Entwicklung und Implementation eines Rollenwerkzeugkastens

Jan Raap

Studienarbeit
am
Fachbereich Informatik,
AB Softwaretechnik,
Universität Hamburg
Betreuer: Dr. Guido Gryczan

Inhaltsverzeichnis

Kapitel 1

Einführung

Ziel dieser praktischen Studienarbeit ist die Entwicklung und Implementierung eines Rollenwerkzeugkastens. Seine möglichen Einsatzumgebungen sollen im Rahmen dieses Textes vorgestellt werden. Des weiteren werden die Komponenten des Rollenwerkzeugkasten präsentiert. Dieses Kapitel legt die Motivation der Arbeit dar.

1.1 Motivation

Die zunehmende Verbreitung von Computern hat zur Folge, daß es zu einer Verschiebung bezüglich der ursprünglichen EDV-Aufgabengebiete kommt. Anstatt nur als einfaches Werkzeug für Textverarbeitung oder Tabellenkalkulation zu dienen, gewinnt der Computer, insbesondere durch Vernetzung, einen immer stärker werdenden Einfluß auf Arbeitprozesse. Neben höherer Effizienz und besseren Kontrollmöglichkeiten, sind vor allem die Rationalisierungsaspekte in diesem Zusammenhang eine treibende Kraft. Gestützt wird diese Entwicklung durch den Fortschritt im Bereich des „papierlosen Büros“. So werden vermehrt digitale Formulare statt der herkömmlicher Papierformulare verwendet. Durch die Verbindung diese beiden Aspekte, Arbeitsbeeinflussung und digitale Formulare, erschließen sich vielseitige Möglichkeiten.

Eine Entwicklung, die diese beiden Aspekte aufgreift, ist *Workflow-Management* (WF). Dabei wird versucht, Geschäftsprozesse digital als sogenannte *Workflows* nachzugebilden. Mit Hilfe eines *Workflow-Management-Systems* (WFMS)

werden diese Workflows dann ausgeführt, wobei hervorgehoben werden muß, daß Steuerung und Kontrolle beim WFMS liegen – somit also eine Maschine den Arbeitsablauf des Menschen bestimmt.

Prozeßmuster stellen einen anderen Ansatz dar. Anstatt den Menschen zu kontrollieren, soll hier eine Unterstützung seiner Arbeit erfolgen. Es handelt sich hierbei um gemeinsame Materialien, welche kooperative Arbeitsprozesse vergegenständlichen und die sich durch ihre Adaptierbarkeit auszeichnen, d.h. der Benutzer kann Prozeßmuster der Situation entsprechend anpassen. Eingesetzt werden Prozeßmuster in Verbindung mit (digitalen) *Vorgangsmappen*, die eine gewisse Ähnlichkeit mit den bekannten Gittermappen[1] haben.

Gemeinsamen ist beiden Ansätzen die Verwendung von *Rollen* bei der Modellierung von Geschäftsvorgängen. Um eine möglichst große Flexibilität zu erreichen, werden bei der Modellierung von Geschäftsprozessen Rollen anstelle von konkreten Personen verwendet. Eine *Rolle* beschreibt dabei eine Menge von Fähigkeiten und Kompetenzen, die alle Rollenträger aufweisen müssen, die diese Rollen spielen sollen ([Jab95], S. 17). Zum Beispiel repräsentiert die Rolle *Kreditsachbearbeiter* sämtliche Mitarbeiter, die die entsprechenden Qualifikationen besitzen, um als Kreditsachbearbeiter zu arbeiten.

In diesem Zusammenhang steht auch der Begriff der *Rollenauflösung*. Gemeint ist hiermit der Vorgang, während der Ausführung von Workflows bzw. Prozeßmustern für eine abstrakte Rolle einen konkreten Rollenträger zu bestimmen. Dieser Rollenträger ist dann für die Bearbeitung des nächsten Arbeitsschrittes zuständig.

Bei der Rollenauflösung handelt es sich um die Hauptfunktionalität des im Rahmen dieser Studienarbeit entwickelten und implementierten Rollenwerkzeugkastens. Bewerkstelligt wird dies durch den *Rollenauflöserautomaten*. Zu den weiteren Bestandteilen des Anwendungssystems gehören der *Automateneinsteller*, der *Rollenverwalter*, der *Rollenanmelder* und der *Rollentester*.

[1] Solche Gittermappen werden häufig in Behörden verwendet. Auf der Vorderseite befindet sich ein Gitter, in dessen Felder der jeweils nächste Empfänger der Mappe eingetragen wird.

1.2 Aufbau der Arbeit

Das Kapitel 2 stellt die Softwareentwicklungsmethodik *WAM* vor. Es wird auf das *Leitbild vom Arbeitsplatz für qualifizierte menschliche Tätigkeit* eingegangen und die Entwurfsmetaphern *Werkzeug*, *Automat* und *Material* erläutert. Die im Zusammenhang mit WAM verwendeten Dokumente *Szenarios*, *Glossare* und *Systemvisionen* werden beschrieben.

Die folgenden beiden Kapitel zeigen mögliche Anwendungsfelder für den Rollenwerkzeugkasten auf. Kapitel 3 geht dabei auf die Entstehung von Workflows und Workflow-Management-Systemen ein. Des weiteren wird das dazugehörige Rollenverständnis präsentiert.

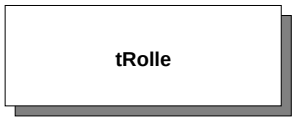
Kapitel 4 diskutiert zuerst die ablaufsteuernde und dann die unterstützende Sichtweise, um darauf aufbauend Prozeßmuster und Vorgangsbearbeitungssysteme einzuführen. Hierbei ergibt sich ein anderes Rollenverständnis.

In Kapitel 5 wird auf den praktischen Teil dieser Arbeit eingegangen. Es werden die verschiedenen Bestandteile des Rollenwerkzeugkastens vorgestellt und ihre Funktionalität, Aufbau und Zusammenspiel beschrieben. Das im vorherigen Kapitel präsentierte Vorgangsbearbeitungssystem bildet den Einsatzrahmen des hier entwickelten Rollenwerkzeugkastens.

Abschließend werden in Kapitel 6 die wesentlichen Punkte dieser Arbeit zusammengefaßt und kritisch betrachtet. Mit einem Ausblick auf zukünftige Entwicklungen schließt diese Arbeit.

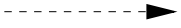
1.3 Graphische Konventionen

Folgende graphische Konventionen werden im Rahmen dieser Arbeit verwendet:



Klassen werden durch Rechtecke dargestellt.

Die Benutzt-Beziehung wird durch einen einfachen Pfeil dargestellt, wobei die Pfeilspitze auf die benutzte Klasse zeigt.



Die Beobachtet-Beziehung wird durch einen gestrichelten Pfeil dargestellt, wobei die Pfeilspitze auf die beobachtete Klasse zeigt.

Kapitel 2

Software-Entwicklung nach WAM

Für die Erstellung des im Rahmen dieser Studienarbeit vorgestellten Rollenwerkzeugkastens (siehe Kapitel 5) wurde die Software-Entwicklungmethodik WAM verwendet, welche am Arbeitsbereich Softwaretechnik gelehrt wird. Das Akronym WAM stehen als Abkürzung für die Begriffe Werkzeug, Aspekt, Material. Anstelle von Aspekt wird auch der Begriff Automat verwendet.

Softwareentwicklung, die nach dieser Methode erfolgt, ist objektorientiert und evolutionär. Sie verwendet allgemeinverständliche Metaphern um interaktive Anwendungssysteme zu erstellen. Das Begriffsverständnis aus dem fachlichen Entwurf, welcher mittels dieser Metaphern erstellt wurde, kann ohne „Bruch“ in ein technisches Modell überführt werden ([KGZ94], S. 20). Es findet somit eine Verschmelzung von zwei Bereichen statt, die bisher getrennt waren: die Analyse des Anwendungsgebiets und die Modellierung des DV-Systems, das in diesem Gebiet eingesetzt werden soll ([KGZ94], S. 1).

Im Zusammenhang mit WAM steht das *Leitbild vom Arbeitsplatz für qualifizierte menschliche Tätigkeit* im Mittelpunkt. Hierbei werden Benutzer „als Experten ihres Anwendungsbereiches verstanden, die über das notwendige Wissen und ausreichende Erfahrung verfügen, um ihre Arbeit selbständig und qualifiziert zu erledigen“([Wul95], S. 5).

2.1 Die Entwurfsmetaphern Werkzeug und Material

Die Analyse und Beschreibung des Anwendungsbereiches erfolgt mit Hilfe der Entwurfsmetaphern Werkzeug und Material. Der Grund hierfür liegt in dem Bezug zu Arbeitsmitteln und -gegenständen, die Menschen bei ihrer alltäglichen Arbeit benutzen. Arbeitsmittel werden dabei als Werkzeuge abgebildet, mit denen Arbeitsgegenstände (Materialien) sondiert und verändert werden können. Das Zusammenspiel zwischen Werkzeug und Material wird durch Aspekte ausgedrückt.

Ob ein Objekt Material oder Werkzeug ist, hängt vom jeweiligen Kontext ab. So ist zum Beispiel ein Bleistift ein Werkzeug, wenn man damit schreibt. Wird dieser Bleistift aber angespitzt, so stellt er ein Material dar. Durch die Verwendung von Aspekten wird der Zusammenhang zwischen den Objekten Anspitzer, Bleistift und Papier deutlich. Das Werkzeug Anspitzer ist mit dem Material Bleistift über den Aspekt *anspitzbar* verbunden, während derselbe Bleistift, als Werkzeug betrachtet, mit dem Material Papier über den Aspekt *beschreibbar* verbunden ist.

Durch die Entwurfsmetaphern Werkzeug und Material wird eine Verbindung von den bisher am Arbeitsplatz vorhandenen gewohnten Gegenständen zu den neuen Komponenten eines Anwendungssystems geschaffen ([KGZ94], S. 25). Dies erleichert es dem Benutzer sich in dem neuen Anwendungssystem zurechtzufinden.

Die bereits beschriebenen Metaphern Werkzeug und Material werden durch weitere Metaphern ergänzt. Dies sind zum Beispiel die Metaphern Automat (siehe auch Kapitel 2.2) und Umgebung. In [Gry95] wird genauer auf diese Metaphern eingegangen.

Nach Kilberth et al. kann die Strukturierung eines Anwendungssystems auf der Ebene von Werkzeugen noch weiter betrieben werden, indem eine Trennung nach Funktionalität und Handhabung vollzogen wird. Das Ergebnis dieses Verfeinerungsprozesses wird in Form von Funktions- und Interaktionskomponente dargestellt. Die Abhängigkeit von der Systemumgebung, z.B. vom Fenstersystem, wird hierdurch auf die Interaktionskomponente reduziert. Kilberth et al. liefern ebenfalls die Begründung, weshalb die technische Teilung auch konzeptionell Bestand hat. „Die Funktionskomponente definiert den Zweck eines Werkzeugs unter

Berücksichtigung der verwendeten Materialien. Sie charakterisiert im jeweiligen Arbeitskontext, *was* ein Werkzeug leistet. Die Interaktionskomponente charakterisiert, *wie* ein Werkzeug dargestellt und bedient wird"([KGZ94], S. 38).

2.2 Die Automatenmetapher

Die Entwurfsmetapher Automat stellt eine Ergänzung der oben eingeführten Metaphern Werkzeug und Material dar. Aufgabe eines Automaten ist es, lästige und formalisierbare Routinetätigkeiten von geringer Komplexität zu übernehmen. Dabei produzieren Automaten vorab festgelegt Ergebnisse. Einmal gestartet laufen sie über längere Zeiträume, ohne daß äußere Eingriffe seitens des Benutzers vonnöten sind, und stellen während dieser Zeit ihre Dienste zur Verfügung. Wie auch bei den Werkzeugen, werden Automaten über Aspekte mit Materialien versorgt. Ein ihnen übergebenes Material bearbeiten sie auf Basis ihres fachlichen Algorithmus. Automaten verfügen über keine eigene Interaktionskomponente, sondern werden mittels eines Einstellwerkzeug kontrolliert, welches auch der Veränderung von Einstellungen dient.

Im Kontext mit WAM werden Automaten nie als technischer Ersatz für qualifizierte menschliche Tätigkeit verwendet; sie können allenfalls dem Menschen lästige Handlungssequenzen zusammenfassen, die auch ohne seinen Eingriff ausführbar sind ([Wul95], S.16). Automaten sind Arbeitsmittel, deren Ergebnisse wieder in menschliches Handeln eingebettet werden.

2.3 Entwicklungsdokumente

Um auf die oben beschriebenen Entwurfsmetaphern Werkzeug und Material im Anwendungskontext zu kommen, werden in Zusammenarbeit mit dem Anwender eine Reihe unterschiedlicher Dokumente erarbeitet. Zu den Dokumenten der WAM-Methode gehören *Szenarios*, *Glossare* und *Systemvisionen*.

Szenarios sind Analysedokumente und dienen der Beschreibung von alltäglichen Arbeitsvorgängen, wie sie vor dem Einsatz der zu entwickelnden Software ablaufen. Sie erfassen somit den Ist-Zustand. Die von den Entwicklern in der Fachsprache der Anwendung erstellten Dokumente werden von den Benutzern

überprüft (Autor-Kritiker-Zyklus) und bilden „eine Grundlage für Diskussion und Entwicklung eines gemeinsamen Verständnisses “([Rie95], S. 8).

Um Mißverständnissen durch implizite Annahmen vorzubeugen, werden beleitend Glossare erstellt, in denen die Bedeutung und der Zusammenhang von verwendeten Begriffen festgehalten werden.

Mit den Systemvisionen wird das zukünftige System beschrieben. Wie auch die Szenarios, werden diese Dokumente von den Entwicklern verfaßt. Systemvisionen werden aber nur für die Diskussion unter Entwicklern verwendet und helfen dabei, „die Möglichkeiten einer Computerunterstützung für die in Szenarios modellierten Arbeitsabläufe abzuschätzen“([KGZ94], S. 100).

Kapitel 3

Workflow und WFMS

Dieses Kapitel beschäftigt sich mit dem Workflow-Begriff und seinem Einsatzrahmen, dem Workflow-Management-System. Dabei wird zuerst auf das Workflow-Verständnis nach S. Jablonski eingegangen. Als zweites wird mit Hilfe von ausgewählten Beispielen die Verwendung von Workflows gezeigt. Drittens werden Kritikpunkte am Workflow-Konzept hervorgehoben.

3.1 Motivation

Eines der Hauptziele eines jeden Wirtschaftsunternehmens ist es, seine Wettbewerbsfähigkeit zu steigern. Hierbei bieten insbesondere Geschäftsprozesse (business processes) Ansatzpunkte. Dies „sind abteilungsübergreifende, aber fachlich zusammenhängende Aktivitäten, die in logischen und zeitlichen Abhängigkeiten zueinander stehen“(Workflow Management Coalition, aufgeführt in [Ver95a], S. 156). Untersuchungen haben gezeigt, daß Geschäftsprozesse nur zu einem geringen Teil produktiv sind. Als Gründe werden dafür ineffektive Arbeitsteilung und deren Ablauflogik genannt. Hier bieten sich Verbesserungsmöglichkeiten zur Steigerung der Produktivität und somit der Wettbewerbsfähigkeit an. Dies kann unter anderem durch die Optimierung und flexiblere Gestaltung von Geschäftsprozessen erreicht werden.

3.2 Workflow-Management-Systeme

Bei Workflow-Management-Systemen (WFMS) handelt es sich um spezielle Software-Lösungen, die Workgroup Computing ermöglichen sollen. Sie unterstützen die Koordination, Steuerung und Überwachung von Benutzern, die an der Lösung von räumlich verteilten Aufgaben arbeiten ([Jab95], S. 13). WFMS kommen als Ausführungssysteme für Geschäftsprozesse in Frage. Hierfür werden Geschäftsprozesse in Workflows transformiert, die dann vom WFMS ausgeführt werden.

Bei McCarthy und Bluestein findet sich folgende Definition von Workflow-Management-System ([MB91], zitiert nach [Jab95], S. 14):

> Workflow management software is a proactive computer system which manages the flow of work among participants, according to a defined procedure consisting of a number of tasks. It coordinates user and system participants, together with the appropriate data resources, which may be accessible directly by the system or off-line, to achieve defined objectives by set deadlines. The coordination involves passing tasks from participant to participant in correct sequence, ensuring that all fulfil their required contributions, taking default actions when necessary.

Kern dieser Definition ist ein vorgegebenes Verfahren, welches aus einer Anzahl von Aufgaben besteht. Anhand dieses Verfahrens werden Benutzer und Daten koordiniert, um ein vorgegebenes Ziel zu erreichen. Dabei überwacht und kontrolliert das System den korrekten Ablauf. Bei dem erwähnten Verfahren handelt es sich um einen Workflow (WF).

Jablonski setzt zusätzlich drei Merkmale für den Einsatz von WFMS voraus ([Jab95], S. 14). So fordert er *Skalierbarkeit*, um für zukünftigen Zuwachs an Benutzern und Workflows gewappnet zu sein. Außerdem muß es die Möglichkeit geben, bestehende Softwaresysteme (legacy software) in das WFMS zu integrieren (*Integration von Alt-Software*). Und letztlich soll das WFMS vor dem Benutzer verbergen, daß es sich um verteilte und heterogene Hard- und Softwareumgebungen handelt (*Transparenz*).

In [Bus95] werden noch weitere Eigenschaften von WFMS genannt. So verfügt jeder Benutzer, der mit dem WFMS verbunden ist, über eine private *Arbeitsliste*

(private worklist). Jeder Eintrag in der Liste repräsentiert eine Aufgabe, die dem Benutzer übertragen wurde. Wählt der Benutzer einen Eintrag aus seiner Arbeitsliste aus, so wird vom WFMS automatisch das entsprechende Anwendungsprogramm gestartet, welches der Benutzer benötigt um die Aufgabe zu erledigen. Somit sind alle Voraussetzung erfüllt damit der Benutzer dann sofort mit der Arbeit beginnen kann.

Mit der Einführung von WFMS werden gewisse Anforderungen und Erwartungen verbunden. Laut [BJ94a] lassen sich im Wesentlichen drei Punkte hervorheben. Zuerst einmal wäre da die *Erhöhung der Produktivität*. Dies wird durch die Automatisierung von Standardprozessen und die Erledigung von Routinearbeit durch den Computer erreicht. Zweitens wird eine *Verbesserung der Qualität* angeführt. Ein Computersystem überwacht dabei die Prozeßausführung, wodurch z.B. das ‚Vergessen' obligatorischer Arbeitsschritte verhindert wird. Und als dritter Punkt wird die *Erhöhung der Arbeitsflexibilität* genannt. Das Überarbeiten von Prozessen soll schneller und einfacher bewältigt werden. Dadurch können veränderte Marktanforderungen besser und schneller berücksichtigt werden.

Abschließend soll noch hervorgehoben werden, daß es durch die Einführung eines WFMS zu einer Verlagerung der Entscheidungsbefugnisse kommt. Statt wie bisher selbständig über den Arbeitsablauf entscheiden zu können, muß der Benutzer sich nun dem WFMS anpassen. War der Benutzer früher in der Lage eigenständig auf Ausnahmen (d.h. Situationen, die nur sehr selten eintreten) zu reagieren, so muß er jetzt warten, bis diese ‚neuen' Situationen in Form veränderter oder neuer Workflows im System erfaßt worden sind. Als weiteres Beispiel für die Reichweite einer WFMS-Einführung kann der Fall der automatischen Werkzeugwahl angesehen werden, wo das System entscheidet, mit welchem Werkzeug eine Aufgabe zu erledigen ist.

Folgende Definition von WFMS wird im Rahmen dieser Arbeit verwendet:

> Ein System, das Workflows ausführt und dabei sicherstellt, daß dies korrekt geschieht.

3.3 Workflow

Die Workflow Management Coalition beschreibt Workflow als „computergestützte Vereinfachung oder Automatisierung eines gesamten Geschäftsprozesses oder

eines Teils davon "([Ver95a], S. 156). Auch S. Jablonski verwendet Geschäftsprozesse für die Definition von Workflow. In [Jab95](S. 15) bezeichnet er „Workflows als das ausführbare Abbild von Geschäftsvorgängen".

Für die Modellierung von Workflows werden in der Literatur vier Aspekte als wesentlich angesehen. Bei [Wul95] und [Bus95] sind dies der Aktivitäts- (functional), Aktoren- (organisational), Ressourcen- (informational) und Abhängigkeitsaspekt (behavioral aspect). Statt des Ressourcenaspekts wird in [Jab95] der kausale Aspekt angegeben. Die Bedeutung dieser Aspekte und wofür sie stehen wird im folgenden erläutert.

1. Der *aktivitätsbezogene Aspekt* beschreibt, *was* getan werden muß, ohne zu sagen, wie, von wem oder mit welchen Daten ([Bus95]). In diesem Zusammenhang wird das Konzept von *komplexen* und *elementaren* Workflows eingeführt. Bei elementaren Workflows handelt es sich um Verweise auf Applikationen, mit denen die eigentliche Funktionalität erbracht werden soll. Komplexe Workflows können sowohl aus elementaren Workflows, als auch aus anderen komplexen Workflows bestehen. Auf diese Art und Weise entsteht eine Hierarchie mit einem *Topworkflow* (Top-Level-Workflow) an oberster Stelle. Verweist ein Workflow auf einen anderen, so wird der verweisende Workflow als *Superworkflow*, und der Workflow, auf den verwiesen wird, als *Subworkflow* bezeichnet.

2. Die Festlegung, *wer* die Ausführung eines elementaren Workflows übernimmt, geschieht durch den *aktorenbezogenen Aspekt*. In Frage kommen dabei sowohl menschliche als auch nicht-menschliche (z.B. Maschinen oder Programmsysteme) Aktivitätsträger. Sie werden unter dem Begriff *Aktoren* zusammengefaßt. Die statische Zuordnung von Aktoren zu elementaren Workflows hat sich als nicht flexibel genug herausgestellt, da dies ein häufiges Ändern der Workflowspezifikation zur Folge hatte. Aus diesem Grund wurde das Rollenkonzept eingeführt. Eine *Rolle* beschreibt dabei eine Menge von Fähigkeiten und Kompetenzen, die alle Aktoren (Rollenträger) aufweisen müssen, die diese Rolle spielen sollen ([Jab95], S. 17). Anstelle von konkreten Aktoren werden dann Rollen den elementaren Workflows zugeordnet. Im Kapitel 3.5 wird näher auf Rollen und Rollenauflösung eingegangen.

3. Der *resourcenbezogene Aspekt* bestimmt, *welche* Informationen(Dokumente) bei der Ausführung eines elementaren Workflows benötigt bzw. erzeugt werden ([Wul95]).

4. (a) Der *Abhängigkeitsaspekt* regelt die Ausführungsreihenfolge von Subworkflows, aus denen sich ein komplexer Workflow zusammmensetzt. In [Jab95] werden zwei mögliche Spezifikationen hierfür vorgestellt. Dies ist einerseits die *präskriptive* Ablaufkontrolle. Alternative Ausführungsreihenfolgen werden hier nur über die Auswertung von vorgegebenen Ablaufbedingungen zugelassen. Wird stattdessen eine Menge *äquivalenter* Ausführungsreihenfolgen unterstützt, spricht man von *deskriptiver* Ablaufkontrolle.

   (b) Durch den *kausalen* Aspekt wird festgelegt, *warum* ein Workflow ausgeführt wird (Kausalität). Diese Information kann über den eigentlichen Kontext des Workflows hinaus verwendet werden. Wird zum Beispiel ein Projekt X eingestellt, lassen sich durch Daten über den Sinn und Zweck eines Workflows (hier: bezieht sich auf Projekt X), alle zu Projekt X gehörigen Workflows bestimmen und können dann „kontrolliert“ beendet werden ([Jab95]).

Neben diesen vier Hauptaspekten wird in der Literatur ein weiterer erwähnt, der *Historie Aspekt*. Hierbei werden die Informationen spezifiziert, die während der Ausführung eines Workflows archiviert werden sollen, z.B. in einer Log-Datei. Dies dient einerseits der Sicherheit; andererseits stellen diese Daten auch eine wertvolle Informationsquelle dar.

Abschließend soll Workflow für das zukünftige Verständnis wie folgt definiert werden:

> Workflows sind ausführbare Abbilder von Geschäftsprozessen.

3.4 Beispiel

Das nun folgende Beispiel stammt aus [Jab95]. Es beschreibt den (vereinfachten) komplexen Workflow *Reisekostenabrechnung* (siehe Abbildung 3.1/[Jab95], S. 16, Abb. 1). Dieser Workflow besteht aus den fünf elementaren Workflows *Ausfüllen,*

Genehmigen, Bearbeiten, Auszahlen und *Ablegen*. Wie bereits angeführt (siehe Kapitel 3.3), verweisen elementare Workflows auf Applikationen (gepunktete Linie). In diesem Fall sind dies ein Reisekostenabrechnungs-, Buchungs- und Archivsystem. Desweiteren ist jedem elementaren Workflow eine Rolle zugeordnet, welche stellvertretend für alle Aktoren steht, die ihn bearbeiten können. In Kapitel 3.5 wird noch ausführlicher auf Rollen und ihre Auflösung eingegangen.

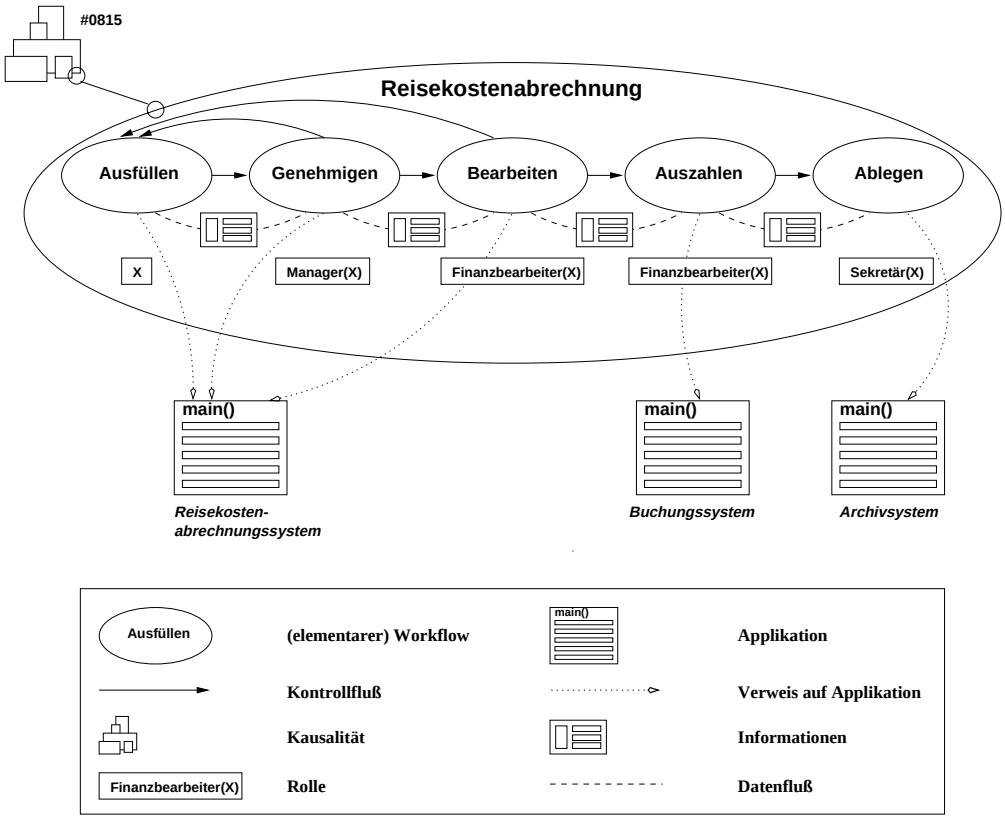


Abbildung 3.1: Beispiel Reisekostenabrechnung

Die Abarbeitung des komplexen Workflows *Reisekostenabrechnung* beginnt mit der Ausführung des ersten elementaren Workflows (*Ausfüllen*). Mit Hilfe des Reisekostenabrechnungssystems gibt der Antragsteller X die entsprechenden Informationen bzgl. der Reise ein, für die eine Kostenerstattung erfolgen soll. Anhand der einfachen Pfeile (mit schwarzem Pfeilkopf) wird nun der nächste auszuführende elementare Workflow bestimmt. Dabei besteht die Möglichkeit, wie z.B. im Fall *Genehmigen* zu sehen ist, daß der Kontrollfluß auch wieder zurückfliessen kann. Gründe hierfür können unvollständige oder inkonsistente In-

formationen sein. Die Weitergabe von Informationen zwischen Workflows läßt sich an den gestrichelten Linien erkennen (Datenfluß). Leider wird aus der Originalabbildung nicht ersichtlich, in welche Richtung der Datenfluß fließt. Es ist aber wahrscheinlich, daß diese mit der Richtung des Kontrollflußes übereinstimmt. Ebenfalls unklar ist, weshalb beim Kontrollrückfluß keine Informationen mit übergeben werden. Der Kontext, in dem der komplexe Workflow eingebettet ist, wird hier durch *#0815* dargestellt (Kausalität).

3.5 Rollen und Rollenauflösung

Für die Ausführung von Workflows werden Aktoren benötigt um Aufgaben zu übernehmen. Die Auswahl der Aktoren geschieht dabei unter fähigkeitsbezogenen Gesichtspunkten. Dies bedeutet, daß nur derjenige Aktor eingesetzt werden kann, der auch für die Durchführung der entsprechenden Aufgabe qualifiziert ist. In der Abbildung 3.2 Zeile a) (aus [Jab95], S. 17, Abb. 3) sind dies Bert, Frank und Petra, die den elementaren Workflows *Ausfüllen, Genehmigen* und *Bearbeiten* zugeordnet werden. Solch eine statische Zuordnung von konkreten Aktoren zu Workflows hat jedoch den Nachteil, daß sie unflexibel ist. Kommt es zu einer personellen Veränderung (z.B. Frank verläßt die Organisation und Martin übernimmt dessen Position), so müssen sämtliche Workflows, die von dieser Veränderung betroffen sind, erneut bearbeit werden, um den neuen Gegebenheiten zu genügen.

Wegen dieses hohen Anpassungsaufwandes wurde das Konzept der *Rolle* eingeführt. Wie bereits erwähnt, beschreibt eine Rolle eine Menge von Fähigkeiten/Kompetenzen, die alle Aktoren (Rollenträger) aufweisen müssen, die diese Rolle spielen sollen ([Jab95], S. 17). Dies bedeutet für die Workflow-Modellierung, daß nicht mehr die Aktoren, sondern die notwendigen Qualifikationen für die Durchführung der entsprechenden Aufgabe im Vordergrund stehen. Und diese Qualifikationen werden durch das Konzept der Rolle repräsentiert. Ersetzt man die statischen Aktoren aus Abb. 3.2 Zeile a) durch Rollen, wie dies in Abb. 3.2 Zeile b) geschehen ist, haben personelle Veränderung keine Auswirkungen mehr auf diese Workflow-Spezifikation. Statt dessen muß nur noch an einer Stelle diesen Veränderungen Rechnung getragen werden, nämlich bei der Zuordnung von Rollenträgern zu den jeweiligen Rollen.

Aber auch die Verwendung von einfachen Rollen läßt noch Raum für Verbes-

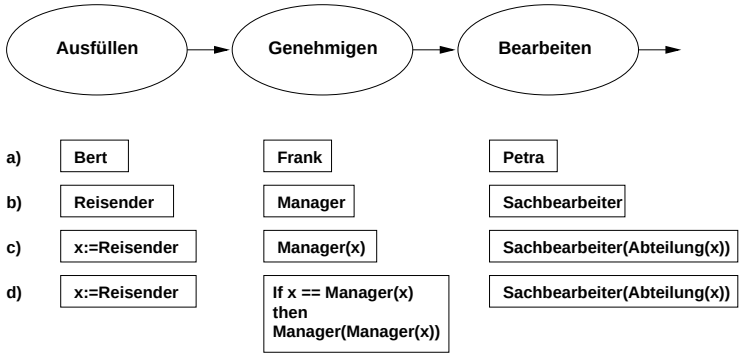


Abbildung 3.2: Szenarien zur Rollenauflsung

serungen. So wäre es vorstellbar, daß für die Rolle *Manager* in Abb. 3.2 Zeile b) sämtliche Manager eines Unternehmens in Frage kommen würden und somit alle (abteilungsübergreifend) die Genehmigung erteilen könnten. Da dies aber nur wenig Sinn macht, bietet es sich an, Rollen in einen organisationsbezogenen Kontext einzubetten. Dabei wird die Menge der Rollenträger „durch Prädikate über organisatorische Strukturen eingeschränkt“([Jab95], S. 17). Diese erweiterten Rollenbeschreibungen werden als *organisationsbezogene Rollen* bezeichnet([Jab95], S. 17). Bei den Prädikaten handelt es sich um organisationsbezogene Aussagen, durch welche die bisher organisationsunabhängigen Rollen in einen organisatorischen Kontext eingebettet werden. Für die Abb. 3.2 Zeile c) bedeutet dies, daß nur noch Manager von X die Genehmigung erteilen dürfen, also Manager, die zur selben Abteilung wie X gehören. Die gleiche Einschränkung gilt auch für die Auswahl des entsprechenden Sachbearbeiters.

Trotz der relativ umfassenden Möglichkeiten der organisationsbezogenen Rollen können Situation entstehen, in denen einfache Prädikate nicht mehr ausreihend bzw. mächtig genug sind. Als Beispiel ist folgende Situation vorstellbar: Manager Frank ist für eine gewisse Zeit nicht verfügbar, da er Urlaub hat. Während dieser Zeit wird Bert stellvertretend Abteilungsmanager. Tätigt er im Verlauf seiner Vertreterzeit eine Dienstreise, so wäre es ihm möglich dank seiner Stellvertreterfunktion seine eigene Reisekostenabrechnung genehmigen. Um dies zu verhindern, werden *organisatorische Zuordnungsstrategien* eingeführt. In Abb. 3.2 Zeile d) wird solch eine Zuordnungsstrategie verwendet, mit der Folge, daß der Antrag eines Managers immer von seinem übergeordneten Manager genehmigt werden muß. Dies geschieht mit Hilfe einer Fallunterscheidung, die

überprüft, ob der Antragsteller ein Manager ist. Trifft dies zu, kommt er nicht als Rollenträger in Frage, da ansonsten die Gefahr einer Selbstgenehmigung besteht ([Jab95], S. 17-18).

Der anschliessende Text wird auf folgender Rollen- und Rollenauflösungsdefinition aufsetzen:

> Eine Rolle steht stellvertretend für eine Menge von Aktoren. Voraussetzung für die Zugehörigkeit zu einer Rolle sind entsprechende Fähigkeiten und Kompetenzen.
>
> Der Vorgang der Auswahl eines einzelnen Aktoren aus der Menge aller Rollenträger einer Rolle wird als Rollenauflösung bezeichnet. Durch hinzufügen von Prädikaten oder sprachlichen Konstrukten kann die Menge der potentiellen Rollenträger zusätzlich eingeschränkt werden.

3.6 Schlußbetrachtung und Kritik

WFMS stellen einen interessanten Ansatz zur maschinellen Unterstützung bzw. Ausführung von Geschäftsprozessen dar. Verbunden ist der Einsatz solcher Systeme allerdings mit erheblichen Anfangsaufwand. So weist Versteegen in [Ver95b] darauf hin, daß sich die Einführung von WFMS ohne externe Beratung nur schwer durchführen läßt. Eines der Hauptprobleme ist dabei die Modellierung eines lebendigen Unternehmens. Dies vollständig und korrekt durchzuführen ist extrem schwierig, da sich ein Großteil der Geschäftsprozesse ständig wandelt ([Ver95b], S. 142). Schätzung, wonach WFMS bis zu 1.000 verschiedene Workflowtypen verwalten müssen können, die von bis zu 10.000 Benutzern bedient werden ([Jab95], S. 15), machen deutlich, welche Herausforderungen hier warten. Ist solch ein WFMS erst einmal installiert, stellt sich die Frage nach der Wartbarkeit. Inwieweit der einzelne Benutzer nach Einführung des Systems Eingriffsmöglichkeiten hat, und wenn ja, wie weit diese reichen konnte leider nicht festgestellt werden. Berücksichtigt man aber, daß für die Betreuung solch großer Systeme normalerweise sehr umfangsreiches Wissen vonnöten ist, so läßt dies den Schluß zu, daß der einzelne Benutzer nur über geringe Eingriffsmöglichkeiten verfügen kann. Statt dessen erscheint es wahrscheinlich, daß Eingriffe in das WFMS nur durch

qualifiziertes Fachpersonal geschehen. Dadurch kann es aber zu Engpässen kommen, wenn die Anzahl entsprechender Fachkräfte zu klein gewählt wurde. Bei den erwähnten Eingriffen kann es sich um das Erstellen oder Verändern von Workflows handeln. Verzögerungen können entstehen, wenn ein Benutzer die Notwendigkeit einer Änderung eines Workflows feststellt und anstatt selber tätig werden zu können dies durch jemand anderes machen lassen muß, da ihm die nötigen Eingriffmöglichkeiten fehlen.

Weitere Probleme bei der Modellierung können durch ‚falsche' Angaben entstehen. So neigen Mitarbeiter dazu, wenn sie gefragt werden, wie lange sie für die Bearbeitung einer Aufgabe brauchen, großzügigere Angaben zu machen. Die so entstehenden Workflows entsprechen als Folge hiervon nicht den ursprünglichen Geschäftsprozessen, die sie ja eigentlich abbilden sollen.

Mit einem anderen Aspekt der Modellierungsproblematik setzt sich Lucy Suchman in [Suc95] auseinander. Sie beschäftigt sich dabei mit den unterschiedlichen Sichtweisen und Interpretationen bezüglich der Darstellung von Arbeitsabläufen. Als Beispiel führt sie die Erfahrungen von Brun-Cottan und Wall an, welche diese mit einer auf Video aufgezeichneten Arbeitstätigkeit gemacht haben.

Im Rahmen der Aufzeichnung wurde die Arbeit eines erfahrenen Benutzers gezeigt, der einen automatischen Papiereinzug verwendet, um Dokumente einzuscannen. Hierbei war zu erkennen, daß der Benutzer bei einer falsch ausgerichteten Seite eines mehrseitigen Dokuments, diese Seite manuell neu ausrichtete, um anschließend das gesamte Dokument erneut zu scannen. Entsprechende Funktionen der Software, die bereits gescannte Seite digital korrekt auszurichten, benutzte er dabei nicht. Von Seiten der Software-Entwickler wurde dieses Verhalten als Versagen des Benutzers, die von der Software angebotenen Möglichkeiten zu wertzuschätzen, gewertet.

Brun-Cottan und ihre Arbeitskollegen stellten aber fest, daß die Vorgehensweise des Benutzers im Kontext seiner Arbeitserfahrung weitaus vernünftiger war. Die Verwendung der Software-Funktion zum korrekten Ausrichten der Seite hätte zwar das erneute Scannen des gesamten Dokuments vermieden, aber mehr Zeit und Aufmerksamkeit seitens des Benutzers erfordert. Während das erneute Scannen zwar mehr Maschinenzeit benötigt, ermöglicht es dabei jedoch dem Benutzer sich um andere Dinge zu kümmern.

Dieses Beispiel zeigt, wie eine Videoaufzeichnung, von der eigentlich ange-

nommen wird, daß sie ‚unparteiisch' sei, vollkommen unterschiedlich interpretiert werden kann. Übertragen auf die Modellierung von Workflows bedeutet diese Erkenntnis, daß bei der Interpretation von Geschäftsprozessen, in Abhängigkeit vom Betrachter (Benutzer vs. Software-Entwickler), sehr unterschiedliche Ergebnisse zustande kommen können. Als Folge hiervon können Workflows entstehen, die nur bedingt einsetzbar sind und deshalb überarbeitet werden müssen.

Untermauert wird diese Ansicht durch eine weitere Feststellung von Suchman. Sie weist darauf hin, daß bei der Darstellung von Arbeit eine Tendenz besteht, wonach Arbeit mit zunehmenden Abstand ‚verschwindet'. Anders ausgedrückt bedeutet dies, je weiter man von der Arbeit anderer entfernt ist, desto schlichter, oft sogar stereotypisch, wird die eigene Vorstellung dieser Arbeit.

Zur Modellierung gehört auch die Verwendung von Rollen. Durch organisatorische Zuordnungsstrategien soll bestimmten Situationen Rechnung getragen werden. Im obigen Beispiel ging es darum zu verhindern, daß Antragsteller und die Person, die den Antrag genehmigt, identisch sind. Es wird versucht, alle möglichen Situationen bei der Auswahl von Rollenträgern zu berücksichtigen und diese Situationen dann formal darzustellen. Die dabei entstehenden organisatorischen Zuordnungsstrategien können allerdings sehr kompliziert seien. Insbesondere wenn Ausnahmen mit erfaßt werden sollen. Will z.B. der Geschäftsführer Urlaub machen, so erfolgt dies in Absprache mit dem Stellvertreter und den Abteilungsleitern. Das Schema, den Urlaub einfach durch den Vorgesetzten genehmigen zu lassen, funktioniert in diesem Fall nicht, da es keinen gibt.

Ob in Anbetracht der angeführten Kritikpunkte überhaupt eine komplette Modellierung möglich ist, erscheint mir fraglich.

Die Verlagerung von Entscheidungsbefugnissen vom Benutzer zum WFMS kann sich ebenfalls negativ auswirken. Der Benutzer wird durch die automatische Werkzeugauswahl gezwungen sich an das System anzupassen, anstatt wie bisher selber entscheiden zu können, auf welche Art und Weise er seine Arbeit erledigt. Und wenn die obige Annahme bzgl. der geringen Eingriffmöglichkeiten seitens des Benutzers zutrifft, so kann das Gefühl beim Benutzer entstehen, zum bloßen Dateneingeber degradiert geworden zu sein. Aus meiner Sicht können sich diese beiden Punkte dann in Form sinkender Motivation niederschlagen. Desweiteren kann das Verantwortungsbewußtsein des Benutzers zurückgehen, da ja das WFMS auf den korrekten Ablauf achtet (Mentalität: *Das System wird Fehler schon melden*). Die totalen Überwachungsmöglichkeiten des WFMS können

leicht einen ‚Big brother is watching you'-Effekt beim Benutzer entstehen lassen ([Ver95a], S. 158).

Kapitel 4

Prozeßmuster und VBS

Die im vorigen Kapitel vorgestellen Workflows gelten als eine Ausprägung der *ablaufsteuernden* Sichtweise. Diesen werden jetzt Prozeßmuster gegenübergestellt, die von der *unterstützenden* Sichtweise geprägt worden sind.

4.1 Motivation

Workflows bieten nur sehr eingeschränkte Eingriffsmöglichkeiten seitens des Benutzers. Dies liegt in der ablaufgesteuerten Sichtweise begründet, die drauf abzielt, „menschliches Arbeitshandeln durch Maschinen zu ersetzen oder bis auf notwendige Dateneingaben (Input) zu reduzieren“([Gry95], S. 5). Im Gegensatz dazu steht die unterstützende Sichtweise, bei der Steuerung und Kontrolle beim Menschen verbleiben und Maschinen nur unterstützenden Charakter besitzen. Diese Sichtweise dient als Grundlage bei der Entwicklung von Prozeßmustern.

4.2 Ablaufsteuernde Sichtweise

Die ablaufsteuernde Sichtweise zielt darauf ab, Arbeitsabläufe soweit wie möglich auf Maschinen zu übertragen. Um dies zu erreichen, sollen Handlungsanweisungen für den Menschen so formuliert werden, daß die Anwendung der Handlungsanweisungen mechanisch erfolgen kann ([Gry95], S.46). Dabei werden Arbeitsprozesse in immer kleinere Einheiten zerlegt, die man dann weitestgehend zu automatisieren versucht. „Dem liegt der Gedanke zugrunde, daß durch die Zergliederung

der Arbeitsprozesse in der Anwendungswelt, diese ebenso beherrschbar und kontrollierbar werden, wie die Programme, die nach diesen Kriterien entworfen und implementiert worden sind“([Gry95], S.15). Der Mensch wird bei dieser Vorgehensweise immer mehr in seiner Selbständigkeit eingeschränkt bis er nur noch Daten eingeben darf oder in Teilbereichen ganz durch einen Automaten abgelöst wird. Es wird somit eine Austauschbarkeit von menschlichen und maschinellen Handeln angestrebt.

Der Grund für diese Zielsetzung liegt in der Betrachtungsweise des Menschen und seines Handelns. So wird in der ablaufsteuernden Sichtweise der Mensch als ‚potentieller Störfaktor‘ betrachtet, den man soweit wie möglich zu eleminieren hat ([Gry95], S.25). Sein Arbeitshandeln wird als ‚grundsätzlich fehlerhaft‘ angesehen und „muß deswegen mit Mitteln der Regelung und Steuerung beherrscht werden“ ([Gry95], Ergebnis 2.5).

Bei Systemen, die nach der ablaufsteuernden Sichtweise entwickelt worden sind, geht die Initiative immer vom System aus. Dies bedeutet, daß sich der Mensch dem System unterordnen muß, anstatt selbständig seine Arbeit gestalten zu können.

4.3 Unterstützende Sichtweise

Im Gegensatz zur ablaufsteuernden Sichtweise, die technikzentriert ist, steht bei der unterstützenden Sichtweise das menschliche Handeln im Vordergrund ([Gry95], S. 53). Dieses wird als überlegen gegenüber der maschinellen Instruktionsausführung bewertet. Der Benutzer wird als Experte des Anwendungsbereiches verstanden ([Wul95], S. 9) und verfügt über notwendiges Wissen und Erfahrungen, um auch auf neue Situation selbständig reagieren zu können. Maschinen sollen ihn bei seiner Arbeit unterstützen, wobei hervorzuheben ist, daß die Initiative immer vom Benutzer ausgeht.

Die Kooperation mit anderen Benutzern darf nicht durch computergestütze Mittel behindert werden. Stattdessen sollen diese Mittel nur unterstützend zum Einsatz kommen. Dies wird ebenfalls durch das Postulat der unterstützenden Sichtweise ausgedrückt, welches besagt, „daß zunächst die Merkmale des menschlichen Handelns zu betrachten sind und erst dann Möglichkeiten zu dessen technischer Unterstützung“([Gry95], S. 61).

Auch im Bereich der Koordinierung der Kooperation steht der Mensch im Vordergrund. Sie erfolgt durch direkte, zwischenmenschliche Kommunikation.

4.4 Prozeßmuster

Bevor auf Prozeßmuster eingegangen wird, sollen zuvor noch einmal kurz Workflows erörtert werden.

Mit Hilfe von Workflows versucht man eine möglichst genaue Abbildung von Geschäftprozessen zu erzeugen. Diese Abbildungen werden von einem WFMS verwendet, um Arbeitsabläufe zu steuern. Dabei ist eine hohe Standardisierung der Geschäftsprozesse unabdingbar, weil ansonsten als Folge der potentiellen Vielfalt von Geschäftsprozessen es zu einer unüberschaubaren Anzahl von Workflows kommen kann.

Die maschinelle Interpretation von Workflows erfordert eine möglichst vollständige Formalisierung von Hintergrundwissen („implicit knowledge“). Dies ist aber nur eingeschränkt durchführbar, „da jede Formalisierung, d.h. explizite Nennung eines Bestandteils des Hintergrundwissens, selbst wieder auf einer Menge von nicht expliziten Bestandteilen des Hintergrundwissen beruht“ ([Gry95], S. 46).

Ein anderer Ansatz wird mit Hilfe der unterstützenden Sichtweise beschritten. Dabei wird weiterhin das Potential des Menschen genutzt, anstatt es, wie bei der ablaufsteuernden Sichtweise, immer mehr zu reduzieren. Pläne haben hierbei eine ganz andere Bedeutung. Gryczan definiert *Plan* (unterstützende Sichtweise) in [Gry95], Begriff 3.4, wie folgt:

> Ein Plan ist ein Hilfsmittel, mit dem Erfahrungen vergegenständlicht werden. Er wird in einer Situation vom handelnden Subjekt als Hilfsmittel verwendet, um punktuelle Handlungsanleitung zu geben.

Hierbei ist hervorzuheben, daß Pläne nur als *Hilfsmittel* verwendet werden. Sie geben Orientierung in einer Situation, „ohne daß eine im formalen Sinn präskriptive Vorgabe für Handlungen vorliegt“ ([Gry95], S. 44). Somit unterscheiden sich Pläne der unterstützenden Sichtweise wesentlich von der ablaufsteuernden Sichtweise, welche den Plan in den Mittelpunkt stellt und dem Menschen quasi seine Kompetenzen aberkennt, indem ihm genau gesagt wird, was er zu tun hat.

Auch bezüglich Kooperation weist die unterstützende Sichtweise Vorteile gegenüber der ablaufsteuernden Sichtweise auf. So kommt es häufig vor, daß Menschen zusammenarbeiten, um ein Ziel zu erreichen, welches sie alleine nicht erreichen könnten. Dabei ist anzumerken, daß die Anzahl der beteiligten Menschen relativ klein ist und sie sich persönlich über ihren Arbeitszusammenhang bekannt sind ([Gry95], S. 163). Diese Zusammenarbeit (Kooperation) bedarf einer Koordinierung, da es ansonsten zu Konfliktsituationen, wie dem konkurrierenden Zugriff auf ein Material, kommen kann. Solch eine Konfliktsituation kann in der Regel durch Kommunikation aufgelöst werden. Dies macht aber deutlich, „daß es keine allgemeine technische Lösung für die Auflösung einer solchen Konkurrenzsituation gibt“([Gry95], S. 169).

Die Merkmale und Erfahrungen der kooperativen Arbeitsprozesse können durch *Prozeßmuster* vergegenständlicht werden. Das heißt, es werden die beteiligten Personen und die von ihnen ausgeführten Tätigkeiten beschrieben. Des weiteren gehören auch die Abhängigkeiten zwischen den einzelnen Tätigkeitsbeschreibungen dazu. Diese Abhängigkeiten werden als Tätigkeitennetz bezeichnet ([Gry95], S. 179). Eine formale Definition des Begriffs Prozeßmuster findet sich in [Gry95], Begriff 7.3:

> Ein Prozeßmuster ist ein gemeinsames Material zur Vergegenständlichung eines kooperativen Arbeitsprozesses. Durch das Prozeßmuster werden Verantwortlichkeiten von Personen oder Rollenträgern und Tätigkeiten in einem kooperativen Arbeitsprozeß festgelegt. Ein Prozeßmuster besteht aus der Angabe der Abhängigkeiten von und zwischen Tätigkeiten, die bei der kooperativen Arbeit zu erledigen sind und dazu notwendigen Dokumenten.

In diesem Zuammenhang muß noch erwähnt werden, daß Prozeßmuster den ‚Normalfall‘ repräsentieren , „der sich aufgrund von Erfahrungen herausgebildet hat“([Gry95], S. 178).

Durch *situierte Adaption* ist es möglich, Tätigkeitsbeschreibungen dynamisch aus dem Prozeßmuster zu entfernen oder gegebenenfalls hinzuzufügen. Ebenfalls möglich sind Veränderung an den Abhängigkeiten zwischen den einzelnen Tätigkeitsbeschreibungen.

Werden Prozeßmuster eingesetzt, so spricht man von *Situierter Koordination*. Gryczan definiert diesen Begriff in [Gry95], Begriff 7.5, wie folgt:

> Situierte Koordination ist die wechselseitige Abstimmung über die Reihenfolge und Zuständigkeit von Tätigkeiten bei kooperativer Arbeit. Der beschreibbare Teil eines gemeinsamen Erfahrungshintergrunds über eine kooperative Arbeitssituation wird durch Prozeßmuster vergegenständlicht. Das Prozeßmuster kann situativ adaptiert werden, um den Gegebenheiten der Koordinationssituation zu entsprechen.

Hier soll noch einmal hervorgehoben werden, daß Prozeßmuster nicht den Anspruch auf Vollständigkeit bzgl. der Modellierung aller möglichen Anwendungssituation erheben. Sie beschreiben stattdessen nur den ‚Normalfall' und können durch situierte Adaption an die jeweilige Situation angepaßt werden.

4.5 Vorgangsbearbeitungssysteme

Zum besseren Verständnis von Vorgangsbearbeitungssystemen (VBS) ist es sinnvoll, zuerst einmal den Begriff der *Arbeitsumgebung* einzuführen. Nach [Gry95], Begriff 6.1, wird dieser folgendermaßen beschrieben:

> Eine Arbeitsumgebung zur Unterstützung qualifizierter menschlicher Tätigkeiten ist der Ort für eine anwendungsfachlich motivierte Zusammenstellung von Werkzeugen, Automaten und Materialien. Durch die Arbeitsumgebung werden keine Reihenfolgebedingungen für die Verwendung von Werkzeugen und Arbeitsmitteln festgelegt.

Innerhalb einer Arbeitsumgebung befindliche Werkzeuge, Automaten und Materialien stehen dem Benutzer exklusiv zur Verfügung. Somit ist es nicht möglich aus verschiedenen Umgebungen gleichzeitig auf diese Gegenstände zuzugreifen ([Gry95], Ergebnis 6.1, S. 141). Für kooperative Arbeit ist es aber notwendig, daß alle beteiligten Benutzer Zugriff auf die jeweils benötigten Materialien erhalten. Durch Postkörbe werden explizite Verbindungen zwischen den Umgebungen geschaffen über die dann der Austausch von Materialien stattfinden kann. Dabei werden Postkörbe in Eingangs- und Ausgangskörbe unterschieden.

Für den Austausch werden sogenannte *Vorgangsmappen* verwendet. Hierbei handelt es um die „Zusammenfassung eines Transportbehälters, in dem die Materialien zur Bearbeitung einer kooperativen Aufgabe enthalten sind, und eines

Prozeßmusters“([Gry95], Begriff 8.1, S. 189). In diesem Zusammenhang sei auf die große Ähnlichkeit zwischen Vorgangsmappen und den in der Realität existierenden Gittermappen hingewiesen. Mit Hilfe eines Versandautomaten werden die Vorgangsmappen zwischen den verschiedenen Arbeitsumgebungen ausgetauscht. Der Versandautomat überwacht die Posteingangs- und ausgangskörbe und transportiert Material aus den Ausgangskörben in die per Empfängeradresse bestimmten Posteingangskörbe ([Gry95], S. 184).

Die als nächstes vorgestellten Werkzeuge wurden im Rahmen einer Diplomarbeit ([Wul95]) von Martina Wulf implementiert. Es handelt sich dabei aber um Labormuster bei denen die realisierte Funktionalität im Vordergrund stand. Sie alle sind Bestandteil der Arbeitsumgebung.

Das erste vorgestellte Werkzeug dient der Verwaltung von Vorgangsmappen. Hiermit ist man in der Lage Vorgangsmappen aus den Postkörben (sowohl Eingangs- als auch Ausgangskörbe) in die Arbeitsumgebung zu bewegen. Die umgekehrte Richtung ist ebenfalls möglich, wodurch der Versandautomat auf die Vorgangsmappen zugreifen kann. Des weiteren können neue Vorgangsmappen erzeugt und vorhandene betrachtet werden.

Möchte man eine neue Vorgangsmappe erstellen, so wird man im allgemeinen auf bereits bestehende Prozeßmuster zurückgreifen wollen. Diese werden in einem Prozeßmusterrepertoire gesammelt und können mit Hilfe eines Auflisters sondiert werden. Dieser Auflister ermöglicht es, Einträge des Prozeßmusterrepertoires zu löschen oder zu bearbeiten. Außerdem können neue Prozeßmuster erzeugt werden. Das eigentliche Bearbeiten oder Betrachten von Prozeßmustern erfolgt mittels des Prozeßmuster-Editors.

Neben einer Liste von Dokumenten, die zum Prozeßmuster gehören, wird vom Prozeßmuster-Editor angezeigt, wer das Prozeßmuster erzeugt hat. Ferner gehört noch eine Kurzbeschreibung des Prozeßmusters dazu. In den Prozeßmuster-Editor ist der Tätigkeitennetz-Editor eingebettet.

Bevor näher auf diesen Editor eingegangen wird, soll zunächst einmal das Tätigkeitsbeschreibung-Repertoire vorgestellt werden. Hierbei handelt es sich um einen Behälter, der alle Tätigkeitsbeschreibungen enthält. Der Zugriff auf dieses Repertoire erfolgt durch einen Auflister, der dieselbe Funktionalität wie der Prozeßmusterrepertoire-Auflister besitzt. Und auch bei der eigentlichen Bearbeitung bzw. Betrachtung von Tätigkeitsbeschreibungen wird, wie bei den Prozeß-

mustern, mit dem Tätigkeitsbeschreibung-Editor ein separates Werkzeug verwendet.

Im Tätigkeitsbeschreibung-Editor wird angeben, welche Dokumente im Zusammenhang mit der Durchführung einer Tätigkeit benötigt bzw. erzeugt werden. Zusätzlich gibt eine Kurzbeschreibung Aufschluß über die Tätigkeit und eventuelle Besonderheiten. Abschließend findet eine Zuordnung statt. Dabei wird festgelegt, von wem diese Tätigkeit auszuführen ist. Dies kann entweder eine bestimmte Person oder auch eine Rolle sein.

Es folgt nun die Beschreibung des bereits erwähnten Tätigkeitennetz-Editors. Hierbei handelt es sich um ein eingebettetes Werkzeug, mit dem sich das Tätigkeitennetz eines Prozeßmusters bearbeiten läßt. Tätigkeiten können eingefügt oder entfernt werden. Dies kann Veränderungen der Abhängigkeiten zwischen Tätigkeiten bedeuten, wenn z.B. eine Tätigkeit eine neue Nachfolgetätigkeit erhält. Bei diesen Veränderungen ist allerdings darauf zu achten, daß die Abhängigkeiten zyklenfrei bleiben, „weil anwendungsfachlich motivierte Tätigkeiten sich nicht wechselseitig bedingen können“([Gry95], S. 197). Das Material Prozeßmuster gewährleistet diese Zyklenfreiheit. Eine weitere Einschränkung besteht darin, daß ein Material nicht gleichzeitig in mehreren Umgebungen bearbeitet werden kann. Der Benutzer muß deshalb für eine Serialisierung sorgen, weil „das parallele Arbeiten an einer Vorgangsmappe fachlich nicht interpretierbar ist“([Gry95], S. 192).

Während der Ausführungszeit kann sich der einzelne Benutzer anhand des Prozeßmusters ein Bild über den gegenwärtigen Bearbeitungszustand machen. Bereits erledigte Tätigkeiten sind im Tätigkeitennetz abgehakt. Dieses ‚Abhaken‘ obliegt dem Benutzer nachdem er eine Tätigkeit durchgeführt hat und wird mit Hilfe des Prozeßmuster-Editors gemacht.

Ist es notwendig, ein Prozeßmuster zu adaptieren, so ist darauf zu achten, „daß die Abhängigkeiten zwischen den Tätigkeiten ‚sinnvoll‘ bleiben. Randbedingungen für die sinnvolle Änderung von Abhängigkeiten sind, daß

1. durch die Änderung die bisher erledigten Tätigkeiten erkennbar bleiben, d.h. nicht entfernt werden dürfen,

2. als erledigt gekennzeichnete Tätigkeiten nicht von unerledigten, d.h. [nicht] abgehakten Tätigkeiten abhängen dürfen, und

3. Abhängigkeiten zwischen als erledigt gekennzeichneten Tätigkeiten nicht verändert werden dürfen“([Gry95], S. 200).

Sinn und Zweck dieser Einschränkungen ist es, die Unverfälschheit der Historie zu garantieren. Trotzdem sind Änderungen, welche die weitere Bearbeitung einer kooperativen Aufgabe betreffen, weiterhin möglich. Die situative Adaptierbarkeit von Prozeßmustern bleibt somit bestehen ([Gry95], S. 201).

Als letztes Werkzeug sei das Info-Werkzeug genannt. Es gibt dem Benutzer einen Überblick über alle Vorgangsmappen, an denen er bereits gearbeitet hat oder für deren Bearbeitung er schon fest eingeplant ist. Selektiert man eine Vorgangsmappe aus der Liste, so erfährt man ihren augenblicklichen Aufhaltsort und wie lange sie sich dort schon ist. Aufgrund dieser Informationen ist es möglich die Arbeit einzelner Benutzer zu kontrollieren. Hier tut sich ein Zwiespalt zwischen dem Wunsch nach „Transparenz im Sinne der Möglichkeit, mehr Informationen über den Stand eines Vorgangs zu erhalten“ und der Möglichkeit erhöhter Kontrolle/Überwachung auf. „Zu welchen Zwecken und von wem das Info-Werkzeug verwendet wird, ist deshalb innerhalb der kooperativen Gruppe festzulegen“([Gry95], S. 203).

4.6 Rollen und Rollenauflösung

Eine so differenzierte Rollenmodellierung, wie sie im Zusammenhang mit Workflows im vorigen Kapital (siehe Kapitel 3.5) beschrieben wurde, ist bei Prozeßmustern nicht nötig. Der Grund hierfür liegt in den weitreichenden Eingriffsmöglichkeiten seitens des Benutzers. Kommt es zu einer Situation, in der das Ergebnis einer Rollenauflösung nicht den Erfordernissen entspricht, so kann der Benutzer korrigierend eingreifen.

Wie auch bei WFMS, wird mittels einer Rollenverwaltung eine Zuordnung von Benutzer zu Rolle vorgenommen. Dabei kann es vorkommen, daß ein Benutzer mehrere Rollen übernimmt. Als Ergebnis der Rollenauflösung wird der Name eines Rollenträgers zurückgegeben.

4.7 Schlußbetrachtung und Kritik

Mit Prozeßmustern wird ein anderer Weg beschritten, als dies bei Workflows üblich ist. Statt den Menschen als ‚potentiellen Störfaktor' und sein Arbeitshandeln als ‚grundsätzlich fehlerhaft' einzustufen, wie es im Rahmen der ablaufsteuernden Sichtweise geschieht, basieren Prozeßmuster auf der unterstützenden Sichtweise. Dies hat zur Folge, daß die Benutzer als Experten des Anwendungsbereiches verstanden werden, die über das notwendige Wissen und Erfahrung verfügen, um auf neue Situationen selbständig reagieren zu können.

Ein weiterer Unterschied zwischen Prozeßmustern und Workflows besteht bei der Modellierung von Anwendungssituationen. Workflow-Management-Systeme versuchen möglichst alle erdenklichen Anwendungssituationen zu modellieren. Bei Prozeßmustern wird „dem Umstand Rechnung getragen, daß die Vielfalt von Anwendungssituationen nicht in einem Rechnersystem durch einen vordefinierten Ablauf modelliert und gesteuert werden kann"([Gry95], S. 178). Vielmehr wird ein Prozeßmuster so gestaltet, „daß ‚an ihm' der Normalfall des kooperativen Arbeitsprozesses vergegenständlicht werden kann"([Gry95], S. 178). Durch entsprechende Eingriffsmöglichkeiten seitens der Benutzer können Prozeßmuster den jeweiligen Bedürfnissen angepaßt werden und bieten somit eine größere Flexibität als Workflows.

Prozeßmuster stellen also einen Ansatz dar, mit dem es möglich ist, die große Vielfalt von Anwendungssituation zu bewältigen, ohne dabei eine unüberschaubaren Anzahl von Arbeitsprozeßbeschreibungen erstellen zu müssen.

Abschließend soll noch einmal hervorgehoben werden, daß auch im Zusammenhang mit Prozeßmustern die Möglichkeit beziehungsweise die Gefahr erhöhter Kontrolle besteht.

Kapitel 5

Der Rollenwerkzeugkasten

Der Rollenwerkzeugkasten besteht aus mehreren Bestandteilen, die sich in unterschiedlicher Weise mit Rollen beschäftigen. Das Kernstück stellt der *Rollenauflöser* dar. Dies ist ein Automat, der eine Rollen entgegennimmt, sie auflöst und als Resultat den Namen einer Person zurückgibt. Mit Hilfe des *Automateneinstellers* kann ein anderer Auflösungsalgorithmus am Rollenauflöser eingestellt oder der Automat beendet werden. Der *Rollenverwalter* dient, wie der Name schon erahnen läßt, der Verwaltung von Rollen. Zu Beginn seiner Arbeitstätigkeit wählt der Benutzer am *Rollenanmelder* die Rollen aus, für die er während seiner Arbeitszeit zur Verfügung stehen will. Weiterhin wurde auch noch ein *Rollentester* entwickelt, mit dem sich die Funktionstüchtigkeit des Rollenauflösers überprüfen läßt. In den folgenden Abschnitten wird genauer auf die einzelnen Werkzeuge, ihre Funktionalität und ihren Aufbau eingegangen.

Vorweg soll noch das Rollenverständnis vorgestellt werden, welches der Entwicklung des Rollenwerkzeugkastens zugrunde lag:

Als Einsatzrahmen wurde für diese Arbeit ein Vorgangsbearbeitungssystem (siehe Kapitel 4.5 und [Wul95]) angenommen. Dabei wurde das in diesem Zusammenhang eingeführte Rollenverständnis übernommen. Dies hatte zur Folge, daß u.a. die in Kapitel 3.5 vorgestellten *organisationsbezogenen Rollen* bzw. *organisatorische Zuordnungsstrategien* nicht berücksichtigt wurden. Statt dessen wurde die relativ schlichte Zuordnung von Benutzern zu Rollen verwendet. Diese Zuordnung läßt sich auch damit begründen, daß das Einsatzfeld kleine Gruppen darstellen ([Gry95], S. 185).

Die Entwicklung des Rollenwerkzeugkastens fand unter SunOS/Solaris und

Linux statt. Dabei wurde Sniff+2.1 von TakeFive als Entwicklungsumgebung eingesetzt. Die verwendete Klassenbibliothek SEBib ist am Arbeitsbereich Softwaretechnik entwickelt worden und benutzt Tcl/Tk für die graphische Benutzungsschnittstelle (GUI).

5.1 Der Rollenauflöserautomat

Beim Rollenauflöserautomat handelt es sich um die wichtigste Komponente aus dem Rollenwerkzeugkasten. Seine Aufgabe ist das Auflösen von Rollen. Dazu nimmt er den Namen einer Rolle entgegen und gibt, in Abhängigkeit vom eingestellten Algorithmus, den Namen eines konkreten Rollenträgers zurück. Der Automat wurde nach dem WAM-Verständnis entwickelt (siehe dazu auch [Gry95], S. 119) und verfügt über keine eigene Benutzungsschnittstelle. Statt dessen erfolgt die Steuerung des Automaten mit Hilfe eines Einstellwerkzeugs (siehe Kapitel 5.2).

Zu Beginn seiner Arbeitstätigkeit erstellt der Rollenauflöserautomat zwei Listen (`pRollenliste` und `pAnmeldungsliste`), deren jeweilige Elemente vom Typ `tRolle`[1] sind. Zur Erstellung greift er auf Informationen der Rollendatenbank zurück. Bei der `pAnmeldungsliste` werden allerdings sämtliche Informationen bzgl. der Rollenträger gelöscht. Diese Liste dient dazu, Daten zu erfassen, die beim Anmeldevorgang von Benutzern entstehen. Dies bedeutet, daß hier registriert wird, für welche Rollen ein Benutzer sich angemeldet. Meldet sich ein Benutzer vom System ab, so werden seine entsprechenden Daten aus der Liste entfernt. Die `pRollenliste` enthält also ‚statische' Informationen und die `pAnmeldungsliste` enthält die ‚dynamische' Informationen beinhaltet.

Soll nun eine Rolle aufgelöst werden, so durchsucht der Automat zuerst einmal die Liste mit den Daten der angemeldeten Benutzer (`pAnmeldungsliste`) und versucht dort potentielle Rollenträger zu finden. Enthält diese Liste keine passenden Einträge, greift er auf die Daten der `pRollenliste` zurück. Durch dieses zweistufige Verfahren soll gewährleistet werden, daß vorrangig solche Rollenträger ausgewählt werden, die angemeldet sind und somit aktiv zur Verfügung

[1] Ein Objekt der Klasse `tRolle` besitzt folgende Attribute: Zwei Strings mit dem Namen der Rolle und ihrer Beschreibung (`sRollenname`, `sRollenbeschreibung`), sowie eine String-Liste mit den Namen der Rollenträger (*`pRollentraegerliste`).

stehen.

Gibt es mehr als nur einen potentiellen Rollenträger, so wird die Auswahl eines konkreten Rollenträgers in Abhängigkeit vom eingestellten Algorithmus ausgeführt. Die im Rahmen dieser Studienarbeit implementieren Algorithmen sind sehr einfach und sollen nur die generelle Machbarkeit demonstrieren.

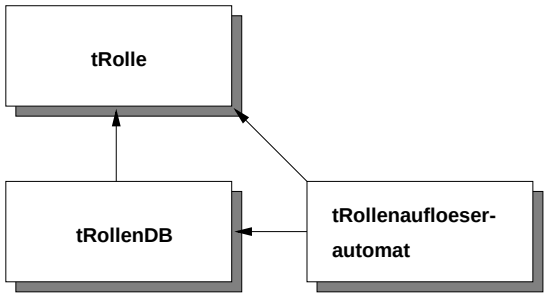


Abbildung 5.1: Klassen des Rollenauflösers

Abbildung 5.1 zeigt den Aufbau des Rollenauflöserautomaten. In diesem Zusammenhang soll noch auf eine weitere Besonderheit hingewiesen werden. Da es möglich sein soll, mit dem Rollenauflöserautomaten über die Grenzen eines Prozeßraums hinaus Verbindung aufzunehmen, wurde ein *Rollenauflöser-Proxy* entwickelt, welcher mittels Sockets mit dem Rollenauflöserautomaten kommuniziert (genauere Angaben finden sich in Kapitel 5.2).

5.2 Der Automateneinsteller

Dieses Werkzeug (siehe Abb. 5.2) dient der Steuerung des Rollenauflöserautomaten. Mit seiner Hilfe kann festgestellt werden, welcher Auflösungsalgorithmus zur Zeit aktiv ist. Wird ein anderer Algorithmus aktiviert, so wird diese Änderung sofort vom Automaten übernommen. Zusätzlich besteht die Möglichkeit den Rollenauflöserautomaten zu beenden.

Im vorherigen Kapitel wurde bereits erwähnt, daß für die Kommunikation mit dem Rollenauflöserautomaten sogenannte *Rollenauflöser-Proxy* zur Verfügung stehen. Für die Entwicklung des `tRollenauflöserProxy` wurde auf das Entwurfsmuster *Proxy* zurückgegriffen. An dieser Stelle soll nur kurz auf das Konzept von

Abbildung 5.2: Einstellwerkzeug für den Rollenauflöserautomaten

Proxies eingegangen werden. Eine genaue Beschreibung findet sich in [GHJV95], S. 207ff.

Mit einem *(remote) Proxy* kann man die Tatsache verbergen, daß sich ein Objekt in einem anderen Adreßraum befindet. Zu diesem Zweck wird ein *Proxy-Objekt* (Stellvertreter-Objekt) erzeugt, welches über dieselbe Schnittstelle verfügt, wie das zu vertretende Objekt. Des weiteren verhält sich das *Proxy-Objekt* genauso wie das Objekt im anderen Adreßraum.

Intern nimmt hier das *Proxy-Objekt* einen Methoden-Aufruf entgegen, kodiert ihn und übermittelt anschließend eine Zeichenkette via Sockets an das eigentliche Objekt. Dieses dekodiert die empfangenen Daten, führt den enthaltenen Methoden-Aufruf aus. Das Ergebnis dieses Aufrufs wird erneut kodiert und an das *Proxy-Objekt* zurückgesendet, welches die empfangenen Daten wieder dekodiert. Als letzter Schritt wird das Resultat des Methoden-Aufrufs an das aufrufende Objekt übergeben.

Wie der `tRollenauflöserProxy` in das Werkzeug *Automateneinsteller* integriert wird, ist in der Abbildung 5.3 zu sehen. Durchgezogene Linien stehen dabei für Benutzt-Beziehungen, während gestrichelte Linien Beobachter-Beziehungen darstellen.

5.3 Der Rollenverwalter

Die Verwaltung der Rollendaten erfolgt mit Hilfe des *Rollenverwalters*. Genauer ausgedrückt, ist hiermit das Erstellen neuer Rollen bzw. das Betrachten, Verändern oder Löschen von vorhandenen Rollen gemeint.

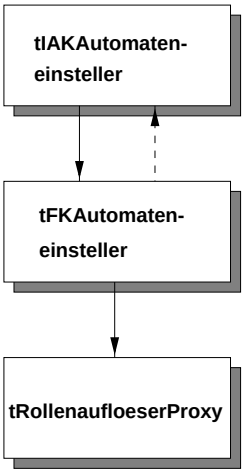


Abbildung 5.3: Klassen des Automateneinstellers

Auf der linken Seite der Abbildung 5.4 befindet sich eine Liste mit den Namen aller Rollen. Wird eine Rolle aus dieser Liste selektiert, so werden ihre Daten in den entsprechenden Feldern angezeigt. Hierbei handelt es sich um die Rollenbeschreibung, den Rollennamen und eine Liste mit Rollenträgern. Die Mitarbeiterliste enthält die verbleibenden Mitarbeiter, die nicht als Rollenträger eingetragen wurden. Rollenträger- und Mitarbeiterliste sind komplementäre Listen, d.h. wird ein Mitarbeiter als Rollenträger eingetragen, so wird er aus der Liste der verbleibenden Mitarbeiter entfernt. Gleiches gilt auch für den umgekehrten Fall.

Das Feld *Modifiziert* gibt Auskunft darüber, ob eine Rolle verändert wurde. Ist dies der Fall, so ist auch der Knopf *Speichern* aktiv. Soll statt einer veränderten Rolle eine andere selektiert werden, so fragt das Werkzeug den Benutzer vorher, ob die Änderungen gespeichert werden sollen. Wird versucht eine neue Rolle abzuspeichern, ohne einen entsprechenden Rollennamen eingegeben zu haben, erscheint eine Fehlermeldung, die den Benutzer auf diesen Umstand hinweist.

Der technische Aufbau des *Rollenverwalters* (siehe Abb. 5.5) ist relativ einfach. Neben der obligatorischen Trennung in Funktions- und Interaktionskomponente werden noch die Klassen `tRolle` und `tRollenDB` benutzt. Letztere dient der persistenten Speicherung der Rollendaten und verwendet als Datenbank `gdbm`.

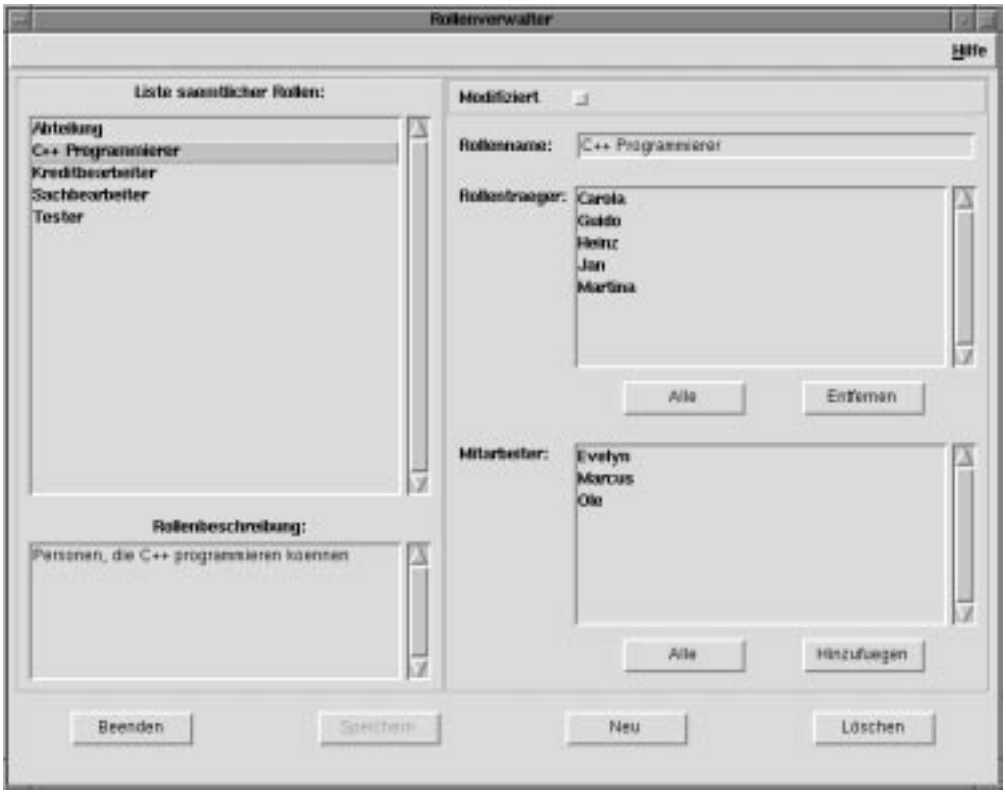


Abbildung 5.4: Rollenverwalter

5.4 Der Rollenanmelder

Bevor ein Benutzer aktiv als Rollenträger zur Verfügung stehen kann, muß er sich zuerst anmelden. Dies geschieht mittels des *Rollenanmelders*. Hierbei hat der Benutzer die Möglichkeit, individuell zu entscheiden, welche seiner Rollen er spielen will. Mit ‚seine Rollen' sind die Rollen gemeint, für die er als Rollenträger eingetragen ist.

Nach erfolgreicher Anmeldung, d.h. es wurde ein gültiger Benutzername eingegeben, werden die Rollen des Benutzers aufgelistet. Anschließend kann selektiv entschieden werden, für welche Rollen der Benutzer aktiv zur Verfügung stehen will. Um ein häufig benutztes Muster selektierter Rollen nicht jedes Mal von neuem einzustellen, besteht die Möglichkeit solch ein Muster als *Profil* abzuspeichern. Beim *Rollenanmelder*-Werkzeug gibt es drei Knöpfe, die mit solch einem Profil belegt werden können (siehe Abb. 5.6). Wird solch ein Profil-Knopf gedrückt, lädt das Werkzeug das entsprechende Profil aus der Profiledatenbank und stellt

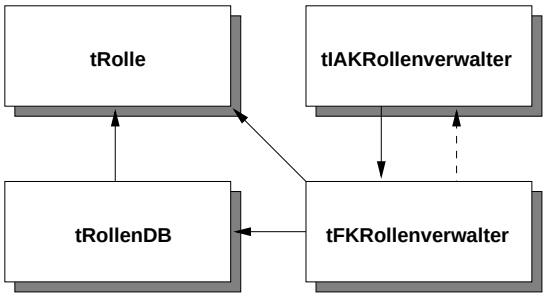


Abbildung 5.5: Klassen des Rollenverwalters

Abbildung 5.6: Rollenanmelder

die abgespeicherte Rollenselektion wieder her.

Anhand der Abbildung 5.7 kann man den Aufbau des Rollenanmelder-Werkzeuges nachvollziehen. Dabei fällt auf, daß auch bei diesem Werkzeug der `tRollenauflöserProxy` verwendet wird. Stellvertretend für den *Rollenauflöser* nimmt er die Anmeldedaten entgegen. Die Informationen bzgl. der Rollen für die ein Benutzer als Rollenträger eingetragen ist, werden aus der Rollendatenbank geholt. Wie auch bei der Rollendatenbank wird für die persistente Speicherung der Profildaten eine Datenbank eingesetzt, welche `gdbm` verwendet.

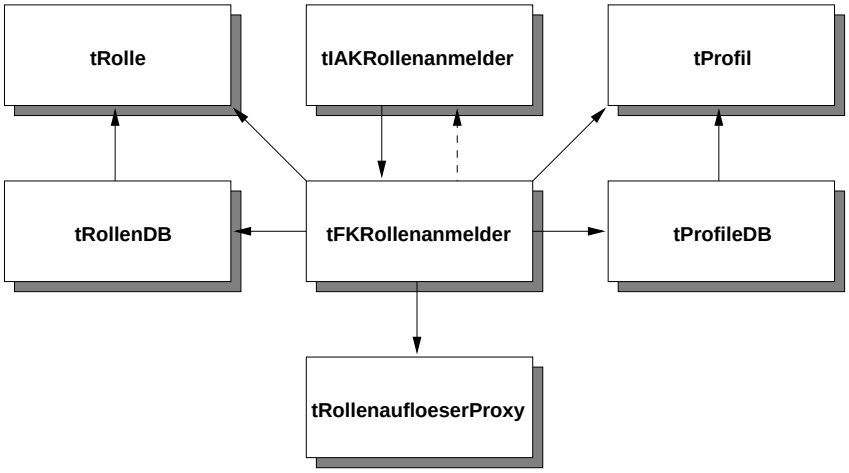


Abbildung 5.7: Klassen des Rollenanmelders

5.5 Der Rollentester

Als letztes Werkzeug soll der *Rollentester* vorgestellt werden. Mit seiner Hilfe kann die Funktionstüchtigkeit des *Rollenauflöserautomaten* überprüft werden. Dabei wird dem *Rollenauflöserautomaten* eine Rolle übergeben und das Ergebnis der Rollenauflösung angezeigt. In Abhängigkeit vom eingestellten Algorithmus und den jeweils angemeldeten Benutzern kann dieses Ergebnis für ein und dieselbe Rolle unterschiedlich ausfallen.

Nach dem Starten des *Rollentesters* (siehe Abb. 5.8) werden sämtliche Rollen, die sich in der Rollendatenbank befinden, in einer Liste angezeigt. Wird ein Eintrag dieser Rollenliste selektiert, so kann entweder durch einen Doppelklick auf diesen Eintrag oder das Drücken des Knopfes *Auflösen* der Auflösevorgang gestartet werden. Das Ergebnis der Rollenauflösung wird in einem separaten Feld angezeigt.

Das Werkzeug *Rollentester* zeigt, wie einfach es ist, unter Verwendung des `tRollenauflöserProxy` und der `tRollenDB`, Rollenauflösung in ein Werkzeug zu integrieren. Der Aufbau ist in Abbildung 5.9 dargestellt.

Abbildung 5.8: Rollentester

5.6 Schlußbetrachtung und Kritik

Mit dem Rollenwerkzeugkasten wurde ein Satz von Werkzeugen erstellt, der es ermöglicht rollenspezifische Aufgaben zu erledigen und dessen Einbindung in ein anderes System relativ leicht möglich ist. Auf der technischen Seite ist insbesondere die Verwendung des Entwurfsmusters *Proxy* hervorzuheben, das hier die Kommunikation über die Grenzen von Adreßräumen hinaus sehr vereinfacht hat. Die Entwicklung und Implementation fand unter dem Gesichtspunkt der generellen Machbarkeit statt. Somit lassen sich vielfältige Ansatzpunkte für Verbesserungen und Kritik finden. Es sollen hier nur drei Punkte erwähnt werden, die bei einer eventuellen Weiterentwicklung verbessert werden sollten.

1. Der *Rollenauflöserautomat* wertet zu Beginn seiner Tätigkeit die Rollendatenbank aus und übernimmt die Rollendaten. Kommt es während seiner Laufzeit zu Veränderung der Rollendatenbank, so werden diese erst bei einem erneuten Start berücksichtigt.

2. Die verwendete Datenbank `gdbm` verwaltet alle Daten in einer einzigen Datei. Je nach Volumen sollte statt dessen eine richtige Datenbank eingesetzt werden. Auch wäre das Entkoppeln der Datenbankzugriffe in einen eigenständigen Prozeß sinnvoll (Einsatz von Proxies).

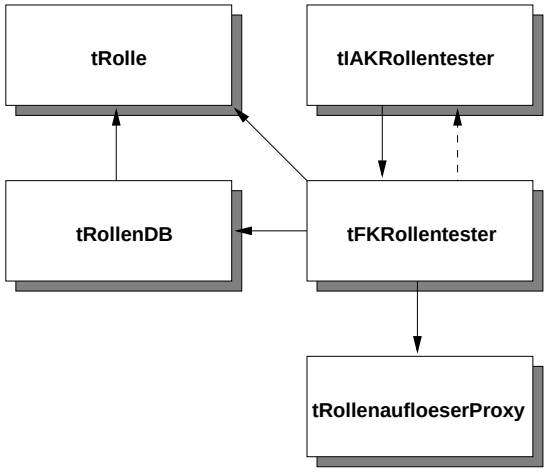


Abbildung 5.9: Klassen des Rollentesters

3. Als Ergebnis der Rollenauflösung wird nur der Name eines Rollenträgers zurückgegeben. Für einen realen Einsatz sollten allerdings sinnvollere (eindeutige) Rückgabewerte verwendet werden, z.B. User-IDs. Mit Hilfe von Systemdiensten können aus den User-IDs verschiedene Informationen gewonnen werden, wie z.B. Namen oder E-Mail Adressen.

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen dieses Kapitels werden die wesentlichen Punkte dieser Arbeit reflektiert und kritisch betrachtet. Mit einem Ausblick auf mögliche Weiterentwicklungen schließt diese Arbeit.

Zentrales Thema dieser Arbeit war die Entwicklung eines aus (Software-) Werkzeugen, Automaten und Materialien bestehenden Rollenwerkzeugkastens, der es mit seinen Bestandteilen ermöglicht, verschiedene rollenspezifische Aufgaben zu erledigen. Gemeint sind hiermit das Auflösen von Rollen, ihre Verwaltung und das Anmelden für selektierte Rollen. Zusätzlich wurde ein Werkzeug entwickelt, mit dem die Rollenauflösung getestet werden konnte.

Zu Beginn dieser Arbeit wurde die Softwareentwicklungsmethodik WAM eingeführt. Sie bildete den Rahmen für die Entwicklung des Rollenwerkzeugkastens, welcher unter Verwendung der hier angeführten Metaphern und Dokumente erstellt wurde.

Im Anschluß daran wurden zwei mögliche Anwendungsfelder des Rollenwerkzeugkastens betrachtet. Dabei zeigte sich, daß Workflows vorrangig in solchen Anwendungsbereichen eingesetzt werden sollten, deren Arbeitsabläufe weitesgehend fest vorgegeben sind, da Workflows nur geringe Eingriffsmöglichkeiten seitens der Benutzer gestatten und somit relativ unflexibel sind. Prozeßmuster wiederum stellen hohe Anforderung an die Qualifikationen des Anwenders, da dieser als ‚Experte' des Anwendungsbereiches betrachtet wird. Nichtsdestotrotz scheinen die Möglichkeiten von Prozeßmustern erfolgsversprechender, da mit verhältnismäßig geringem Aufwand ein großer Anteil möglicher Arbeitssituation abgedeckt werden kann. Zudem kann angenommen werden, daß die im Vergleich zu Workflows viel

größere Verantwortung des Anwenders motivierender wirkt, als die fortschreitende Reduzierung von Entscheidungsbefugnissen, wie sie bei Workflows ausgeübt wird.

Und schließlich wurden die verschiedenen Bestandteile des Rollenwerkzeugkastens präsentiert. Hierbei wurde sowohl auf den technischen Aufbau (Klassendiagramme), als auch auf die Benutzungsschnittstellen eingegangen. Wo es sinnvoll oder notwendig erschien, wurden technische Details hervorgehoben.

Im Rahmen einer Weiterentwicklung des Rollenwerkzeugkastens sollten zuerst die in Kapitel 5.6 erwähten Verbesserungen implementiert werden. Des weiteren sollte geklärt werden, inwieweit die Bevorzugung von angemeldeten Rollenträgern bei der Rollenauflösung sinnvoll ist. Für den Fall, daß es genau einen angemeldeten Rollenträger für eine Rolle gibt, wird dieser in der jetzigen Version immer ausgewählt. Dadurch könnte sich das Anmeldeverhalten dahin gehend verschieben, daß ein Rollenträger sich nur noch für solche Rollen anmeldet, für die auch andere Rollenträger angemeldet sind.

Abschließend sei noch auf [S+97] verwiesen. In diesem Bericht, der die Ergebnisse des *NSF Workshop on Workflow and Process Automation in Information Systems* zusammenfaßt, werden aktuelle Ansätze und Entwicklungen in bezug auf Workflows vorgestellt. Bemerkenswert ist hierbei, daß es Tendenzen in Richtung der unterstützenden Sichtweise gibt. Dies wird am Beispiel der Einflußmöglichkeiten des Menschen hinsichtlich der Definition, Ausführung und Entwicklung von Prozessen deutlich (siehe [S+97], S. 30). Dort wird erstmals neben der bisherigen (ablaufsteuernden) Sichtweise auch ein Ansatz beschrieben, welcher die Merkmale der unterstützenden Sichtweise aufweist. Des weiteren wird darauf hingewiesen, daß beide Sichtweisen im Zusammenhang mit echten Prozessen (*real-world processes*) von Bedeutung sind. Dies läßt eine Abkehr von der einseitigen (ablaufsteuernden) Sichtweise erkennen, wie sie bisher vertreten wurde.

Literaturverzeichnis

[BJ94a] Christoph Bußler und Stefan Jablonski. An Approach to Integrate Workflow Modeling and Organization Modeling in an Enterprise. In *Proceedings of the Third IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE), Morgantown, West Virgina, USA*, April 1994.

[BJ94b] Christoph Bußler und Stefan Jablonski. Implementing Agent Coordination for Workflow Management Systems Using Active Database Systems. In *Proceedings of the 4th International Workshop on Research Issues in Data Engineering: Active Database Systems (RIDE–ADS '94), Houston, Texas, USA*, February 1994.

[Bus95] Christoph J. Bussler. Policy Resolution in Workflow Management Systems. *Digital Technical Journal*, 6(4), 1995.

[Flo91] Christiane Floyd. Arbeitsunterlagen zur Lehrveranstaltung „Einführung in die Softwaretechnik“, 1991.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides. *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Massachusetts, 1995.

[Gry95] Guido Gryczan. *Situierte Koordination computergestützter qualifizierter Tätigkeit über Prozeßmuster*. Dissertation, Universität Hamburg, Fachbereich Informatik, 1995.

[Jab95] Stefan Jablonski. Workflow-Management-Systeme: Motivation, Modellierung, Architektur. *Informatik Spektrum*, 1995.

[KGZ94] Klaus Kilberth, Guido Gryczan und Heinz Züllighoven. *Objektorientierte Anwendungsentwicklung*. Vieweg Verlag, 1994.

[MB91] J.C. McCarthy und W.M. Bluestein. Workflow's Progress. In *The Computing Strategy Report*. Forrester Research Inc., Cambridge, 1991.

[PP91] Wolfgang Prinz und Paola Pennelli. Revelance of the X.500 Directory to CSCW Applications - Directory support for computer based group communication. In J.M. Bowers und S.D. Benford, Hrsg., *Studies in Computer Supported Cooperative Work - Theory, Practise and Design*, Seiten 267–283. Elsevier Science Publisher B.V. (North-Holland), 1991.

[Pri93] Wolfgang Prinz. TOSCA Providing organisational information to CSCW applications. In Giorgio de Michelis, Carla Simone und Kjeld Schmidt, Hrsg., *Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW '93*, Seiten 139–154, 1993.

[Rie95] Dirk Riehle. Muster am Beispiel der Werkzeug und Material Metapher. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Softwaretechnik, März 1995.

[S+97] Amit Sheth et al. Report from the NSF Workshop on Workflow and Process Automation in Information Systems. *ACM SIGSOFT Software Engineering Notes*, 22(1):28–38, January 1997.

[SHL91] H.T. Smith, P.A. Hennessy und G.A. Lunt. An object-oriented framework for modelling organisational communication. In J.M. Bowers und S.D. Benford, Hrsg., *Studies in Computer Supported Cooperative Work - Theory, Practise and Design*, Seiten 145–157. Elsevier Science Publisher B.V. (North-Holland), 1991.

[Suc95] Lucy Suchman. Making Work Visible. *Communications of the ACM*, 38(9):56–64, September 1995.

[Ver95a] Gerhard Versteegen. Alles im Fluß. *iX*, Seiten 152–160, März 1995.

[Ver95b] Gerhard Versteegen. Flußkontrolle. *iX*, Seiten 140–146, September 1995.

[Wul95] Martina Wulf. Konzeption und Realisierung einer Umgebung zur Koordination rechnergestützter Tätigkeiten in kooperativen Arbeitsprozessen. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 1995.

[WW94] Martina Wulf und Dirk Weske. Konzepte zur Materialversorgung verteilter Werkzeugumgebungen am Beispiel der Anbindung einer objektorientierten Datenbank. Studienarbeit, Universität Hamburg, Fachbereich Informatik, AB Softwaretechnik, 1994.