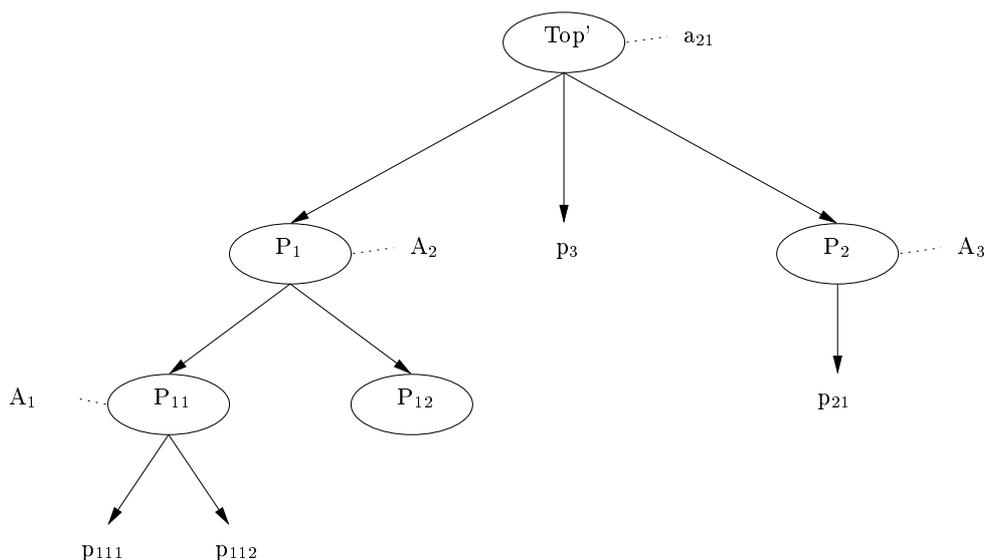


Abb. 7.13: Zuordnung von Aktivitätenkategorien und -vorlagen zu Knoten des Prozessbaumes

Wie bei dem auf Seite 86 behandelten Aspekt 1, sind die Aktivitätenkategorien und -vorlagen, die in Prozessinstanzen eingefügt werden dürfen, den Prozesskategorien und -vorlagen übersichtlich und kompakt zugeordnet. Werden in den Aktivitätenbaum neue Aktivitätsvorlagen eingefügt, müssen bei der Verwendung von Aktivitätenkategorien im Prozessbaum keine Änderungen vorgenommen werden. Das gleiche gilt für den Fall, dass neue Prozessvorlagen in den Prozessbaum eingefügt werden. Der Grund dafür ist, dass die Eigenschaften der übergeordneten Prozesskategorien übernommen werden.

Bei der Ermittlung der Aktivitätsvorlagen, auf die beim Einfügen eines Schrittes in eine Prozessinstanz Bezug genommen werden darf, kann wie folgt vorgegangen werden.

1. Ermittlung der zur abzuändernden Prozessinstanz gehörenden Prozessvorlage.
2. Bestimmung des Weges (im Prozessbaum) von dem entsprechenden Blattknoten der

Prozessvorlage zur Wurzel des Baumes. Für jeden Knoten auf diesem Weg muss dabei die entsprechende Aktivitätszuordnung berücksichtigt werden.

Bezogen auf das in Abb. 7.13 gegebene Beispiel ergibt sich bei einer gegebenen Prozessvorlage p_{112} der Weg zur Wurzel als $p_{112} \rightarrow P_{11} \rightarrow P_1 \rightarrow \text{Top}'$. Die relevanten Aktivitäten-kategorien und -vorlagen sind demnach A_1, A_2 und a_{21} (auf elementarer Ebene sind dies $a_{11}, a_{12}, \dots, a_{21}, \dots$).

Alternativ zu der in Abb. 7.13 gewählten Vorgehensweise (Zuordnung von Aktivitäten-kategorien/-vorlagen zu Knoten des Prozessbaumes) kann derselbe Sachverhalt auch in einer anderen (inversen) Sicht dargestellt werden, indem im Aktivitätenbaum den einzelnen Knoten entsprechende Prozesskategorien oder -vorlagen zugewiesen werden. Zwischen diesen beiden Sichten kann jederzeit „umgeschaltet“ werden.

Im Folgenden soll aus dem Prozessbaum aus Abb. 7.13 der zugehörige Aktivitätenbaum mit zugeordneten Prozesskategorien bzw. -vorlagen ermittelt werden. Der Aktivitätenbaum wird Schritt für Schritt aufgebaut, indem den einzelnen Aktivitätenvorlagen die zugeordneten Prozessvorlagen gegenübergestellt werden. Die Gegenüberstellung der Prozessvorlagen zu den Aktivitätenvorlagen ist in Tab. 7.1 dargestellt.

Aktivitätenvorlagen	Prozessvorlagen
a_{11}	P_{111}, P_{112}
a_{12}	P_{111}, P_{112}
a_{13}	P_{111}, P_{112}
a_{21}	$P_3, P_{21}, P_{111}, P_{112}$
a_{22}	P_{111}, P_{112}
a_{32}	P_{21}
a_{311}	P_{21}
a_{312}	P_{21}

Tabelle 7.1: Zuordnung von Prozessvorlagen zu Aktivitätenvorlagen

Anschließend werden die verschiedenen Prozessvorlagen in die zugehörigen Prozesskategorien umgewandelt und den Aktivitätenvorlagen gegenübergestellt. Dies ist in Tab. 7.2 dargestellt.

Aktivitätenvorlagen	Prozesskategorien
a_{11}	P_{11}
a_{12}	P_{11}
a_{13}	P_{11}
a_{21}	$P_1, P_2, P_{11}, P_{12}, \text{Top}'$
a_{22}	P_1, P_{11}, P_{12}
a_{32}	P_2
a_{311}	P_2
a_{312}	P_2

Tabelle 7.2: Zuordnung von Prozesskategorien zu Aktivitätenvorlagen

Zu guter Letzt werden die Aktivitätenvorlagen, sofern möglich, in die zugehörigen Aktivitäten-kategorien umgewandelt. Die resultierende Zuordnung der Prozesskategorien zu den

Aktivitätenkategorien bzw. -vorlagen ist im Aktivitätenbaum aus Abb. 7.14 dargestellt. Zur Bestimmung der Aktivitätenvorlage, auf die beim Einfügen eines Schrittes in eine Prozessinstanz Bezug genommen werden darf, wird wie bei der ersten Sichtweise vorgegangen.

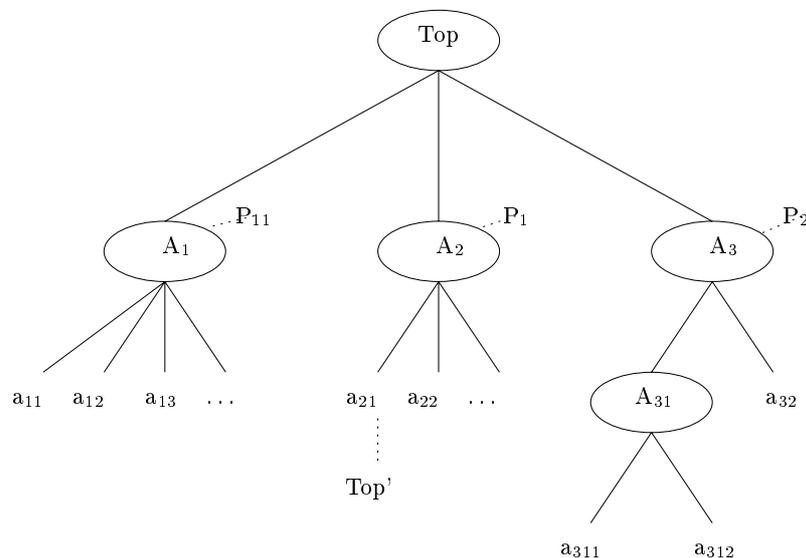


Abb. 7.14: Zuordnung der Prozesskategorien zu den Aktivitätenkategorien bzw. -vorlagen im Aktivitätenbaum als inverse Darstellung des Prozessbaumes aus Abb. 7.13

Auf den ersten Blick erscheint die zweite Sichtweise überflüssig, da mit den Zuordnungen der Aktivitätenkategorien zu Knoten des Prozessbaumes die grundlegende Frage beantwortet werden kann, welche Aktivität in welche Prozessinstanz eingefügt werden kann. Durch die Verwendung der inversen Sicht können jedoch bestimmte Arten von Fragestellungen gezielter beantwortet werden.

Aspekt 3: Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) in eine bestimmte Instanz einfügen?

Wie eingangs von Abschnitt 7.3.1.1 diskutiert, reichen Aspekt 1 und 2 noch nicht aus, um zu ermitteln, ob ein bestimmter Benutzer eine Aktivität in eine Prozessinstanz einfügen darf. Schließlich kann es zu einer Prozessvorlage mehrere Instanzen geben, bei denen der Benutzer in der einen Instanz Aktivitäten einfügen darf und in anderen nicht.

Im wesentlichen wird bei der Bestimmung der Benutzer, die eine bestimmte Aktivität(envorlage) in eine bestimmte Instanz einfügen dürfen, analog zur Vorgehensweise bei der Diskussion des ersten Aspekts auf Seite 86 vorgegangen, jetzt jedoch bezogen auf die Prozesskategorien und -vorlagen. Dazu werden den Knoten im Prozessbaum Mitarbeiterzuordnungsausdrücke zugewiesen, die angeben, welche Benutzer Aktivitäten in Prozessinstanzen der zugehörigen Prozessvorlage einfügen dürfen.

Abb. 7.15 skizziert ein Beispiel eines solchen Prozessbaumes, dessen Knoten Mitarbeiterzuordnungsausdrücke zugeordnet sind. Der Mitarbeiterzuordnungsausdruck „ $R = \text{Arzt AND OE} = \text{BEHANDELNDE_EINHEIT}$ “ ist gültig für die Prozesskategorie P_1 und für die ihr untergeordneten Prozesskategorien und -vorlagen P_{11} , P_{12} und p_{111} . Sie ist nicht gültig für die Prozessvorlage p_{112} , da für sie ein speziellerer Mitarbeiterzuordnungsausdruck festgelegt wurde, der den übergeordneten überschreibt.

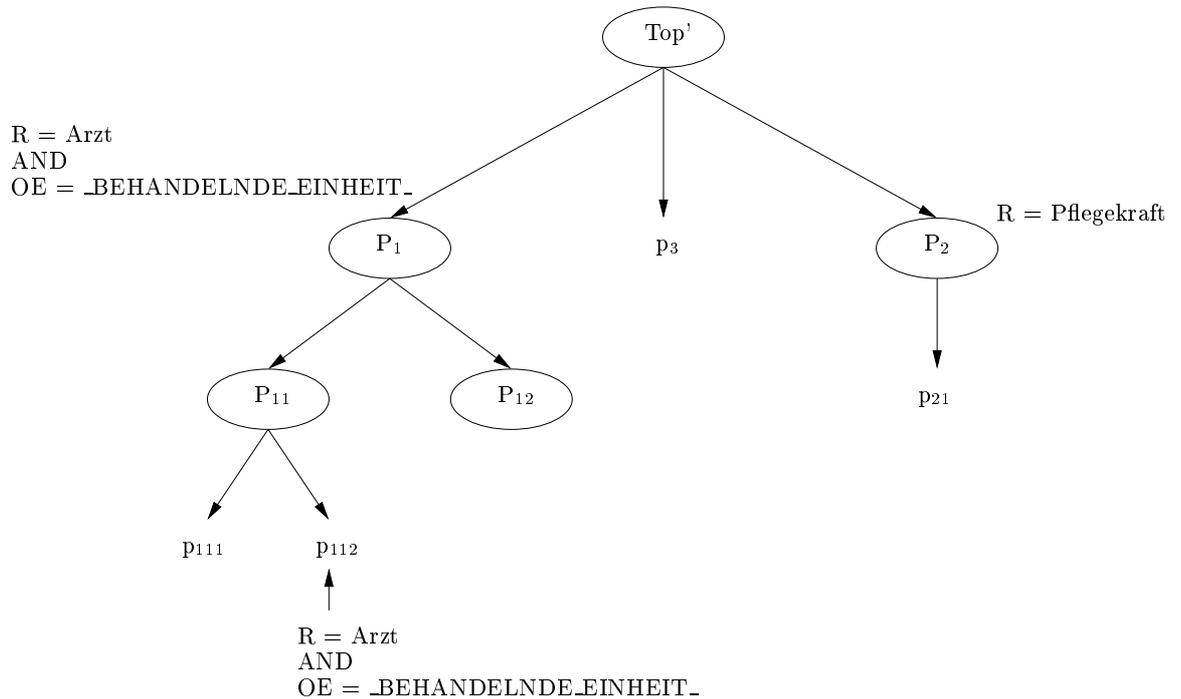


Abb. 7.15: Zuordnung von Mitarbeiterformeln zu einem Prozessbaum

Ein spezielles Problem in diesem Zusammenhang ist, dass die einem Knoten zugeordnete Mitarbeiterformel variabel gestaltet werden können muss, da zwei Instanzen in unterschiedlichen Organisationseinheiten auf derselben Prozessvorlage beruhen können, aber Personen, die an Instanz A auf Station 1 beteiligt sind, dürfen an Instanz B auf Station 2 keine Schritte einfügen. Die benötigte Variabilität kann über eine konkrete Variable erreicht werden, in der beispielsweise die Organisationseinheit gespeichert wird. Im Beispiel aus Abb. 7.15 referenziert die dem Knoten P_1 zugeordnete Mitarbeiterformel eine solche Prozessvariable (`_BEHANDELNDE_EINHEIT_`). Diese muss in Prozessinstanzen mit einem Wert belegt sein, um entsprechende Prüfungen vornehmen zu können. Die Mitarbeiterformel muss diese Variable referenzieren können, da ihre Belegung nicht von Anfang an feststeht oder sie aber verändert wird (z. B. durch Verlegung eines Patienten). Für eine Prozessinstanz soll beispielsweise geprüft werden, ob eine Berechtigung zum Einfügen eines Schrittes vorliegt. Die eben erwähnte Variable ist aber noch nicht geschrieben, d. h. der Mitarbeiterzuordnungsdruck kann nicht ausgewertet werden. Daraus ergibt sich, dass keine Änderung durchgeführt werden darf.

Abb. 7.16 skizziert ein Beispiel zur Verwendung dieser Variablen. In x wird die Organisationseinheit des Bearbeiters der Aktivität A gespeichert, so dass zu jeder Zeit ersichtlich ist, welcher Organisationseinheit Mitarbeiter angehören müssen, wenn Sie in einer laufenden Prozessinstanz Schritte einfügen wollen.

7.3.1.5 Konkretes Beispiel zum dynamischen Einfügen von Aktivitäten

Im Folgenden ist die Vorgehensweise zur Bestimmung der Aktivitäten, die ein bestimmter Benutzer in eine bestimmte Prozessinstanz einfügen darf, an einem konkreten Beispiel

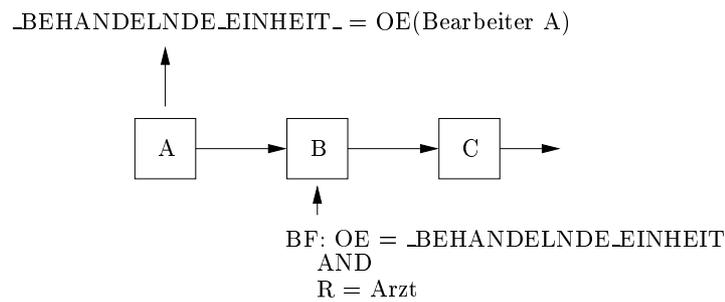


Abb. 7.16: Beispiel zur Verwendung einer Variable zur Steigerung der Variabilität von Bearbeiterformeln

dargestellt.

Gegeben ist der Ablauf einer radiologischen Untersuchung aus Kap. 3.1. Der untersuchende Arzt der Radiologie möchte eine zusätzliche Aktivität „Patient aufklären“ im Ablauf einer CT-Untersuchung einfügen. Für dieses Beispiel ist der Aktivitätenbaum aus Abb. 7.17 skizziert. Lediglich der Bereich der ärztlichen Tätigkeiten ist genauer beschrieben, da in diesem Beispiel nur der ärztliche Bereich betrachtet werden soll. Hier sind auch direkt Bearbeiterzuordnungsausdrücke angegeben, um Berechtigungen für die jeweiligen Aktivitäten ausdrücken zu können. Natürlich sind im Normalfall auch für die weiteren Bereiche Aktivitäten- und Prozessvorlagen sowie zugehörige Bearbeiterzuordnungsausdrücke spezifiziert. Der Einfachheit halber wird hier jedoch auf sie verzichtet.

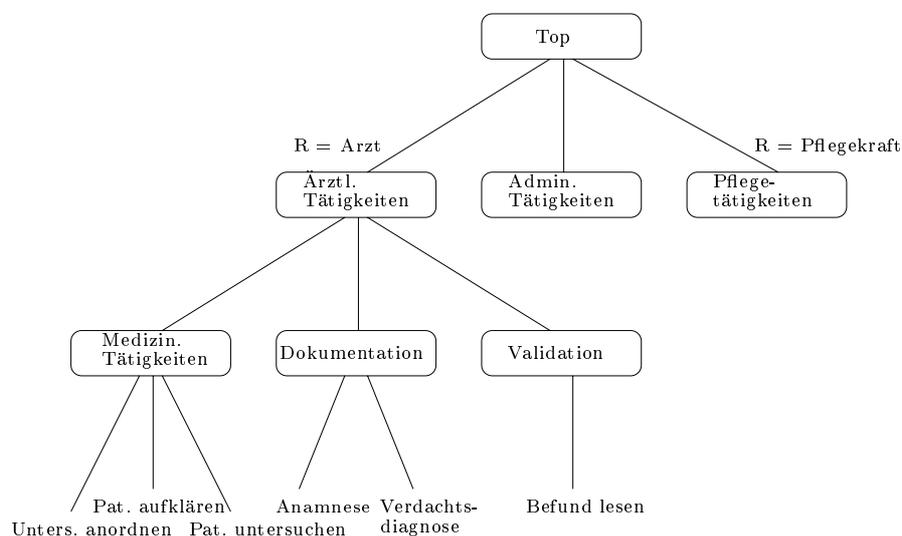


Abb. 7.17: Aktivitätenbaum im klinischen Kontext

Zu diesem Aktivitätenbaum ist der Prozessbaum aus Abb. 7.18 gegeben. Zu den verschiedenen Prozesskategorien sind ebenfalls die zugehörigen Aktivitätenkategorien bzw. -vorlagen angegeben, die in eine Instanz eingefügt werden dürfen. Im Beispiel sind dies die Aktivitätenkategorien *Medizin. Tätigkeiten* und *Admin. Tätigkeiten*.

In Abb. 7.19 ist der zugehörige Aktivitätenbaum dargestellt, in dem den Aktivitätenkatego-

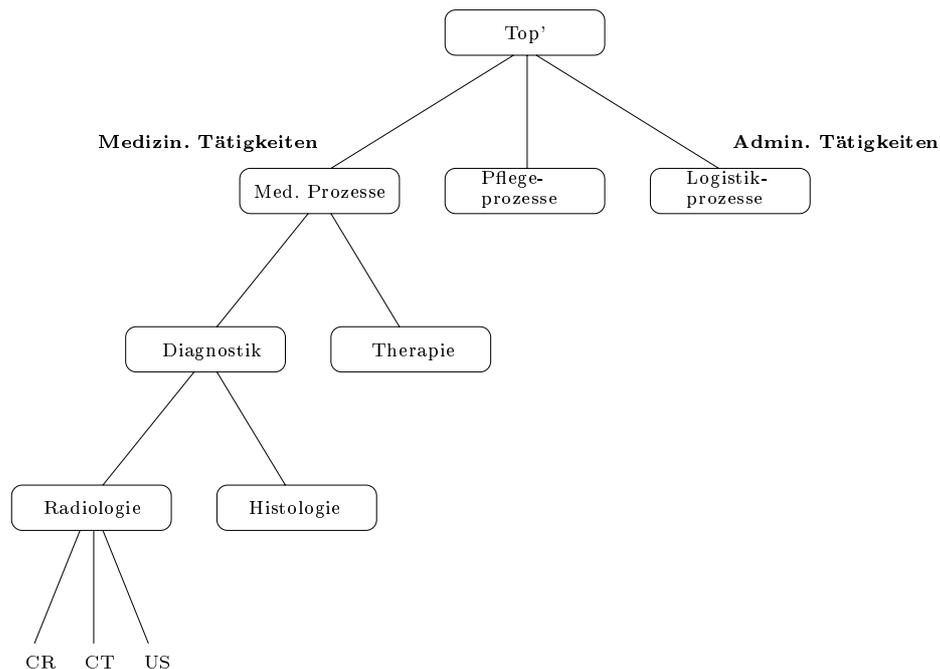


Abb. 7.18: Zugehöriger Prozessbaum mit zugeordneten Aktivitätenkategorien

rien bzw. -vorlagen Prozesskategorien zugeordnet sind. Im Beispiel sind dies die Prozesskategorien *Med. Prozesse* und *Logistikprozesse*.

Desweiteren ist in Abb. 7.20 der Prozessbaum mit den zugehörigen Bearbeiterzuordnungsdrücken skizziert.

Bei der Bestimmung, ob ein Benutzer eine konkrete Aktivität in eine bestimmte Prozessinstanz einfügen darf, wird anhand der in Abschnitt 7.3.1.4 diskutierten Aspekte auf die folgende Art vorgegangen:

1. Ermittlung aller generell von einem Benutzer mit der Rolle *Arzt* einfügbaren Aktivitätenvorlagen anhand des Aktivitätenbaums aus Abb. 7.17. Daraus ergibt sich eine Menge von Aktivitätenvorlagen, die dieser Benutzer – unabhängig von einer bestimmten Prozessinstanz – generell einfügen darf. Darunter fallen alle Aktivitätenvorlagen, die der Aktivitätenkategorie „Ärztl. Tätigkeiten“ untergeordnet sind. (Aspekt 1)
2. Anschließend wird einer der folgenden zwei Wege zur Bestimmung der Aktivitäten, die in eine Prozessinstanz eingefügt werden dürfen, verfolgt.
 - (a) Ermittlung der zur Prozessinstanz *CT Untersuchung* zugehörigen Prozessvorlage *CT*. Dann wird der Weg von diesem Blatt zur Wurzel des Prozessbaumes verfolgt. Im Beispiel ist dieser Weg also beschrieben durch:
 „CT“ → „Radiologie“ → „Diagnostik“ → „Med. Prozesse“ → „Top’ “.
 Auf dem Weg vom Blatt zur Wurzel wird für jeden Knoten die Aktivitätszuordnung berücksichtigt. Hier ist der Prozesskategorie „Med. Prozesse“ die Aktivitätenkategorie „Medizin. Tätigkeiten“ zugeordnet.
 Der Benutzer dürfte also in Prozessinstanzen, die aus Prozessvorlagen der Prozesskategorie „Med. Prozesse“ hervorgehen, Aktivitäten einfügen, die der Aktivitätenkategorie „Medizin. Tätigkeiten“ untergeordnet sind. (Aspekt 2)

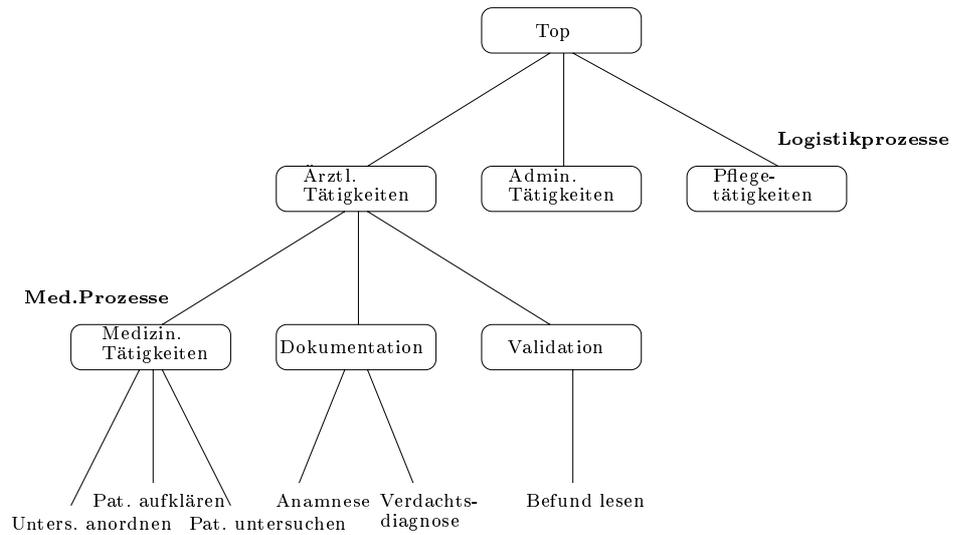


Abb. 7.19: Zugehöriger Aktivitätenbaum mit zugeordneten Prozesskategorien

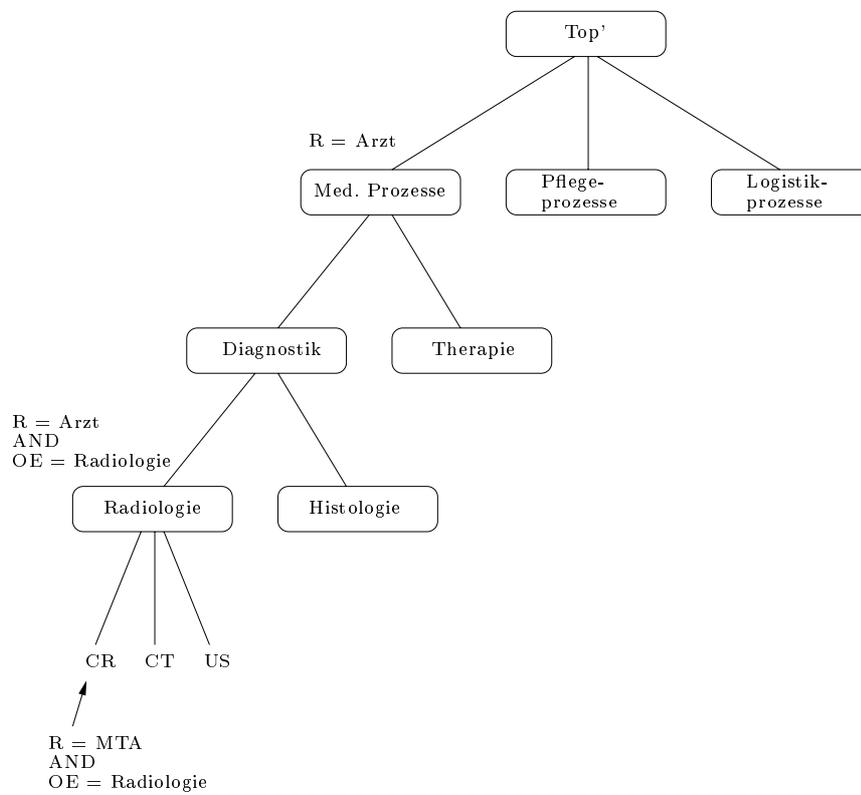


Abb. 7.20: Zugehöriger Prozessbaum aus dem klinischen Kontext

- (b) Ermittlung der zur Aktivitätenvorlage *Pat. aufklären* übergeordneten Aktivitäten-kategorie im Aktivitätenbaum aus Abb. 7.19, und Ermittlung eines Weges von diesem Vorlagenblatt zur Wurzel des Aktivitätenbaumes. Im Beispiel ist dieser Weg beschrieben durch:

„Pat. aufklären“ \rightarrow „Medizin. Tätigkeiten“ \rightarrow „Ärztl. Tätigkeiten“ \rightarrow „Top“.

Auf dem Weg zur Wurzel wird die Zuordnung der Prozesskategorien und/oder -vorlagen zu Knoten des Aktivitätenbaumes berücksichtigt. Hier ist der Aktivitätenkategorie „Medizin. Tätigkeiten“ die Prozesskategorie „Med. Prozesse“ zugeordnet.

Der Benutzer dürfte also die Aktivitäten der Aktivitätenkategorie „Medizin. Tätigkeiten“ in Instanzen einfügen, die der Prozesskategorie „Med. Prozesse“ untergeordnet sind. (Aspekt 2, inverse Darstellung)

3. Zur konkreten Bestimmung, ob ein Benutzer mit der Rolle *Arzt* einen Schritt in eine bestimmte Prozessinstanz einfügen darf oder nicht, werden im Prozessbaum aus Abb. 7.20 die zugehörigen Bearbeiterzuordnungsausdrücke zur Prozessvorlage *CT* ermittelt. Dazu wird von dem Blatt der Prozessvorlage ein Weg zur Wurzel des Prozessbaumes bestimmt. Dieser Weg ist gegeben durch:

„CT“ \rightarrow „Radiologie“ \rightarrow „Diagnostik“ \rightarrow „Med. Prozesse“ \rightarrow „Top“.

Auf diesem Weg werden die den einzelnen Knoten zugeordneten Bearbeiterzuordnungsausdrücke berücksichtigt. In diesem Fall ist der Prozesskategorie „Med. Prozesse“ der Bearbeiterzuordnungsausdruck „R = Arzt“ zugeordnet. Dieser wird jedoch durch den Bearbeiterzuordnungsausdruck „R = Arzt AND OE = Radiologie“ der Prozesskategorie „Radiologie“ überschrieben. Daraus ergibt sich, dass der letztgenannte Bearbeiterzuordnungsausdruck berücksichtigt werden muss. (Aspekt 3)

Im Beispiel will der die Untersuchung durchführende Arzt der Radiologie die Aktivität „Pat. aufklären“ einfügen. Nach der soeben beschriebenen Vorgehensweise dürfte der Arzt diesen Schritt in die Prozessinstanz einfügen.

7.3.1.6 Abschließende Bemerkungen

Wie die in diesem Abschnitt vorgestellten Berechtigungskonzepten in einem PMS konkret nutzbar sind, hängt nicht zuletzt von der Implementierung des Systems ab. Möchte ein Benutzer eine neue Tätigkeit in eine Prozessinstanz einfügen, sind unterschiedliche Vorgehensweisen denkbar.

- Dem Benutzer werden zuerst alle im Repository hinterlegten Aktivitätenkategorien und -vorlagen zur Auswahl angeboten, unabhängig davon, ob von ihm entsprechende Aktivitäten eingefügt werden dürfen oder nicht. Nach Auswahl einer konkreten Aktivitätenvorlage erfolgt dann eine Überprüfung gemäß der Aspekte 1 – 3 (A-posteriori Überprüfung).
- Vorab werden, unter Nutzung der Aspekte 1 – 3, alle Aktivitätenvorlagen ermittelt, auf die der Benutzer beim Einfügen eines neuen Schrittes in eine gegebene Prozessinstanz Bezug nehmen darf. D. h. die Überprüfung erfolgt hier vor der Auswahl der entsprechenden Aktivitätenvorlage durch einen Benutzer (A-priori Überprüfung). Diese Variante garantiert eine bessere Bedienbarkeit des Systems, ist jedoch auch aufwändiger zu implementieren.

7.3.2 Rechtekonzept für das dynamische Löschen von Aktivitäten

Im vorangehenden Abschnitt wurde gezeigt, welche Art von Berechtigungen zum dynamischen Einfügen von Schritten erforderlich sind und wie diese Rechte festgelegt werden können. Im Folgenden wird gezeigt, welche Arten von Berechtigungen zum dynamischen Löschen von Aktivitäten erforderlich sind. Ferner soll aufgezeigt werden, wie diese Rechte festgelegt werden können. Wie bei den Rechten zum dynamischen Einfügen von Schritten gelten auch beim dynamischen Löschen die eingangs dieser Arbeit geforderte Konsistenz und Pflegbarkeit als wichtiges Ziel der Rechtedefinition.

7.3.2.1 Erforderliche Rechte zur Kontrolle dynamischer Löschoperationen

Wie beim dynamischen Einfügen, wird auch beim dynamischen Löschen von Aktivitäten aus einer Prozessinstanz eine entsprechende Berechtigung für die Person benötigt, die eine Aktivität entfernen will. In diesem Zusammenhang sind folgende Fragestellungen von Bedeutung:

1. Welche Art von Aktivitäten (bzw. Aktivitätenvorlagen) darf ein bestimmter Benutzer generell löschen?

Es muss kontrollierbar sein, wer generell zum Löschen von Aktivitäten – unabhängig von der konkreten Instanz – berechtigt sein soll. Beispielsweise darf im klinischen Kontext eine Krankenschwester keine ärztlichen Tätigkeiten aus dem Prozessmodell löschen.

2. Welche Art von Aktivitäten (bzw. Aktivitätenvorlagen) dürfen aus den Instanzen einer bestimmten Prozessvorlage gelöscht werden?

Aspekt 1 ist unabhängig von konkreten Prozessvorlagen. Abhängig von der Kategorie einer Prozessvorlage muss es aber auch möglich sein, die Art von Aktivitäten, die in den entsprechenden Instanzen gelöscht werden dürfen, zu begrenzen. D.h. es muss auf der Ebene der Prozessvorlagen festgelegt werden können, auf welche Aktivitätenvorlagen beim Löschen Bezug genommen werden darf.

3. Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) aus Instanzen einer bestimmten Prozessvorlage löschen?

Die Bedingungen 1 und 2 reichen allein noch nicht aus, um zu ermitteln, ob ein bestimmter Benutzer eine bestimmte Aktivität aus einer bestimmten Prozessinstanz löschen darf. Zu einer bestimmten Prozessvorlage kann es Instanzen geben, in denen ein bestimmter Benutzer eine Aktivität X löschen darf, während er dies für eine andere Instanz nicht darf. Im Kontext des Krankenhausalltags etwa dürfen nur solche Personen Schritte aus einer Prozessinstanz löschen, die der Organisationseinheit der behandelnden Station angehören, d.h. ein Mitarbeiter der Intensivstation darf keine Schritte in Prozessinstanzen der Chirurgischen Abteilung löschen.

4. Welche Aktivitätenknotten dürfen in einer bestimmten Prozessvorlage gelöscht werden?

Für eine konkrete Prozessvorlage muss für jeden einzelnen Knoten bzw. Schritt festlegbar sein, ob dieser zur Laufzeit einer Prozessinstanz gelöscht werden darf oder nicht. Es ist z. B. denkbar, dass es innerhalb einer Prozessvorlage zwei verschiedene Aktivitäten derselben Aktivitätenvorlage gibt, von der die eine löscher sein soll, die andere dagegen nicht. Beispielsweise sollte eine Steuerrückzahlung von zwei verschiedenen Personen

geprüft werden [BFA99]. Allerdings ist es in Ausnahmesituationen möglich, die Prüfung nur durch eine Person vornehmen zu lassen. In diesem Falle, müsste einer der parallel ablaufenden Prüfkategorien aus der Prozessinstanz entfernt werden.

Diese Fragestellungen sollen in der Folge aufgegriffen werden, um zu zeigen, wie in ADEPT entsprechende Berechtigungen festgelegt werden können.

7.3.2.2 Definition von Löschberechtigungen

Bei der Bestimmung der löschbaren Aktivitäten und der berechtigten Benutzer wird analog zur Vorgehensweise des dynamischen Einfügens von Aktivitäten vorgegangen. Im Folgenden soll daher nur kurz auf die Problematik der Berechtigungen im Zusammenhang mit dem Löschen von Aktivitäten eingegangen werden, wobei auf die Fragestellungen 1 – 4 aus Abschnitt 7.3.2.1 Bezug genommen wird. Weiterhin gelten die in den Abschnitten 7.3.1.2 und 7.3.1.3 gemachten Vereinfachungen und Kategorisierungen.

Aspekt 1: Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) darf ein bestimmter Benutzer generell löschen?

Um festzulegen, welche Aktivitäten generell von einem bestimmten Benutzer gelöscht werden dürfen, unabhängig von einer Prozessinstanz, müssen für die einzelnen Aktivitätsvorlagen bzw. auf übergeordneter Ebene den Aktivitätskategorien im Aktivitätsbaum entsprechende Bearbeiterzuordnungsausdrücke zugewiesen werden. Ein entsprechender Ansatz hierzu wird im Folgenden anhand von Beispielen vorgestellt.

Gegeben seien die im Repository hinterlegten Aktivitätskategorien und -vorlagen sowie deren hierarchische Beziehungen im Aktivitätsbaum aus Abb. 7.21. Für die Festlegung der Rechte gelten die gleichen Bedingungen wie sie in Abschnitt 7.3.1.4 beschrieben sind, d. h. für einzelne Kategorien definierte Berechtigungen sind auch auf untergeordneten Kategorien gültig und Berechtigungen können durch speziellere oder allgemeinere Bearbeiterformeln überschrieben werden.

Abb. 7.21 skizziert einen Aktivitätsbaum, dessen Knoten oder Blättern Bearbeiterformeln zugeordnet sind. Benutzer, die die Rolle *Arzt* besitzen, dürfen Aktivitäten löschen, deren Aktivitätsvorlage der Aktivitätskategorie A_1 oder der ihr untergeordneten Kategorie A_{11} angehören. Konkret dürfen sie Aktivitäten mit den Aktivitätsvorlagen a_{12} , a_{13} und a_{111} löschen. Für Instanzen, die aus der Aktivitätsvorlage a_{112} hervorgehen, muss der Benutzer zusätzlich zur Rolle *Arzt* der Organisationseinheit *Intensiv 1* angehören, da für diese Aktivitätsvorlage eine speziellere Bearbeiterformel spezifiziert ist. Diese speziellere Bearbeiterformel überschreibt die Berechtigung von Aktivitätskategorie A_1 .

Aspekt 2: Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) dürfen aus den Instanzen einer bestimmten Prozessvorlage gelöscht werden?

In diesem Abschnitt wurden bisher Berechtigungskonzepte vorgestellt, die regeln, welche Art von Aktivitäten bzw. Aktivitätsvorlagen ein Benutzer, unabhängig von einer konkreten Prozessinstanz, generell löschen darf. Wie bei den Berechtigungskonzepten für das Einfügen von Schritten, erfolgen diese Festlegungen auch beim Löschen noch ohne Bezugnahme auf konkrete Prozessvorlagen. Dies reicht jedoch nicht aus, um die Berechtigungen auf bestimmte Prozessinstanzen einzuschränken.

Zusätzlich zur Abhängigkeit von der Art der Aktivitätsvorlage (vgl. Aspekt 1) muss es

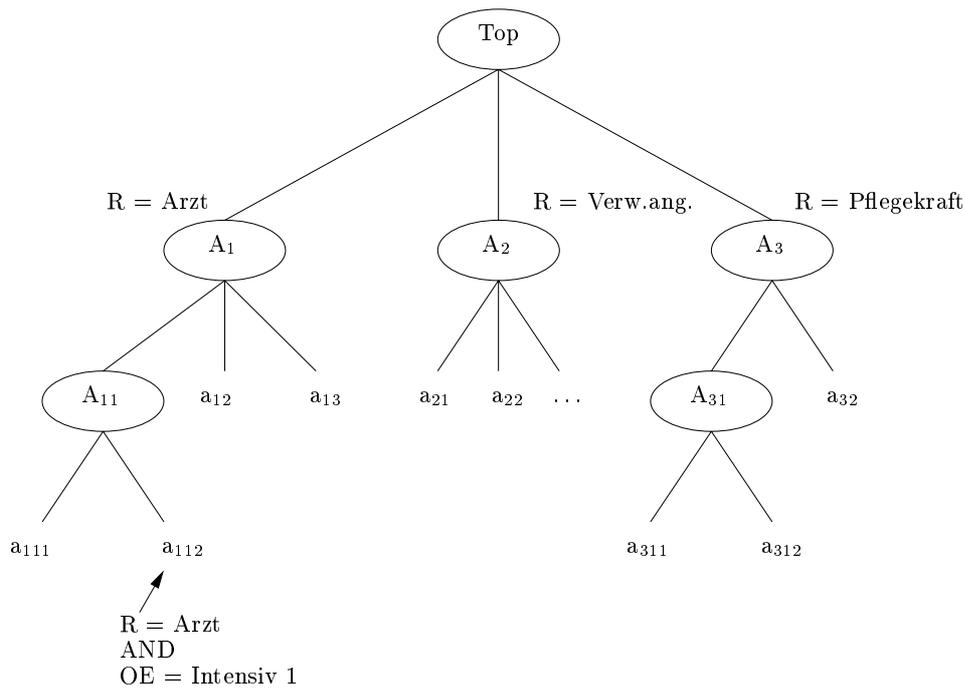


Abb. 7.21: Beispiel für allgemeine Berechtigungen zum Löschen von Aktivitäten

auch möglich sein die Art von Aktivitäten, die in einer Instanz gelöscht werden dürfen, zu begrenzen. Daher muss auch hier für Prozessvorlagen festgelegt werden können, auf welche Aktivitätsvorlagen beim Löschen Bezug genommen werden darf.

Bei der Festlegung entsprechender Berechtigungen muss wiederum darauf geachtet werden, dass die resultierenden Berechtigungen mit einem vertretbaren Aufwand pflegbar sind und sich robust gegenüber Änderungen im Repository verhalten. Dies legt wieder nahe, Prozess- und Aktivitätenkategorien bei der Rechtedefinition einzubeziehen und sich die hierarchischen Beziehungen zwischen den Kategorien zu nutze zu machen.

Im Folgenden soll anhand eines Beispiels dieser Vorgang verfolgt werden. Dazu werden im Prozessbaum den verschiedenen Prozesskategorien bzw. -vorlagen entsprechende Aktivitäten-kategorien oder -vorlagen zugeordnet, auf die beim Löschen von Aktivitäten Bezug genommen werden darf. Ein Beispiel für die Zuordnung von Aktivitäten-kategorien bzw. -vorlagen zu Prozesskategorien ist im Prozessbaum aus Abb. 7.22 skizziert.

Soll nun eine Aktivität aus einer bestimmten Prozessinstanz entfernt werden, wird wie beim Einfügen vorgegangen.

1. Ermittlung der zur abzuändernden Prozessinstanz gehörenden Prozessvorlage.
2. Bestimmung eines Weges von dem Knoten der Prozessvorlage zur Wurzel des Prozessbaumes. Für jeden Knoten auf diesem Weg muss dabei die entsprechende Aktivitätszuordnung berücksichtigt werden.

Bezogen auf das in Abb. 7.22 gegebene Beispiel ergibt sich bei einer gegebenen Prozessvorlage p_{12} der Weg zur Wurzel als $p_{12} \rightarrow P_1 \rightarrow \text{Top}$. Hier gibt es nur die eine Aktivitäten-kategorie A_2 zu berücksichtigen, die der Prozesskategorie P_1 zugeordnet ist. Das bedeutet in Prozessinstanzen, die aus Vorlagen der Prozesskategorie P_1 hervorgehen, dürfen alle Aktivitäten gelöscht werden, mit Vorlagen der Aktivitäten-kategorie A_2 .

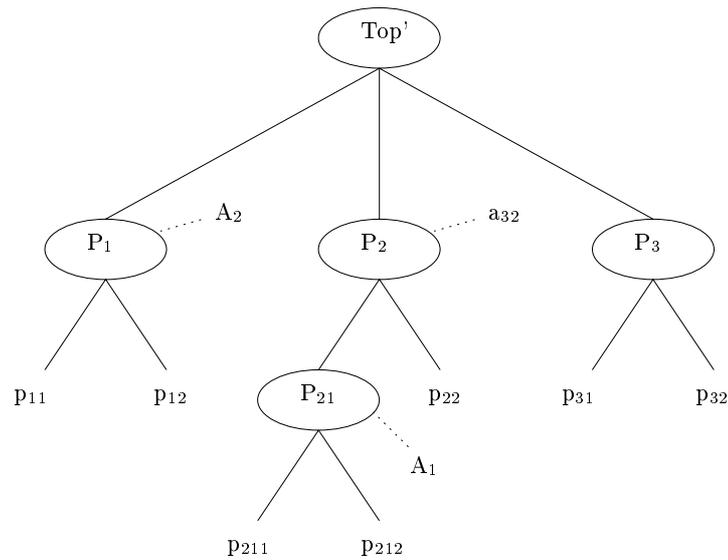


Abb. 7.22: Beispiel für einen Prozessbaum, dessen Knoten Aktivitätenkategorien bzw. -vorlagen zugeordnet sind

Alternativ zu der in Abb. 7.22 dargestellten Vorgehensweise der Zuordnung von Aktivitätenkategorien bzw. -vorlagen zu Knoten des Prozessbaumes kann derselbe Sachverhalt auch in der inversen Sicht dargestellt werden. In diesem Fall werden den Prozesskategorien bzw. -vorlagen den Knoten im Aktivitätenbaum zugewiesen. Zwischen diesen beiden Sichten kann jederzeit „umgeschaltet“ werden.

Bezogen auf den vorhergehenden Prozessbaum aus Abb. 7.22 ergibt sich der Aktivitätenbaum aus Abb. 7.23. Soll nun eine Aktivität aus einer Prozessinstanz gelöscht werden, wird der gleiche Weg beschritten, wie in der zuvor beschriebenen Sicht, lediglich wird hier der Aktivitätenbaum aus Abb. 7.23 verwendet.

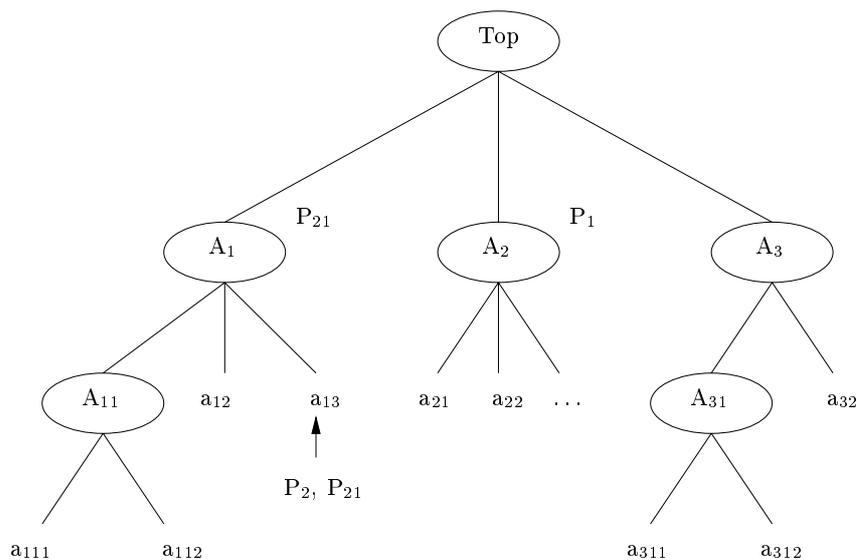


Abb. 7.23: Aktivitätenbaum mit zugeordneten Prozesskategorien als inverse Darstellung des Prozessbaumes aus Abb. 7.22

Aus dem Aktivitätenbaum aus Abb. 7.23 und dem Prozessbaum aus Abb. 7.22 lässt sich jetzt auch leicht ablesen, welche Aktivitäteninstanzen niemals gelöscht bzw. in welchen Prozessinstanzen niemals Aktivitäten entfernt werden dürfen. In beiden Fällen ist der entsprechenden Prozess-/Aktivitätenkategorie keine Aktivitäten- bzw. Prozesskategorie zugeordnet.

Wie bei den Berechtigungen zum Einfügen von Schritten sind die Berechtigungen zum Löschen von Schritten auf bestimmten Instanzen durch die Zuordnung von Aktivitätenkategorien zu Knoten des Prozessbaumes übersichtlich und kompakt definierbar. Werden neue Aktivitätsvorlagen in den Aktivitätenbaum eingefügt, müssen bei der Verwendung von Aktivitätenkategorien im Prozessbaum keine Änderungen vorgenommen werden. Das gleiche gilt für Prozessvorlagen, die in den Prozessbaum eingefügt werden, da die Eigenschaften der übergeordneten Kategorien übernommen werden.

Aspekt 3: Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) aus Instanzen einer bestimmten Prozessvorlage löschen?

Zur Bestimmung der Benutzer, die eine Aktivität aus einer gegebenen Instanz einer bestimmten Prozessvorlage generell löschen dürfen, wird analog zur Vorgehensweise des ersten Aspektes auf Seite 99 vorgegangen. Jedoch werden jetzt den Prozesskategorien im Prozessbaum entsprechende Bearbeiterzuordnungsausdrücke zugeordnet, die die zum Löschen berechtigten Benutzer spezifizieren. Natürlich kann auch hier auf der Vorlagenebene ein speziellerer Ausdruck als der auf der Kategorieebene angegebene Ausdruck definiert werden. Der Prozessbaum des vorhergehenden Beispiels hat z. B. die Berechtigungen aus Abb. 7.24.

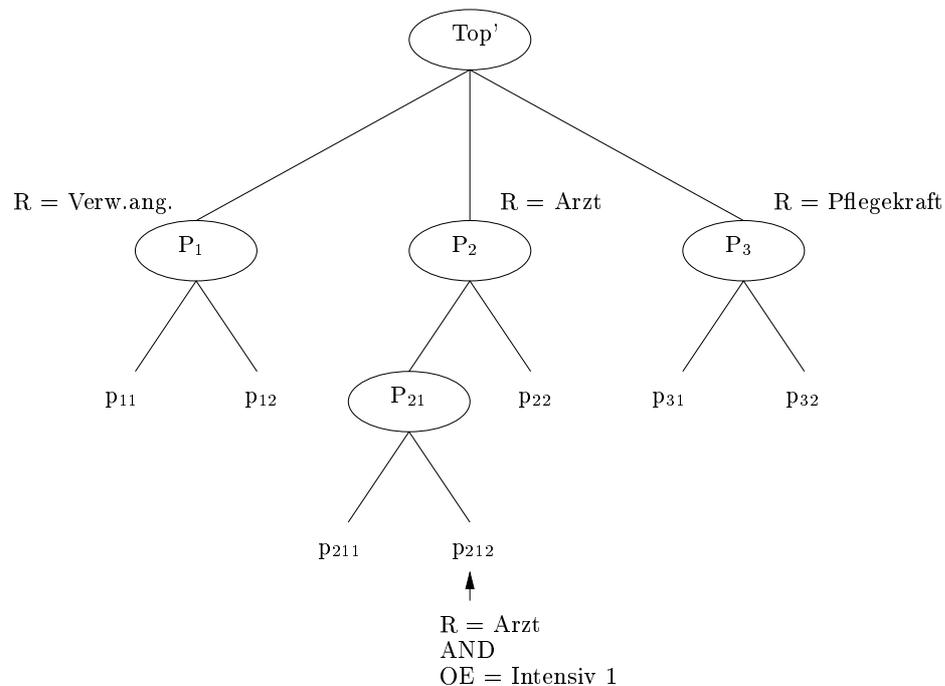


Abb. 7.24: Beispiel für Bearbeiterzuordnungen im Prozessbaum, um Löschoptionen durchführen zu können

Durch die zuvor beschriebenen Arten, Berechtigungen zum Löschen anhand der Prozess- bzw. Aktivitätenkategorien zu definieren, werden die Berechtigungen global festgelegt und müssen nicht für jede Aktivitäten- bzw. Prozessvorlage einzeln definiert werden. Jedoch ist es möglich, auf der Vorlagenebene speziellere Rechte festzulegen, die die übergeordneten Berechtigungen für den Einzelfall außer Kraft setzen. Durch diese Definition der Rechte an einer zentralen Stelle, wird ausgeschlossen, dass Rechte wiederholt oder dass für einzelne Vorlagen keine Berechtigungen festgelegt werden. So sind die Berechtigungen übersichtlich definiert und auch die Pflegbarkeit der Rechte ist gewährleistet.

Das Problem, dass zwei Instanzen derselben Prozessvorlage in unterschiedlichen Organisationseinheiten ablaufen können und Personen der einen Organisationseinheit nicht Instanzen der anderen Organisationseinheit verändern können dürfen, kann durch die Verwendung der beim dritten Aspekt der Einfügeberechtigungen auf Seite 92 beschriebenen konkreten Variable, umgangen werden.

Aspekt 4: Welche Aktivitätenknoten dürfen in einer bestimmten Prozessvorlage gelöscht werden?

Für jeden Aktivitätenknoten einer Prozessvorlage kann festgelegt werden, ob er aus einer Prozessinstanz gelöscht werden darf oder nicht. Gegebenenfalls ist diese Information auch unspezifiziert (Default). Will nun ein bestimmter Benutzer einen Prozessschritt löschen, so muss er hierfür nicht nur gemäß der Aspekte 1 – 3 autorisiert sein, sondern der betreffende Knoten muss zusätzlich als löscherbar markiert sein. Ist dies nicht der Fall, darf kein Benutzer den Schritt entfernen.

7.3.2.3 Konkretes Beispiel für das dynamische Löschen von Aktivitäten

Im Folgenden ist die Vorgehensweise zur Bestimmung von Aktivitäten, die ein bestimmter Benutzer aus einer bestimmten Prozessinstanz löschen darf, an einem konkreten Beispiel skizziert.

Gegeben ist der Ablauf einer radiologischen Untersuchung (vgl. Kap. 3.1). Eine Pflegekraft der behandelnden Station möchte die Aktivität „Patient vorbereiten“ entfernen.

Im Folgenden sind die zugehörigen Aktivitäten- und Prozessbäume gegeben. In Abb. 7.25 ist der Aktivitätenbaum mit den zugehörigen Bearbeiterzuordnungsausdrücken skizziert. Sowohl in diesem als auch allen folgenden Prozess- und Aktivitätenbäumen dieses Beispiels erfolgt der Einfachheit halber eine Beschränkung auf den Bereich der Pflege. Normalerweise sind auch die anderen Bereiche mit entsprechenden Kategorien und Vorlagen ausgefüllt. Der Übersichtlichkeit halber wurde hier jedoch darauf verzichtet.

In Abb. 7.26 sind die Prozesskategorien mit zugeordneten Aktivitätenkategorien skizziert, in Abb. 7.27 der inverse Baum. Hier finden sich die Aktivitätenkategorien und -vorlagen, denen Prozesskategorien bzw. -vorlagen zugeordnet sind. Die Berechtigungen, um in konkreten Prozesskategorien Änderungen vorzunehmen, sind in Abb. 7.28 dargestellt.

Anhand der in Kap. 7.3.2.2 diskutierten Aspekte wird bestimmt, ob ein konkreter Benutzer mit der Rolle *Pflegekraft* die Aktivität „Patient vorbereiten“ aus der laufenden Prozessinstanz entfernen darf.

1. Ermittlung aller generell vom Benutzer mit der Rolle *Pflegekraft* löscherbaren Aktivitätenvorlagen im Aktivitätenbaum aus Abb. 7.25 unter Berücksichtigung der Bearbeiterzuordnungsausdrücke. Daraus ergibt sich eine Menge von Aktivitätenvorlagen, die dieser Benutzer unabhängig von einer konkreten Prozessinstanz generell löschen darf. Im Bei-

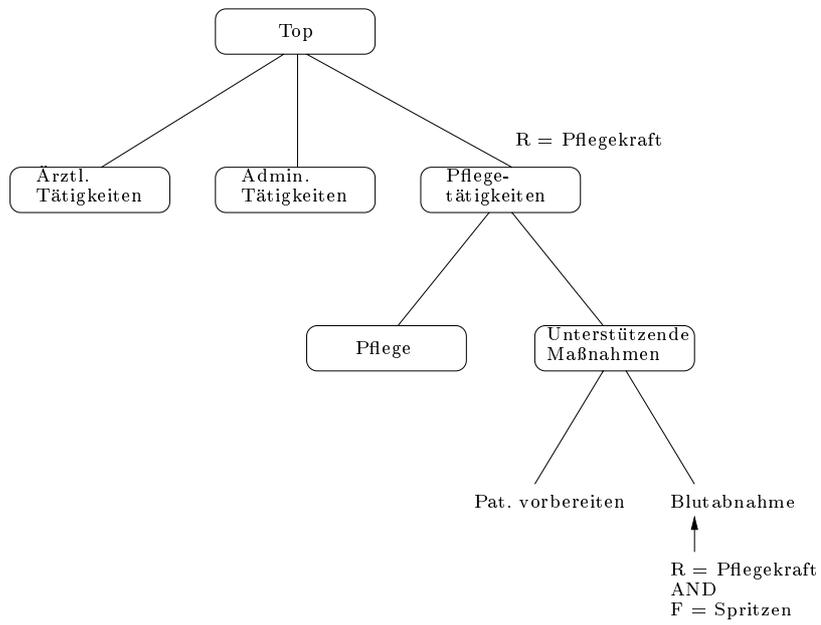


Abb. 7.25: Aktivitätenbaum mit zugehörigen Bearbeiterzuordnungsauadrücken

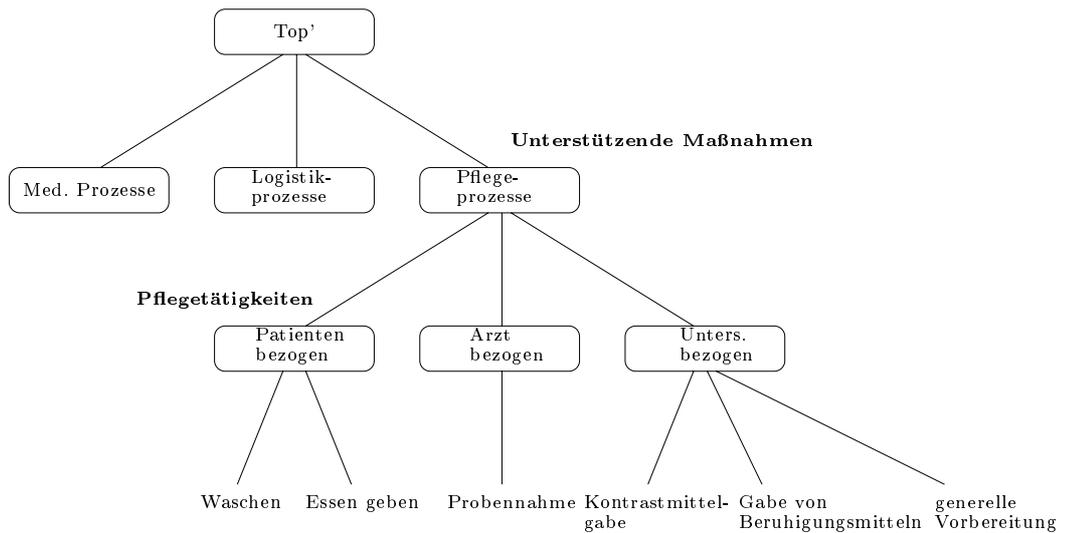


Abb. 7.26: Prozessbaum mit zugeordneten Aktivitätenkategorien und -vorlagen

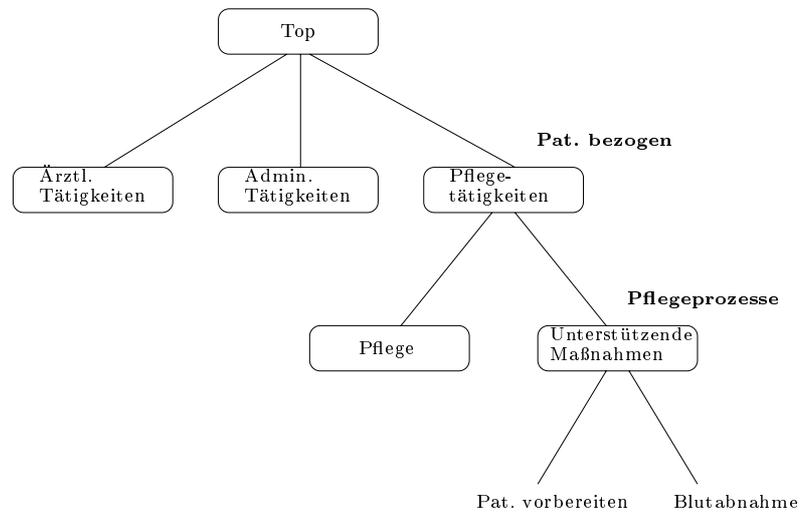


Abb. 7.27: Aktivitätenbaum mit zugeordneten Prozesskategorien und -vorlagen

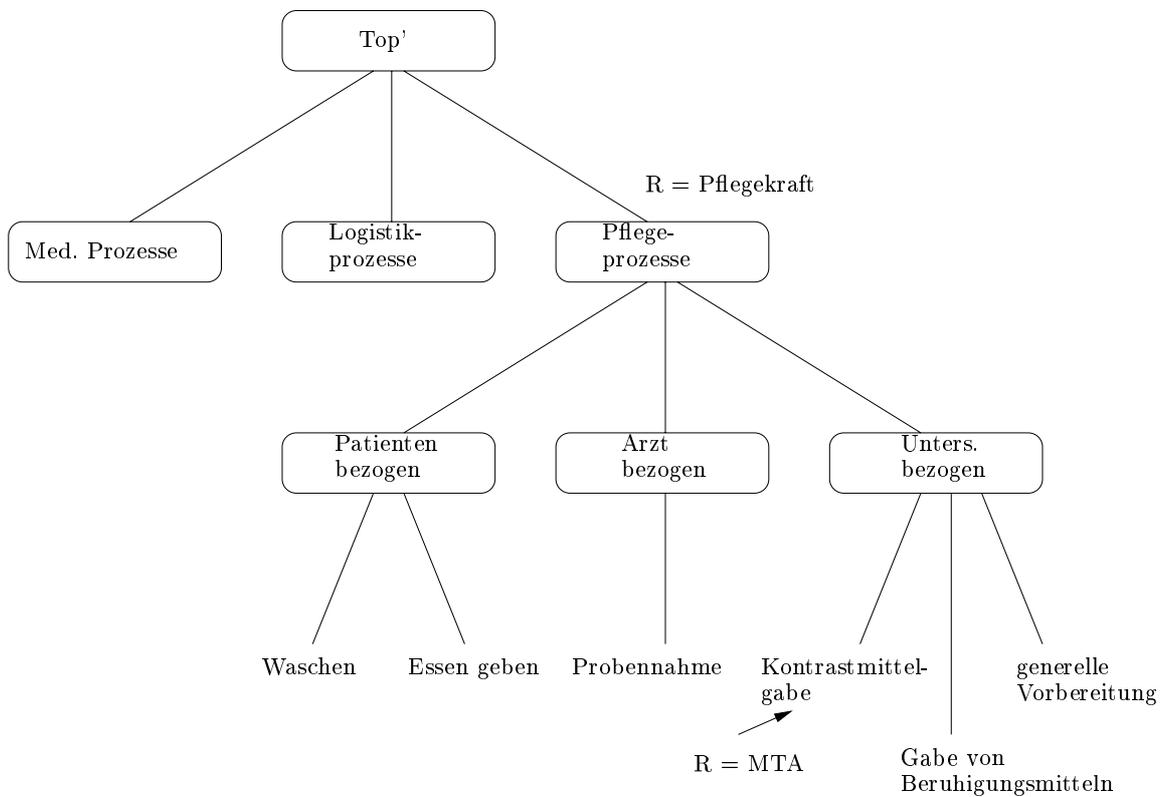


Abb. 7.28: Prozessbaum mit zugeordneten Bearbeiterzuordnungsausdrücken

spiel darf die Pflegekraft lediglich die Aktivität „Pat. vorbereiten“ generell entfernen. (Aspekt 1)

2. Anschließend wird einer der folgenden zwei Wege zur Bestimmung der Aktivitäten, die in einer bestimmten Prozessinstanz gelöscht werden dürfen, verfolgt.

- (a) Ermittlung der Prozessvorlage aus dem Prozessbaum aus Abb. 7.26, die zur zu ändernden Prozessinstanz gehört. Anschließend Bestimmung eines Weges von diesem Vorlagenblatt zur Wurzel des Prozessbaumes. Im Beispiel ist der Weg gegeben durch:

„generelle Vorbereitung“ → „Unters. bezogen“ → „Pflegeprozesse“ → „Top“.
Auf dem Weg zur Wurzel wird für jeden Knoten die zugeordnete Aktivitätenkategorie berücksichtigt. Im Beispiel ist der Prozesskategorie „Pflegeprozesse“ die Aktivitätenkategorie „Unterstützende Maßnahmen“ zugeordnet. Das bedeutet, dass der Benutzer in Prozessinstanzen, die aus Vorlagen der Prozesskategorie „Pflegeprozesse“ oder ihrer untergeordneten Kategorien hervorgehen, Aktivitäten löschen kann, die der Aktivitätenkategorie „Unterstützende Maßnahmen“ untergeordnet sind. (Aspekt 2)

- (b) Ermittlung eines Weges im Aktivitätenbaum aus Abb. 7.27 von der Aktivitätenvorlage „Patient vorbereiten“ zur Wurzel des Aktivitätenbaumes. Dieser Weg ist im Beispiel beschrieben durch:

„Pat. vorbereiten“ → „Unterstützende Maßnahmen“ → „Pflegetätigkeiten“ → „Top“.

Auf dem Weg vom Vorlagenblatt zur Wurzel des Aktivitätenbaumes wird die Zuordnung der Prozesskategorien/-vorlagen zu Knoten des Aktivitätenbaumes berücksichtigt. Hier ist der Aktivitätenkategorie „Unterstützende Maßnahmen“ die Prozesskategorie „Pflegeprozesse“ zugeordnet und der übergeordneten Aktivitätenkategorie „Pflegetätigkeiten“ die Prozesskategorie „Pat. bezogen“. D. h. der Benutzer darf der Aktivitätenkategorie „Unterstützende Maßnahmen“ untergeordnete Aktivitätenvorlagen in Prozessinstanzen löschen, die sowohl aus Vorlagen der Prozesskategorie „Pflegeprozesse“ als auch der Prozesskategorie „Pflegetätigkeiten“ hervorgehen. (Aspekt 2, inverse Darstellung)

3. Zur Bestimmung, ob der konkrete Benutzer Schritte aus einer Prozessinstanz entfernen darf und welche dies sind, werden im Prozessbaum aus Abb. 7.28 die Bearbeiterzuordnungsausdrücke auf dem Weg vom Vorlagenblatt bis zur Wurzel des Prozessbaumes ermittelt. Im Beispiel der Prozessvorlage „generelle Vorbereitung“ ist auf der Ebene der Prozesskategorie „Pflegeprozesse“ der Bearbeiterzuordnungsausdruck „R = Pflegekraft“ definiert, der besagt, dass Benutzer, die die Rolle *Pflegekraft* einnehmen, in Prozessinstanzen, die aus Prozessvorlagen dieser oder der untergeordneten Prozesskategorien hervorgehen, Aktivitäten löschen darf. (Aspekt 3)

Im Beispiel will die Pflegekraft die Aktivität „Pat. vorbereiten“ aus dem Ablauf der radiologischen Untersuchung entfernen. Anhand der oben durchgeführten Schritte, darf diese Person den Schritt „Pat. vorbereiten“ aus Prozessinstanzen entfernen, die in Prozessinstanzen der Kategorie „Pflegeprozesse“ auftreten.

7.3.2.4 Abschließende Bemerkungen

Auch in Bezug auf das Löschen von Aktivitäten gilt wieder, dass die konkrete Reihenfolge, in der die Berechtigungen gemäß den Aspekten 1 – 4 überprüft werden, implementierungsabhängig ist. Im Allgemeinen dürfte es im Zusammenhang mit dem Löschen von Aktivitäten,

einem Benutzer, der eine Löschoption anwenden möchte, zunächst alle Schritte der Prozessinstanz zu offerieren, die aufgrund ihrer Kennzeichnung (vgl. Aspekt 4) und ihres aktuellen Zustandes löscherbar sind. Wählt der betreffende Benutzer dann einen dieser Schritte aus, muss zusätzlich geprüft werden, ob er hierfür die entsprechenden Berechtigungen gemäß den Aspekten 1 – 4 besitzt.

7.3.3 Rechtekonzept für das dynamische Verschieben von Aktivitäten

In den vorangehenden Abschnitten wurden Rechtekonzepte für das dynamische Einfügen und Löschen von Aktivitäten entwickelt und beschrieben. Im Folgenden soll ein ähnliches Konzept für das dynamische Verschieben von Aktivitäten entwickelt werden.

Das dynamische Verschieben von Schritten ist in der Regel ein Herauslösen eines Aktivitätsknotens aus seinem bisherigen Kontext und Einfügen dieses Schrittes in einen anderen Kontext innerhalb derselben Prozessinstanz. Für die Umsetzung des dynamischen Verschiebens von Schritten gibt es zwei alternative Vorgehensweisen. Entweder wird die Operation des dynamischen Verschiebens von Schritten als Löschen mit anschließendem Einfügen neuer Schritte betrachtet oder es werden eigene Berechtigungen zum Verschieben von Schritten definiert. Bei der Umsetzung der ersten Variante dürfte eine Aktivität durch solche Benutzer „verschoben“ werden, die einerseits die entsprechenden Löschberechtigungen besitzen und andererseits die erforderlichen Einfügeberechtigungen. Dieser Ansatz ist im Allgemeinen jedoch zu restriktiv. Beispielsweise kann das Aufklärungsgespräch im Kontext einer klinischen Untersuchung (schon aus juristischen Gründen) nicht entfallen, es kann jedoch vorzeitig durchgeführt werden, d.h. die Aktivität wird im Ablauf vorgezogen.

Anhand dieses Beispiels ist ersichtlich, dass der „Umweg“ über das Löschen eines Schrittes und das spätere Einfügen an einer anderen Stelle sehr aufwendig und äußerst problematisch ist. Aus diesem Grund sollten explizite Rechte für das Verschieben von Schritten definiert werden können. Hierauf wird im Folgenden näher eingegangen, wobei analog zum Beispiel des dynamischen Einfügens von Schritten in Kap. 7.3.1 und des dynamischen Löschens von Schritten in Kap. 7.3.2 vorgegangen wird.

7.3.3.1 Erforderliche Rechte zur Kontrolle dynamischer Verschiebeoperationen

Die Bestimmung der berechtigten Personen, die konkrete Schritte innerhalb einer Prozessinstanz verschieben dürfen bzw. die Bestimmung der Aktivitäten, die ein konkreter Benutzer verschieben darf, wird anhand der folgenden Fragestellungen diskutiert:

1. Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) darf ein bestimmter Benutzer generell verschieben?

Unabhängig von der Prozessvorlage bzw. -instanz, in der eine Aktivität verschoben werden soll, stehen einem Benutzer im Allgemeinen nicht alle im Repository hinterlegten Aktivitätsvorlagen zur Auswahl. Beispielsweise darf im Klinikkontext eine Pflegekraft nicht die Medikamentengabe vor eine Blutuntersuchung verschieben. Hierdurch würden die Laborwerte verfälscht.

2. Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) dürfen in Instanzen einer bestimmten Prozessvorlage verschoben werden?

Nicht alle Aktivitäten der Prozessinstanz einer bestimmten Prozessvorlage dürfen an eine beliebige Stelle im Ablauf verschoben werden. Beispielsweise darf, bezogen auf

Krankenhausprozesse, die Aufklärung eines Patienten über eine Untersuchung oder Operation aus juristischen Gründen nicht erst nach Durchführung dieser Maßnahme stattfinden. Der Patient muss der Maßnahme im vornherein zustimmen. Wurde er zuvor nicht über die Maßnahme aufgeklärt, hatte er auch keine Chance, die Durchführung der Untersuchung oder Operation zu verhindern.

3. Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) in Instanzen einer bestimmten Prozessvorlage verschieben?

Die Bedingungen 1 und 2 reichen allein noch nicht aus, um zu bestimmen, ob ein bestimmter Benutzer eine bestimmte Aktivität in einer bestimmten Prozessinstanz verschieben darf. Zu einer Prozessvorlage kann es mehrere Instanzen geben, für die ein Benutzer eine Aktivität verschieben darf, während er dies für andere Instanzen nicht darf. Beispielsweise können radiologische Untersuchungen von unterschiedlichen Stationen im Krankenhaus angefordert werden. Nun darf aber ein Mitarbeiter von Station 1 keine Änderungen an Instanzen durchführen, die von Station 2 angefordert wurden und umgekehrt. Das bedeutet, Benutzer dürfen nur an Instanzen Änderungen vornehmen, an denen sie auch beteiligt sind. Diese Beteiligung kann z. B. anhand der Organisationseinheit eines Mitarbeiters definiert werden. Ein Beispiel für diesen Fall ist in Abb. 7.6 auf Seite 85 gegeben.

Diese Fragestellungen sollen in der Folge aufgegriffen werden, um zu zeigen, wie in ADEPT entsprechende Berechtigungen festgelegt werden können.

7.3.3.2 Definition der Verschieberechte

Bei der Bestimmung der verschiebbaren Aktivitäten und der berechtigten Benutzer wird analog zur Vorgehensweise des dynamischen Einfügens und Löschens von Aktivitäten vorgegangen. Im Folgenden soll daher nur kurz auf die Problematik der Berechtigungen im Zusammenhang mit dem Verschieben von Aktivitäten eingegangen werden, wobei auf die Fragestellungen 1 – 3 aus Abschnitt 7.3.3.1 Bezug genommen wird. Weiterhin gelten die in den Abschnitten 7.3.1.2 und 7.3.1.3 gemachten Vereinfachungen und Kategorisierungen.

Aspekt 1: Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) darf ein bestimmter Benutzer generell verschieben?

Um festzulegen, welche Aktivitäten generell von einem bestimmten Benutzer verschoben werden dürfen unabhängig von einer Prozessinstanz, müssen für die einzelnen Aktivitätsvorlagen bzw. Aktivitätskategorien im Aktivitätenbaum entsprechende Bearbeiterzuordnungsausdrücke zugeordnet werden. Ein Ansatz hierzu wird im Folgenden anhand eines Beispiels vorgestellt.

Gegeben seien die im Repository hinterlegten Aktivitätenkategorien und -vorlagen sowie deren hierarchische Beziehungen im Aktivitätenbaum in Abb. 7.29. Für die Festlegung der Rechte gelten die gleichen Bedingungen wie sie in Abschnitt 7.3.1.4 beschrieben sind, für einzelne Kategorien definierte Berechtigungen sind auch auf untergeordneten Kategorien gültig und Berechtigungen können durch speziellere oder allgemeinere Bearbeiterformeln überschrieben werden.

In Abb. 7.29 ist ein Aktivitätenbaum skizziert, dessen Knoten oder Blättern Bearbeiterformeln zugeordnet sind. Im Beispiel gilt die Bearbeiterzuordnung „R = Arzt“ für die Akti-

vitätenkategorie A_1 und alle ihr untergeordneten Aktivitätenkategorien und -vorlagen, die keine speziellere Bearbeiterzuordnung besitzen (a_{12} , a_{13} , a_{112}). Für die Aktivitätsvorlage a_{111} ist die speziellere Bearbeiterzuordnung „R = Arzt AND OE = Radiologie“ festgelegt. Das bedeutet, dass die Aktivitätsvorlage a_{111} nur von Ärzten der Radiologie verschoben werden darf.

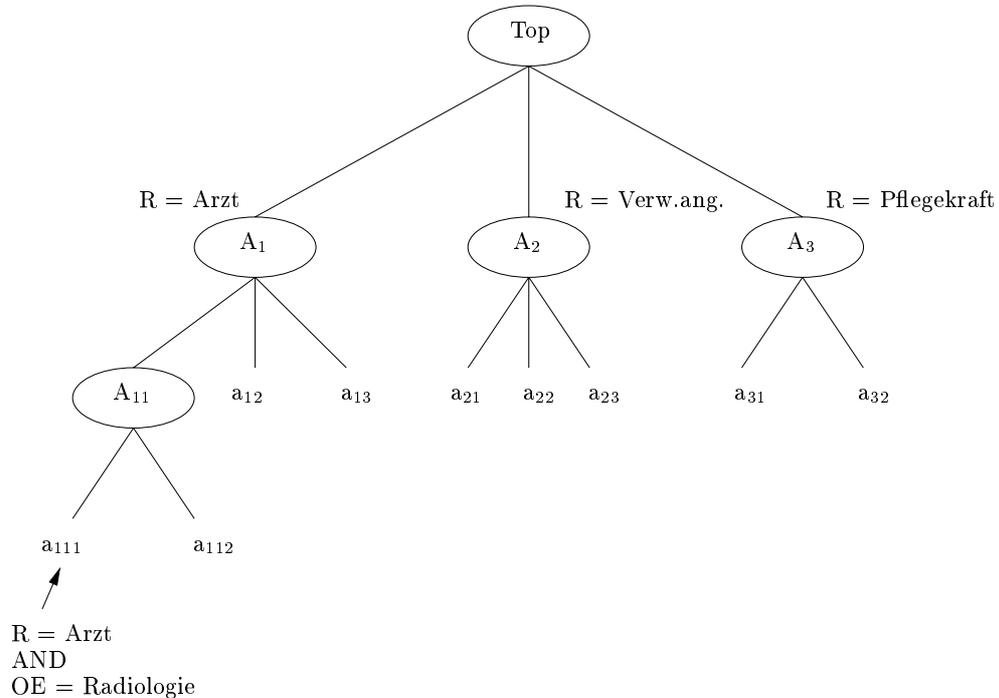


Abb. 7.29: Aktivitätenbaum mit Bearbeiterzuordnungen

Aspekt 2: Welche Art von Aktivitäten (bzw. Aktivitätsvorlagen) dürfen in Instanzen einer bestimmten Prozessvorlage verschoben werden?

In diesem Kapitel wurden bisher Berechtigungskonzepte vorgestellt, die regeln, welche Art von Aktivitäten bzw. Aktivitätsvorlagen ein Benutzer unabhängig von einer Prozessinstanz generell verschieben darf. Wie bei den Berechtigungskonzepten für das Einfügen und Löschen von Schritten erfolgten auch beim Verschieben diese Festlegungen noch ohne Bezugnahme auf konkrete Prozessvorlagen. Dies reicht jedoch nicht aus, um die Berechtigungen auf bestimmte Prozessinstanzen einzuschränken.

Zusätzlich zur Abhängigkeit von der Prozessvorlage muss es möglich sein, die Art von Aktivitäten, die in einer Instanz verschoben werden dürfen, zu begrenzen. Daher muss für Prozessvorlagen festgelegt werden können, auf welche Aktivitätsvorlagen beim Verschieben Bezug genommen werden darf.

Im Folgenden soll anhand eines Beispiels dieser Vorgang nachvollzogen werden. Dazu werden im Prozessbaum den verschiedenen Prozesskategorien bzw. -vorlagen entsprechende Aktivitätenkategorien oder -vorlagen zugeordnet, auf die beim Verschieben von Aktivitäten Bezug genommen werden darf. Ein Beispiel für die Zuordnung von Aktivitätenkategorien bzw. -vorlagen zu Prozesskategorien ist im Prozessbaum aus Abb. 7.30 dargestellt.

Soll nun eine Aktivität in einer bestimmten Prozessinstanz verschoben werden, wird wie beim Einfügen bzw. Löschen vorgegangen.

1. Ermittlung der zur abzuändernden Prozessinstanz gehörenden Prozessvorlage.
2. Bestimmung eines Weges von der Prozessvorlage zur Wurzel des Prozessbaumes. Für jeden Knoten auf diesem Weg muss dabei die entsprechende Aktivitätenkategorienzuordnung berücksichtigt werden.

Bezogen auf das in Abb. 7.30 skizzierte Beispiel, ergibt sich für eine gegebene Prozessinstanz der Prozessvorlage p_{121} der folgende Weg: $p_{121} \rightarrow P_{12} \rightarrow P_1 \rightarrow \text{Top}'$. Hier sind allen Prozesskategorien auf dem Weg zur Wurzel des Prozessbaumes Aktivitätenkategorien oder -vorlagen zugeordnet, d.h. in Instanzen der Prozessvorlage p_{121} dürfen alle Aktivitäten verschoben werden, deren Vorlagen aus der Aktivitätenvorlage a_{22} oder den Aktivitätenvorlagen der Aktivitätenkategorien A_{11} und A_3 hervorgehen.

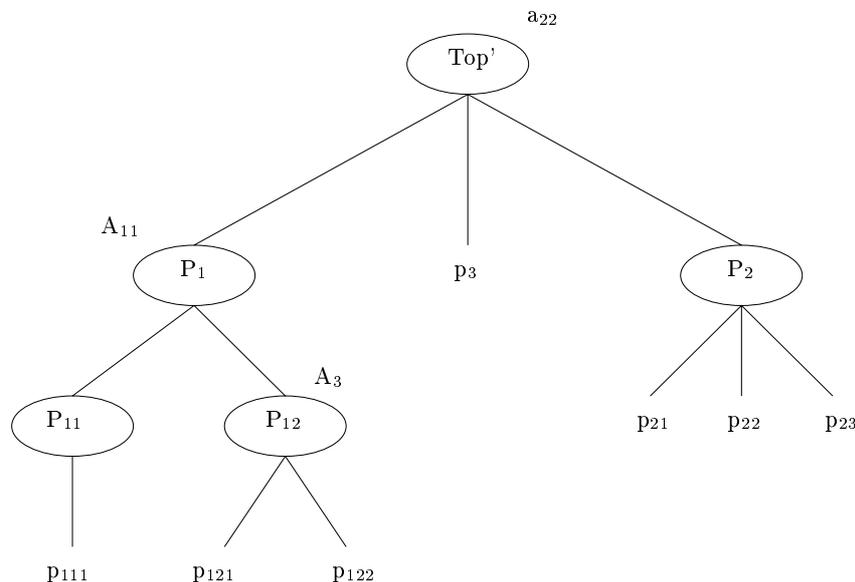


Abb. 7.30: Prozessbaum mit zugehörigen Aktivitätenkategorien bzw. -vorlagen

Wie bei den vorangegangenen Berechtigungskonzepten für das Einfügen und Löschen von Aktivitäten existiert auch hier die inverse Sicht. Auf die Beschreibung dieser Sicht soll hier jedoch verzichtet werden.

Aspekt 3: Welche konkreten Benutzer dürfen eine bestimmte Aktivität(envorlage) in Instanzen einer bestimmten Prozessvorlage verschieben?

Zur Bestimmung der Benutzer, die eine Aktivität in einer Instanz einer bestimmten Prozessvorlage verschieben dürfen, wird analog zur Vorgehensweise des ersten Aspekts des Verschiebens auf Seite 108 vorgegangen. Jedoch werden jetzt den Prozesskategorien im Prozessbaum Mitarbeiterzuordnungsdrücke zugeordnet, die die zum Verschieben berechtigten Benutzer spezifizieren. Natürlich kann auch hier auf der Vorlagenebene ein speziellerer Ausdruck als der auf der Kategorieebene angegebene Ausdruck definiert werden. Der Prozessbaum des vorhergehenden Beispiels hat z. B. die Mitarbeiterzuordnungen aus Abb. 7.31.

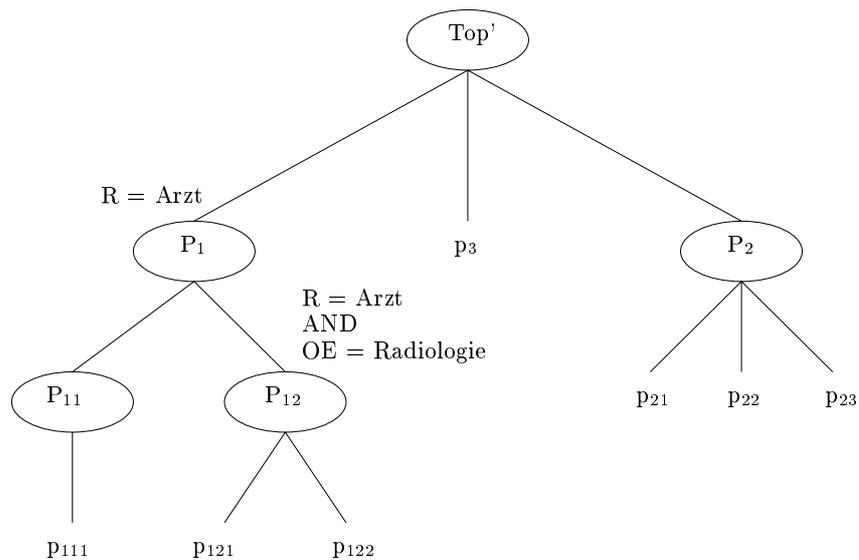


Abb. 7.31: Prozessbaum mit zugehörigen Bearbeiterzuordnungen

7.3.3.3 Konkretes Beispiel für das dynamische Verschieben von Aktivitäten

Im Folgenden ist die Vorgehensweise zur Bestimmung von Aktivitäten, die ein bestimmter Benutzer in einer bestimmten Prozessinstanz verschieben darf, an einem konkreten Beispiel skizziert.

Gegeben ist wieder der Ablauf einer radiologischen Untersuchung aus Kap. 3.1. Der behandelnde Arzt von Station 2 möchte die Aktivität „Patient vorbereiten“ mit der Prozessvorlage „Kontrastmittelgabe“ direkt vor den Schritt „Untersuchung durchführen“ verschieben.

Im Folgenden sind die zugehörigen Prozess- und Aktivitätenbäume gegeben. Abb. 7.32 skizziert den Aktivitätenbaum mit den zugehörigen Bearbeiterzuordnungen. Der Übersichtlichkeit halber wurde sowohl in diesem als auch allen anderen Aktivitäten- und Prozessbäumen auf die genaue Definition aller Kategorien und Vorlagen verzichtet. Lediglich die Bereiche, die in diesem Beispiel von Bedeutung sind, sind genauer betrachtet worden.

Abb. 7.33 zeigt den zugehörigen Prozessbaum mit den zugeordneten Aktivitätenkategorien bzw. -vorlagen. Der zugehörige Prozessbaum mit Bearbeiterzuordnungen ist in Abb. 7.34 dargestellt.

Die Bestimmung, ob ein konkreter Benutzer mit der Rolle *Arzt* der Organisationseinheit *Radiologie* die Aktivität „Patient vorbereiten“ verschieben darf, wird anhand der in Abschnitt 7.3.3.2 diskutierten Fragestellungen vorgenommen. Hierbei wird angenommen, dass, wenn eine Person einen Schritt generell verschieben darf, sie diesen an jede beliebige Stelle im Ablaufgraphen verschieben darf.

1. Ermittlung aller generell vom Benutzer mit der Rolle *Arzt* verschiebbaren Aktivitätenvorlagen im Aktivitätenbaum aus Abb. 7.32 unter Berücksichtigung der Bearbeiterzuordnungsausdrücke. Daraus ergibt sich eine Menge von Aktivitätenvorlagen, die dieser Benutzer generell, also unabhängig von einer konkreten Prozessinstanz verschieben darf. (Aspekt 1)
2. Ermittlung der Prozessvorlage, die verschoben werden soll. Im Beispiel ist dies die

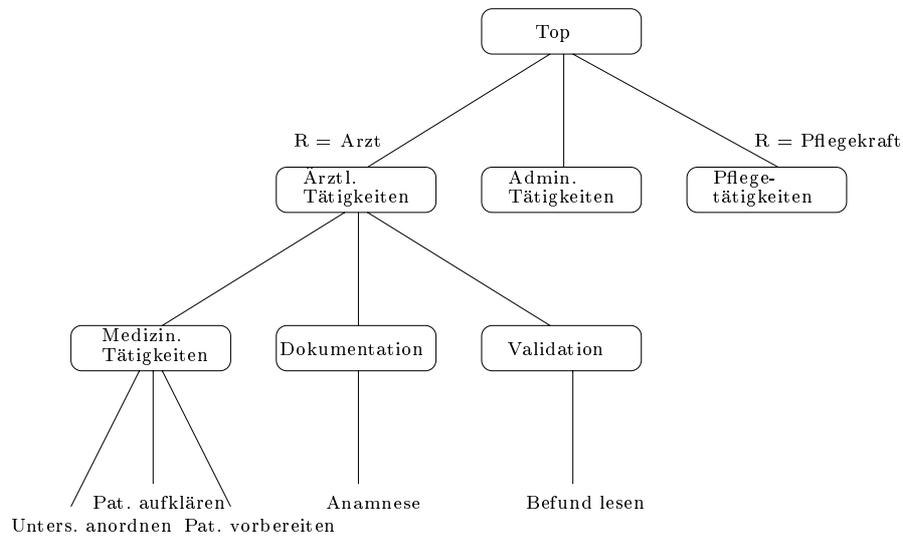


Abb. 7.32: Aktivitätenbaum mit Bearbeiterzuordnungen zum Beispiel des dynamischen Verschiebens von Schritten

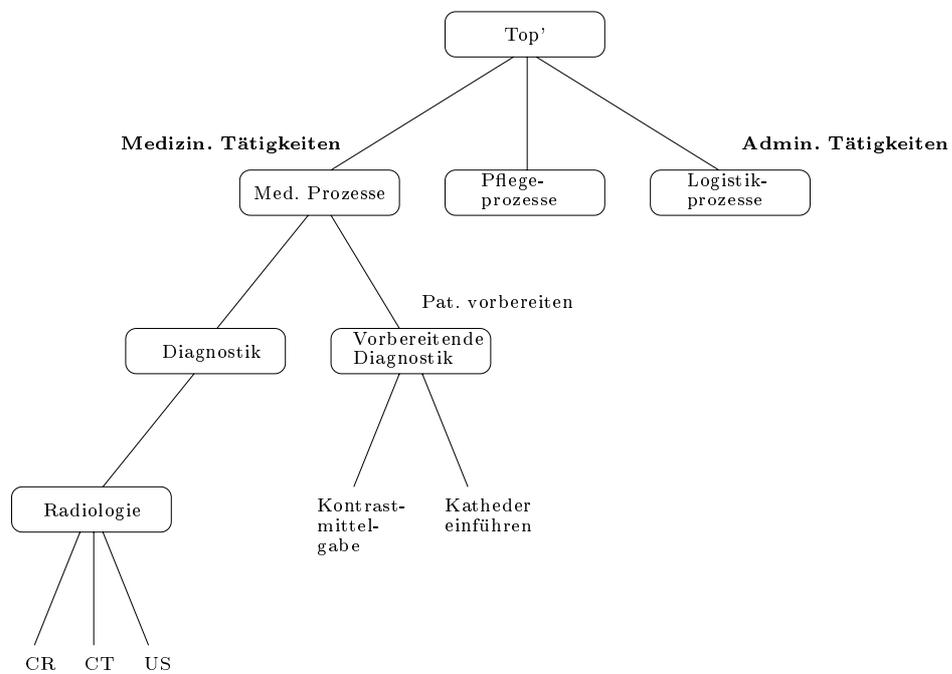


Abb. 7.33: Zugehöriger Prozessbaum mit zugeordneten Aktivitätenkategorien bzw. -vorlagen

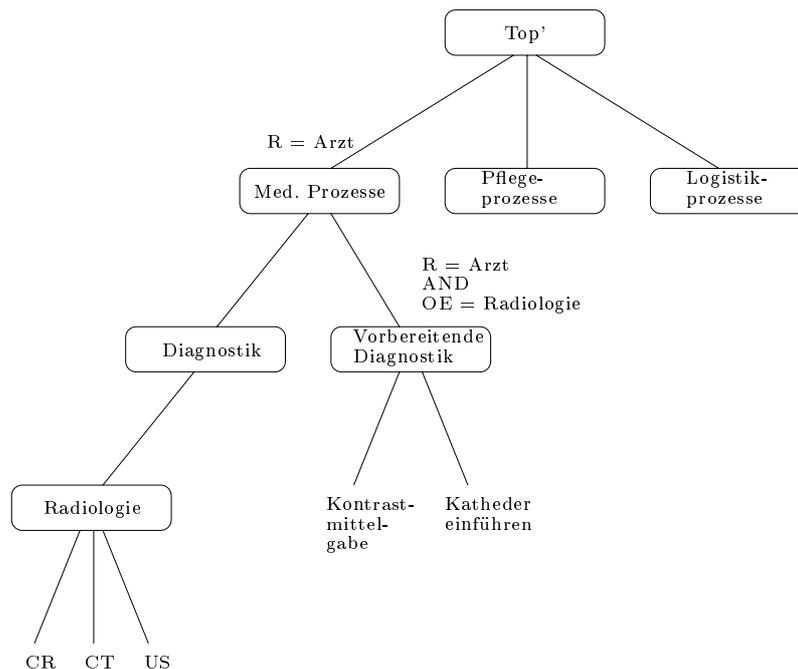


Abb. 7.34: Zugehöriger Prozessbaum mit zugeordneten Bearbeiterzuordnungen

Prozessvorlage *Kontrastmittelgabe*. Dann wird ein Weg von diesem Vorlagenblatt zur Wurzel des Prozessbaumes aus Abb. 7.33 bestimmt. Der Weg ergibt sich zu:

„Kontrastmittelgabe“ \rightarrow „Vorbereitende Diagnostik“ \rightarrow „Med. Prozesse“ \rightarrow „Top“ . Auf dem Weg zur Wurzel werden für alle Knoten die zugeordneten Aktivitätenkategorien berücksichtigt. Der Prozesskategorie „Vorbereitende Diagnostik“ ist die Aktivitätenkategorie „Pat. vorbereiten“ zugeordnet und der Prozesskategorie „Med. Prozesse“ die Aktivitätenkategorie „Medizin. Tätigkeiten“ . D. h. der berechtigte Benutzer darf in den aus der Prozesskategorie „Med. Prozesse“ hervorgehenden Prozessinstanzen alle Aktivitäten verschieben, die der Aktivitätenkategorie „Medizin. Tätigkeiten“ untergeordnet sind. Außerdem darf der Benutzer in den Prozessvorlagen, die der Prozesskategorie „Vorbereitende Diagnostik“ untergeordnet sind, die Aktivität „Pat. vorbereiten“ verschieben. (Aspekt 2)

3. Zur Bestimmung, ob der konkrete Benutzer Schritte in der Prozessinstanz verschieben darf und welche dies sind, werden im Prozessbaum aus Abb. 7.34 die Bearbeiterzuordnungsausdrücke auf dem Weg vom Prozessvorlagenblatt bis zur Wurzel des Prozessbaumes ermittelt. Die der Prozessvorlage „Kontrastmittelgabe“ direkt übergeordnete Prozesskategorie „Vorbereitende Diagnostik“ ist ein Bearbeiterzuordnungsausdruck festgelegt, der für die untergeordneten Vorlagen Gültigkeit hat. Das bedeutet, wenn die Prozessvorlage „Kontrastmittelgabe“ verschoben werden soll, muss der Akteur die Rolle *Arzt* besitzen und der Organisationseinheit *Radiologie* angehören. (Aspekt 3)

In diesem Beispiel will der behandelnde Arzt von Station 2 den Schritt *Patient vorbereiten* durch *Kontrastmittelgabe* verschieben. Anhand der eben durchgeführten Untersuchung darf der behandelnde Arzt den Schritt nicht(!) verschieben, da er nicht der Organisationseinheit *Radiologie* angehört. Nur Ärzte der Organisationseinheit *Radiologie* dürfen die Aktivität *Patient vorbereiten* in Kombination mit der *Kontrastmittelgabe* verschieben.

7.3.4 Weitergehende Betrachtungen

Im Folgenden werden einige weiterführende Aspekte im Zusammenhang mit Berechtigungen für dynamische Änderungen diskutiert. Der Einfachheit halber wird sich dabei auf additive Änderungen beschränkt, d. h. auf das Einfügen von Schritten in Prozessgraphen.

Bisher nicht behandelt wurden Fragestellungen der Art, in welchen Regionen eines Prozessinstanzgraphen eine bestimmte Aktivität eingefügt werden darf und in welchen nicht. Entsprechende Festlegungen können für die Praxis durchaus sinnvoll sein, insbesondere im Zusammenhang mit bereichsübergreifenden Prozessen, bei denen Ausschnitte (Partitionen) des Prozessgraphen von verschiedenen Organisationseinheiten durchgeführt werden². In Abb. 7.35 ist ein Beispiel für einen Prozess gegeben, der aus mehreren Regionen, d.h. zusammenhängenden Teilgraphen, besteht. Es wird angenommen, dass die Aktivitäten aus diesen Regionen in verschiedenen Organisationseinheiten bearbeitet werden. Die Aktivitäten E und F der Region R1 sind also einem anderen Unternehmensbereich zugeordnet, wie die Aktivitäten A, B, C, D und G der Region R2.

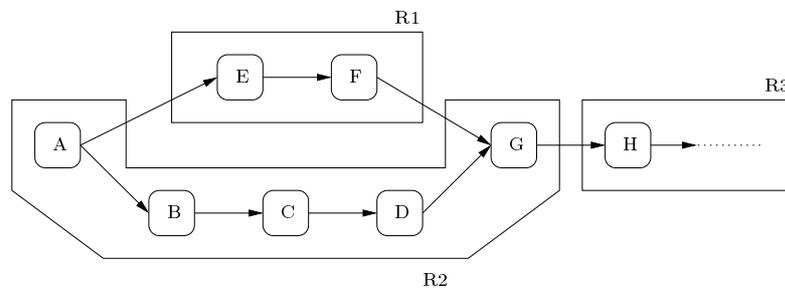


Abb. 7.35: Beispiel eines Ausführungsgraphen, dessen Aktivitäten in mehreren Bereichen parallel ablaufen

Die Anwendbarkeit einer (dynamischen) Änderung kann in einem solchen Szenario nicht vom Benutzer, der die Änderung vornehmen will, abhängig sein. Sie kann zusätzlich von der Region des Prozessgraphen, die von der Änderung betroffen ist, abhängig sein. Das bedeutet, dass Änderungen, die für eine bestimmte Region erlaubt sind, nicht notwendigerweise auch auf andere Regionen anwendbar sein müssen. Dies soll anhand von zwei Beispielen illustriert werden:

Beispiel 1: Eine bestimmte Einfügeoperation ist nur in einer bestimmten Region des Prozessgraphen zulässig.

Beispiel 2: Bezogen auf den Prozessgraphen aus Abb. 7.35 darf der Benutzer U_1 die Aktivität x in Region R1 einfügen, Benutzer U_2 dagegen nicht. U_2 darf die Aktivität x aber in andere Regionen des Prozessgraphen hinzunehmen.

Diese Beispiele sind mit den bisherigen Konzepten (vgl. Abschnitt 7.3.1) jedoch nicht abbildbar. Die Umsetzung der Beschränkung auf bestimmte Regionen ist auch nicht trivial. Zum einen wäre die Rechtfestlegung bei der Verwendung von bestimmten Regionen sehr aufwendig, zum anderen wären Änderungen problematisch, die an den Rändern von Regionen durchgeführt werden sollen. Soll z. B. in Abb. 7.35 eine Aktivität zwischen den Aktivitäten A und E eingefügt werden, ist nicht klar definiert, welcher Region die neu eingefügte Akti-

²Dies ist zumindest nicht direkt abbildbar. Lediglich bei der Verwendung von Sub-Prozessen wäre dieses Problem beherrschbar (vgl. Abb. 7.8). Allerdings ist dieser Weg auch sehr aufwendig.

viät ausschließlich angehören soll. Um dieses Problem zu beseitigen, müssten am Prozess-Metamodell Änderungen derart vorgenommen werden, dass Aktivitäten einer Region einen definierten Start- und End-Knoten besitzen. Für den Ausführungsgraphen in Abb. 7.35 würde sich der „neue“ Ausführungsgraph, wie in Abb. 7.36 dargestellt, ergeben. Hier hat jede Region einen definierten Start- und Ende-Knoten und nur zwischen diesen wäre das Einfügen von Schritten erlaubt. Formell ausgedrückt wäre das Einfügeintervall in Region R1 gegeben durch $]s_1, \dots, e_1[$.

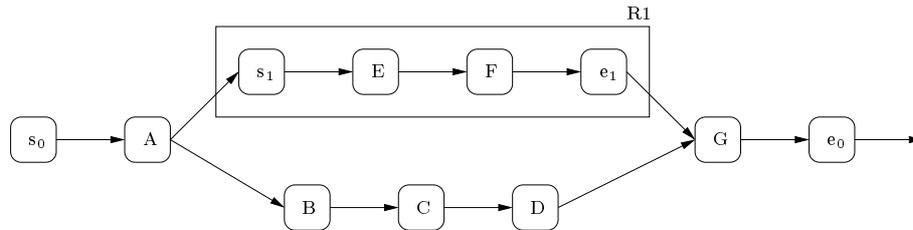


Abb. 7.36: Beispiel für den Prozessinstanzgraphen aus Abb. 7.35 nach Änderungen des Prozess-Metamodells

Semantische Abhängigkeiten zwischen Prozessschritten, die beim Löschen bzw. Einfügen von Schritten berücksichtigt werden, lassen sich mit dem vorgestellten Ansatz ebenfalls nicht umsetzen. Beispielsweise kann es sinnvoll sein, dass wenn im Prozessgraphen eine Aktivität B vorkommt, immer auch eine Aktivität A im Graphen enthalten ist und diese zwingend vor B ausgeführt wird. Um im Kontext von Klinikprozessen zu bleiben, muss z. B. einem Schritt „Befund lesen“ zwingend ein Schritt „Befund schreiben“ vorausgehen. Für das Löschen von Schritten bedeutet das: Wird die Aktivität „Befund schreiben“ gelöscht, muss auch die Aktivität „Befund lesen“ aus dem Prozessgraphen entfernt werden. Umgekehrt kann ein Schritt „Befund lesen“ nur dann in den Prozessgraphen eingefügt werden, wenn entweder die Aktivität „Befund schreiben“ bereits im Graphen enthalten ist oder gleichzeitig eingefügt wird. Auch die explizite Berücksichtigung solcher Abhängigkeiten macht Erweiterungen des zugrundeliegenden Metamodells erforderlich. Auf sie kann im Rahmen dieser Diplomarbeit nicht eingegangen werden.

7.3.5 Fazit

Da Ad-hoc Abweichungen nicht bereits bei der Prozessmodellierung berücksichtigt werden können, müssen noch zur Laufzeit entsprechende Anpassungen der Prozessstrukturen vorgenommen werden können. Hierfür wurde ein Konzept entwickelt, das es erlaubt diese Berechtigungen festzulegen und zu kontrollieren. Im Allgemeinen werden dazu die bekannten Bearbeiterzuordnungsdrücke verwendet und Kategorien von Aktivitäten bzw. Prozessen eingeführt. Letztere sind auch im Hinblick auf die Organisation der verschiedenen Vorlagen hilfreich.

Zur Festlegung von Änderungsberechtigungen wurden also keine grundlegenden Konzepte entwickelt, sondern es wurde versucht, mithilfe von bekannten Strukturen entsprechende Berechtigungen zu definieren.

Kapitel 8

Zusammenfassung

Im Rahmen der in der Geschäftswelt immer stärker in den Vordergrund rückenden Optimierung von bereichsübergreifenden Geschäftsprozessen gewinnt der Einsatz von Prozess-Management-Systemen zunehmend an Bedeutung. Sowohl komplexe Fertigungsprozesse als Dienstleistungsprozesse müssen durch den Einsatz solcher Systeme adäquat unterstützt werden können. Mit wachsender Abdeckung des Bereiches vollständig planbarer Prozessabläufe (Production Workflows) geht es dabei auch zunehmend um die Unterstützung nicht standardisierbarer und nicht immer vorhersehbarer Prozessabläufe. Als unmittelbare Folge hieraus ergibt sich auch die Frage, welche Personen die notwendigen dynamischen Anpassungen zur Laufzeit vornehmen können sollen.

Eine nähere Betrachtung der kommerziell verfügbaren Prozess-Management-Systeme MQ-Series Workflow, Staffware, WorkParty und ProMInanD hat verdeutlicht, wie wenig diese Vertreter von Production Workflow- und Ad-hoc Workflow-Systemen geeignet sind, auf zur Laufzeit notwendige Abweichungen vom modellierten Standardablauf eingehen zu können. Entweder sind die Arbeitsabläufe im System starr festgelegt, so dass der Benutzer keinerlei Möglichkeit hat, entsprechende Änderungen vorzunehmen oder die Arbeitsabläufe sind nicht fest modelliert, so dass der Benutzer zur Laufzeit den nächsten Arbeitsschritt bestimmen muss.

Die Notwendigkeit des Eingreifens in den Ablauf fest vorgegebener Prozesse lässt sich anhand des einfachen Beispiels einer radiologischen Untersuchung leicht verdeutlichen. In Krankenhäusern bilden Abweichungen vom standardisierten Behandlungs- und Untersuchungsprozedere den „Normalfall“ – Krankheiten, Behandlungen und Untersuchungen nehmen vielfach einen nicht vorhersehbaren und nicht deterministischen Verlauf. Gerade dieses Beispiel zeigt auch die Notwendigkeit der Definition von Berechtigungen auf, welche den Zugriff auf Prozessabläufe und deren Änderungen regeln. Die Änderung eines Medikationsplans durch eine Pflegekraft ist z. B. schwer vorstellbar.

Im Wissen um die Schwächen existierender Systeme und die Notwendigkeit der benannten Eigenschaften von Prozess-Management-Systemen wird seit 1995 in der Abteilung Datenbanken und Informationssysteme der Universität Ulm mit dem „Application Development Based on Encapsulated Premodeled Process Templates“-Ansatz (ADEPT) ein Prozess-Management-System der nächsten Generation entwickelt. Das ADEPT-System ist in der Lage, Abweichungen vom Standardablauf sowohl während der Ausführung eines Prozesses als auch mögliche Änderungen bereits zum Zeitpunkt der Modellierung zu berücksichtigen. Für den Benutzer ist es somit ein Leichtes, Abweichungen in der Realität auch im Prozess-Management-System abbilden zu können. Um mit ADEPT arbeiten zu können, sind zunächst Berechtigungen für die Ausführung von Aktivitäten während des Standardablaufs sowie Berechtigungen für den Zugriff auf verschiedene Datenelemente zu definieren. Diese Rechte werden im ADEPT-

System durch Bearbeiterformeln bestimmt, die beliebig komplex gestaltet werden können. Für den Fall von Abweichungen müssen derartige Berechtigungen ebenfalls definiert sein. Darüber hinaus ist die Existenz weiterer Rechte vorausgesetzt, die bestimmen, welche Aktivitäten im Prozessablauf geändert werden dürfen und in welchen Prozessabläufen diese Änderungen durchführbar sind. Bei der Festlegung von Berechtigungen, seien es Ausführungs-, Zugriffs- oder Änderungsrechte, gilt jedoch immer, dass diese möglichst kompakt definiert werden können müssen, um die Pflegbarkeit der Berechtigungen zu wahren. Die in dieser Arbeit entwickelten Konzepte unterbreiten hierfür Vorschläge. Erstmals werden für dynamische Prozessänderungen entsprechende Berechtigungskonzepte vorgestellt, die für die praktische Nutzung adaptiver Prozess-Management-Systeme unerlässlich sind.

Literaturverzeichnis

- [Bann92] Bannert, M.: *Zugriffsschutzmechanismen für den Entwurf sicherer IT-Systeme*. Diplomarbeit, Universität Karlsruhe, 1992.
- [Bark97] Barkley, J.: *Comparing Simple Role Based Access Control Models and Access Control Lists*. National Institute of Standards and Technology, August 1997.
- [BFA99] Bertino, E.; Ferrari E.; Atluri, V.: *The Specification and Enforcement of Authorization Constraints in Workflow Management Systems*. ACM Transactions on Information and System Security, Vol. 2, No. 1, Februar 1999, pp. 65-104.
- [Dada99] Dadam, R.: *Datenbanksysteme: Grundlagen, Datenmodell und Systemkonzepte – Vorlesung WS 1999/2000* –. Skript zur Vorlesung Datenbanksysteme, Abteilung Datenbanken und Informationssysteme, Universität Ulm, Oktober 1999.
- [DRK00] Dadam, P.; Reichert, M.; Kuhn, K.: *Clinical Workflows – The Killer Application for Process-oriented Information Systems?*. Ulmer Informatik-Berichte, Nr. 97-16, November 1997.
- [FCK95] Ferraiolo, D.F.; Cugini, J.A.; Kuhn, D.R.: *Role-based Access Control (RBAC): Features and Motivations*. National Institute of Standards and Technology, Proceedings 11th Annual Computer Security Applications, November 1995 (verfügbar unter: URL: <http://hissa.ncsl.nist.gov/rbac/newspaper/rbac.html> [Stand 30.3.2001]).
- [FeKu92] Ferraiolo, D.F.; Kuhn, D.R.: *Role-Based Access Control*. Reprinted from: Proceedings of 15th National Computer Security Conference, 1992.
- [FrSc97] Frank, S.; Schultheiß, B.: *Prozeßmodellierung und -steuerung mit WorkParty – ein Erfahrungsbericht*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-23, Abteilung Datenbanken und Informationssysteme, Universität Ulm, April 1997.
- [FSM96] Frank, S.; Schultheiß, B.; Mangold, R.: *Prozeßentwürfe für den Wochenbereich*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-25, Abteilung Datenbanken und Informationssysteme, Oktober 1996.
- [KaDe96] Karl, R.; Deiters, W.: *Staffware Workflow*. Studie zu Workflow-Management – Groupware Computing, Teil 2, dsk Beratungs-GmbH für Bürokommunikation und Computertechnik, Januar 1996, S. 413-474.
- [KaRa90] Karbe, B.; Ramsperger, N.: *Influence of Exception Handling on Support of Co-operative Office Work*. Proc. IFIP WG 8.4 Conf. on Multi-User Interfaces and Applications, Heraklion, Griechenland, September 1990.

- [KaRa91] Karbe, B.; Ramsperger, N.: *Concepts and Implementation of Migrating Office Processes*. In Bianer, W.; Hernandez, D. (Hrsg.): *Verteilte Künstliche Intelligenz und kooperatives Arbeiten*, Springer Verlag, 1991, S. 136-147.
- [Karb94] Karbe, B.: *Flexible Vorgangssteuerung mit ProMinanD*. In: Hasenkamp, U.; Kirn, S.; Syring, M. (Hrsg.): *CSCW – Computer Supported Cooperative Work: Informationssysteme für dezentralisierte Unternehmensstrukturen*. Addison-Wesley, 1994, S. 117-133.
- [Kony96a] Konyen, I.: *Organisations- und Ablaufstrukturen im Krankenhaus – Anforderungen, Werkzeuge und deren Anwendung*. Diplomarbeit, Abteilung Datenbanken und Informationssysteme, Universität Ulm, 1996.
- [Kony96b] Konyen, I.; Reichert, M.; Schultheiß, B.; Frank, S.; Mangold, R.: *Ein Prozeßentwurf für den Bereich der minimal invasiven Chirurgie*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-14, Abteilung Datenbanken und Informationssysteme Universität Ulm, Oktober 1996.
- [KRS96] Konyen, I.; Reichert, M.; Schultheiß, B.: *Organisationsstrukturen einer Universitätsklinik am Beispiel der Uni-Frauenklinik Ulm*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-18, Abteilung Datenbanken und Informationssysteme Universität Ulm, September 1996.
- [KSR96a] Konyen, I.; Schultheiß, B.; Reichert, M.: *Prozeßentwurf für den Ablauf einer radiologischen Untersuchung*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-15, Abteilungen Datenbanken und Informationssysteme Universität Ulm, Juli 1996.
- [KSR96b] Konyen, I.; Schultheiß, B.; Reichert, M.: *Prozeßentwurf eines Ablaufs im Labor*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-16, Abteilung Datenbanken und Informationssysteme Universität Ulm, Juli 1996.
- [LeA194] Leymann, F.; Altenhuber, W.: *Managing Business Processes as an Information Resource*. IBM Systems Journal, Vol. 33, No. 2, 1994, S. 326-348.
- [LeRo00] Leymann, F.; Roller, D.: *Production Workflow*. Prentice Hall, 2000.
- [Meye96] Meyer, J.: *Anforderungen an zukünftige Workflow-Management-Systeme: Flexibilisierung, Ausnahmebehandlung und Dynamisierung – Erörterung am Beispiel medizinisch-organisatorischer Abläufe*. Diplomarbeit, Abteilung Datenbanken und Informationssysteme, Universität Ulm, 1996.
- [NIST] National Institute of Standards and Technology, (verfügbar unter: URL: <http://hissa.ncsl.nist.gov/rbac/> [Stand 30.3.2001]).
- [Ott96] Ott, A.: *Modellierung von Rechten zur Anpassung von Workflows*. Studienarbeit, Institut für Parallele und Verteilte Höchstleistungsrechner, Universität Stuttgart, November 1996.
- [RBD] Reichert, R.; Bauer, T.; Dadam, P.: *Das ADEPT Prozess Management System: Realisierung flexibler und zuverlässiger unternehmensweiter Workflow-Anwendungen*. Proc. Elektronische Geschäftsprozesse (eBusines Processes), Klagenfurt, September 2001 (voraussichtlicher Erscheinungstermin).

- [ReDa96] Reichert, M.; Dadam, P.: *Einsatz von Workflow-Management-Systemen – Praktikumsberichte*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-8, Abteilung Datenbanken und Informationssysteme Universität Ulm, April 1996.
- [Reic00a] Reichert, M.: *Prozessorientierte Workflow-Management-Systeme. Erörterung am Beispiel von IBM MQSeries Workflow (IBM FlowMark)*. Vortrag im Rahmen des DIA-Seminars „Geschäftsprozessmodellierung und Management“, Abteilung Datenbanken und Informationssysteme, Universität Ulm, Februar 2001.
- [Reic00b] Reichert, M.: *Dynamische Ablaufänderungen in Workflow-Management-Systemen*. Dissertation, Abteilung Datenbanken und Informationssysteme, Universität Ulm, Mai 2000.
- [Reic00c] Reichert, M.: *Prozessmanagement im Krankenhaus – Nutzen, Anforderungen und Visionen*. das Krankenhaus, Vol. 92, No. 11, S. 903-909, November 2000.
- [Reic01] Reichert, M.: *Workflow-Management-Systeme. Grundlagen und Systemkategorien*. Vortrag im Rahmen des DIA-Seminars „Geschäftsprozessmodellierung und Management“, Universität Ulm, Abt. DBIS, Februar 2001.
- [Roll96] Roller, D.: *Verifikation von Workflows in IBM FlowMark*. In: Vossen, G.; Becker, J.: Geschäftsprozessmodellierung und Workflow-Management, Thomson Publ., 1996, S. 353-368.
- [RSD97] Reichert, M.; Schultheiß, B.; Dadam, P.: *Erfahrungen bei der Entwicklung vorgangsoientierter, klinischer Anwendungssysteme auf Basis prozessorientierter Workflow-Technologie*. Proc. GMDS'97, Ulm, September 1997, S. 181-187.
- [Rupi97] Rupietta, W.: *Business Processes and Workflow Management*. In: Bernus, P.; Mertins, K.; Schmidt, G. (Hrsg.): Handbook on Architectures of Information Systems, Springer-Verlag, 1997.
- [Saue96] Sauer, U.: *Entwicklung einer Endbenutzerschnittstelle für die flexible Bearbeitung medizinisch-organisatorischer Prozesse*. Diplomarbeit, Universität Ulm, 1996.
- [Schu96a] Schultheiß, B.: *Prozessreengineering in klinischen Anwendungsumgebungen – Beispiele, Vorgehensmodelle, Werkzeuge*. Diplomarbeit, Abteilung Datenbanken und Informationssysteme, Universität Ulm, 1996.
- [Schu96b] Schultheiß, B.; Meyer, J.; Mangold, R.; Zemmler, T.; Reichert, M.: *Prozessentwurf für den Ablauf einer stationären Chemotherapie. Ergebnisse einer Analyse an der Universitätsfrauenklinik Ulm*. Interne Ulmer Informatik-Berichte (Software-Labor), DBIS-5, Abteilung Datenbanken und Informationssysteme Universität Ulm, März 1996.
- [Weil97] Weilbach, P.: *Implementierungsaspekte zur Verwaltung und Synchronisation dynamischer Änderungen in prozessorientierten Workflow-Management-Systemen*. Diplomarbeit, Universität Ulm, 1997.
- [Zeit99] Zeitler, J.: *Integration von Verteilungskonzepten in ein adaptives Workflow-Management-System*. Diplomarbeit, Universität Ulm, 1999.

Name: Sparr, Sonja

Matrikel-Nr.: 0366427

Erklärung

Ich erkläre, dass ich die Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

(Unterschrift)