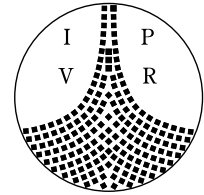


Universität Stuttgart  
**Institut für Parallele und Verteilte  
Höchstleistungsrechner**  
Verbundprojekt PoliFlow



Dipl.-Inform. Thomas Kindler  
Dipl.-Inform. Ottokar Kulendik  
Dipl.-Inform. Kerstin Schneider  
Dipl.-Inform. Reiner Siebert  
Dipl.-Inform. Tobias Soyez  
Prof. Dr. rer. nat. Kurt Rothermel

Institut für Parallele und Verteilte Höchstleistungsrechner  
Universität Stuttgart  
Breitwiesenstraße 20-22, D-70565 Stuttgart

Projektpartner: Hewlett Packard GmbH  
OFD Berlin  
Universität Stuttgart, IAT

**Verbundprojekt PoliFlow**

## **Abschlußbericht**

**Universität Stuttgart, IPVR**

**Förderkennzeichen: 01 IT 401 C/4**

Zuwendungsempfänger: Universität Stuttgart, IPVR  
Förderkennzeichen: 01 IT 401C/4  
Vorhabenbezeichnung: PoliFlow  
Laufzeit des Vorhabens: 01.05.1994 bis 31.12.1998



# Inhaltsverzeichnis

<b>1 Einleitung</b> .....	8
1.1 Problemstellung .....	9
1.2 Zielsetzung und Ergebnisse .....	10
1.2.1 Arbeitsbereiche und Ergebnisse am IPVR .....	12
1.3 Struktur des Berichts .....	14
<b>2 Projektverlauf</b> .....	15
2.1 Projektmanagement .....	15
2.1.1 POLIKOM .....	15
2.1.2 POLIFLOW .....	15
2.1.3 PoliFlow am IPVR .....	16
2.2 Öffentlichkeitsarbeit .....	17
2.2.1 Präsentationen .....	18
2.2.2 Veröffentlichungen .....	20
2.2.3 Das PoliFlow-Infonet am IPVR .....	21
2.3 Organisation und Kennzahlen .....	23
2.3.1 Studentische Arbeiten .....	23
2.3.2 Mitarbeiter/innen .....	24
2.3.3 Externe Kontakte .....	25
<b>3 Realisierung eines WFMS (WR)</b> .....	26
3.1 Grundlagen .....	27
3.1.1 WWW- / Internet-Technologien .....	27
3.2 Problemstellung und Anforderungen .....	31
3.2.1 Problemstellung .....	31
3.2.2 Anforderungen und Entwurfsziele .....	31
3.3 Konzepte und Lösungsansätze .....	32
3.3.1 Stuttgarter Workflow And Telecooperation System (SWATS) .....	32
3.3.2 Ausgewählte eigenimplementierte Komponenten der Architektur .....	38
3.3.3 Prototypische Anwendungen in der Architektur .....	40
3.4 Zusammenfassung, Bewertung und Ausblick .....	43
3.5 Literatur .....	44
<b>4 Flexibilität und Anpassungsfähigkeit in WFMS (AWS)</b> .....	45
4.1 Motivation und Einleitung .....	45
4.2 Problemstellung und Anforderungen .....	47
4.2.1 Problemstellung .....	47
4.2.2 Anforderungen .....	49
4.3 Lösungsansatz .....	50
4.4 Architektur .....	53
4.4.1 Technologien .....	53

4.4.2 Referenzarchitektur für traditionelle WFMS .....	54
4.4.3 Erweiterte Architektur für adaptive WFMS .....	55
4.5 Benutzerschnittstellen .....	58
4.6 Zusammenfassung und Bewertung .....	63
4.6.1 Die Ergebnisse im Überblick .....	63
4.6.2 Weitere Ansätze zur Flexibilisierung von WFMS .....	64
4.6.3 Bewertung und Ausblick .....	65
4.7 Literatur .....	66
<b>5 Sicherheit in WFMS (SWS) .....</b>	<b>69</b>
5.1 IT-Sicherheit und Workflow-Management .....	69
5.1.1 Motivation zur IT-Sicherheit .....	70
5.1.2 Grundlagen der IT-Sicherheit .....	70
5.1.3 Sicherheitsstandards der IT .....	72
5.1.4 Sicherheit in interorganisationellen Workflows .....	74
5.2 Lösungsansatz in PoliFlow .....	76
5.2.1 Sicherheit in Workflow-Management-Systemen .....	77
5.2.2 Grundansatz zur Umsetzung der Sicherheit .....	79
5.2.3 Die SWATS Basis-Architektur .....	82
5.2.4 Die SWATS Sicherheitsarchitektur .....	83
5.2.5 Realisierung der Sicherheitsarchitektur .....	85
5.3 Zusammenfassung .....	89
5.4 Literatur .....	89
<b>6 Gemischte Arbeitsformen (MWM) .....</b>	<b>91</b>
6.1 Motivation .....	91
6.2 CSCW, Groupware & Workflow .....	92
6.2.1 Groupware .....	92
6.2.2 Workflow-Systeme .....	93
6.2.3 Abgrenzung zwischen Groupware und Workflow-Systemen .....	94
6.2.4 Werkzeuge zur Kooperationsunterstützung .....	95
6.3 Problembereiche und Anforderungen .....	96
6.4 Konzepte und Lösungsansätze zur Unterstützung von Kooperation .....	99
6.4.1 Referenzarchitektur .....	99
6.4.2 Integration einer Rückfrage-Funktionalität .....	101
6.5 WWW-basierte Vorgangsbearbeitung .....	108
6.6 Literatur .....	110
<b>7 Konsistenzsicherung und Zuverlässigkeit in WFMS (CAM) .....</b>	<b>112</b>
7.1 Prinzipien der Konsistenzsicherung .....	113
7.2 Eigenschaften der Anwendung .....	115
7.3 Erweiterte Transaktionsmodelle .....	116
7.4 Definition eines Basismodells .....	116

7.4.1 Das ConTract-Modell .....	118
7.4.2 Erweiterungen des ConTract-Modells .....	120
7.5 Aspekte der Kompensation .....	121
7.6 Diskussion und Bewertung .....	123
7.7 Literatur .....	124
<b>8 Technologie Evaluierung (TE).....</b>	<b>127</b>
8.1 Evaluierung von verteilten Systemdiensten.....	127
8.1.1 Anforderungen .....	127
8.1.2 Technologie Evaluierung .....	129
8.1.3 Bewertung.....	132
8.2 Multimedia-Technologien .....	133
8.2.1 Klassifikation .....	133
8.2.2 Asynchrone Multimedia Technologien .....	134
8.2.3 Synchrone Multimedia Technologien.....	147
8.3 Zusammenfassung .....	158
<b>9 Zusammenfassung und Bewertung.....</b>	<b>160</b>

# Abbildungsverzeichnis

Abbildung 2.1: Das PoliFlow-Infonet am IPVR.....	22
Abbildung 3.1: Architektur von SWATS.....	34
Abbildung 3.2: Benutzungsschnittstelle des Prototyps.....	35
Abbildung 3.3: HTML-Formular zur Bearbeitung der Aktivität Wohnung anbieten.....	41
Abbildung 3.4: Applet zur Bearbeitung der Aktivität Bewerber auswählen.....	41
Abbildung 4.1: Integrierter Lösungsansatz: Anpassungsfähige Workflow-Systeme.....	51
Abbildung 4.2: Basisarchitektur und Schichten.....	54
Abbildung 4.3: SWATS-Architektur.....	56
Abbildung 4.4: Kontrollfluß-Sicht.....	60
Abbildung 4.5: Zeitliche Sicht.....	61
Abbildung 4.6: Konstruktsicht.....	62
Abbildung 5.1: Nationale und internationale Sicherheitskriterien.....	73
Abbildung 5.2: Komponentenarchitektur von SWATS.....	83
Abbildung 5.3: Sicherheitsarchitektur von SWATS.....	84
Abbildung 5.4: Aktivitätsbasierte Zugriffskontrolle.....	87
Abbildung 5.5: Sicherheitsmodell für interorganisationelle Workflows.....	88
Abbildung 6.1: Referenzarchitektur.....	100
Abbildung 6.2: Instanziierung der Referenzarchitektur.....	102
Abbildung 6.3: Workflow Arbeitsplatz.....	103
Abbildung 6.4: Rückfrage im Kontext eines Vorgangs.....	105
Abbildung 6.5: Workflow-Benutzungsschnittstelle und Urlaubsantragsformular.....	108
Abbildung 6.6: Urlaubskalender.....	109
Abbildung 7.1: Werkzeug zur Modellierung von Kompensation.....	117
Abbildung 8.1: Object Management Architecture (OMA).....	130

# Tabellenverzeichnis

Tabelle 3.1: Verwendete Komponenten in SWATS Prototyp Release 2.3. ....	37
Tabelle 6.1: Klassifikation von Groupware-Systemen .....	93
Tabelle 6.2: Unterschiede: Groupware <-> Workflow-System .....	95
Tabelle 8.1: Vergleich von OSF DCE und OMG OMA. ....	131
Tabelle 8.2: Dienstgüte für Medien. ....	152

# 1 Einleitung

In diesem Abschlußbericht werden die durchgeführten Aktivitäten und erzielten Ergebnisse des Projekts PoliFlow aus Sicht des Instituts für Parallele und Verteilte Höchstleistungsrechner (IPVR) der Universität Stuttgart beschrieben. Der Berichtszeitraum geht vom 1. Juni 1994 bis 31. Dezember 1998.

PoliFlow war ein Forschungsprojekt der Förderinitiative POLIKOM, das vom BMBF gefördert wurde. Zielsetzung von PoliFlow war die effektive Unterstützung von weit verteilten Vorgängen in der öffentlichen Verwaltung. Hierfür wurde von PoliFlow eine Lösung entwickelt, bei der verschiedene innovative Technologien wie Workflow-Management, Dokumenten-Management, Groupware- und synchrone Telekooperationsdienste über Intra- bzw. Internet integriert wurden. Eine relevante Themenstellung war dabei die Integration von Workflow-, Telekooperations- und Multimediatechniken. Neben einer leistungsfähigen Steuerung der Arbeitsabläufe wird durch die Integration die kooperative Bearbeitung von standort- und organisationsübergreifenden Aufgaben unterstützt. Innerhalb der Geschäftsvorgänge konnten dabei multimediale Dokumente problemlos ausgetauscht und gemeinsam und gleichzeitig bearbeitet werden.

Das Projekt wurde von einem Konsortium bearbeitet, bestehend aus der Hewlett-Packard GmbH, dem Institut für Arbeitswissenschaft und Technologiemanagement (IAT) und dem Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR) sowie dem Anwender, der Bundesvermögensabteilung Berlin. Durch eine benutzerfreundliche, aufgabenangemessene und effiziente Unterstützung der Geschäftsvorgänge hat PoliFlow die Qualität von Arbeitsplätzen und Arbeitsprozessen beim Anwender gesteigert und dadurch quantitative und qualitative Nutzenpotentiale erzielt.

In diesem ersten Kapitel wird in die Problemstellung eingeführt und die Zielsetzung des Projekts aufgezeigt. Am Ende wird noch eine Übersicht über die folgenden Kapitel gegeben. Dabei wird in diesem Kapitel auch auf das Gesamtprojekt eingegangen, um den Projektrahmen aufzuzeigen, in den folgenden Kapiteln wird dann im wesentlichen auf die Aktivitäten und Ergebnisse des IPVR eingegangen.



## 1.1 Problemstellung

In Deutschland ist der Trend erkennbar, Behörden als Dienstleister auszurichten. Als wesentliche Vorteile der Orientierung am Dienstleistungsgedanken werden Bürgernähe, Kostenersparnis und Qualitätsverbesserungen angeführt. Behördenübergreifende Vorgänge sollen auf Dienstleistungsbasis ebenfalls effizienter durchgeführt werden. Dem Bürger oder anderen Behörden sollen durch diese Philosophie erhebliche Vorteile geboten werden. Dabei spielen vor allem zwei Qualitätsmerkmale eine wichtige Rolle: einerseits soll die behördliche Leistungserbringung effizienter gestaltet werden, andererseits ist sicherzustellen, daß die Leistungen weiterhin durch kompetente Behördenmitarbeiter erbracht werden ohne die Ergebnisqualität zu mindern.

Diesen beiden Ansprüchen muß man gerecht werden, damit Behörden erfolgreich in Dienstleistungsunternehmen der öffentlichen Hand umgewandelt werden können. Dieses Ziel kann bei der Komplexität der von den Behörden zu erledigenden Aufgaben nicht ohne Unterstützung moderner Informationstechnologien erreicht werden.

Informationstechnologien, wie sie heute bereits zur Erbringung von Dienstleistungen in der privaten Wirtschaft angeboten werden, können nicht direkt auf Behörden übertragen werden, denn Behörden operieren unter signifikant anderen Rahmenbedingungen. Dazu zählen beispielsweise politische Vorgaben, rechtliche Grundlagen und in verstärktem Maße Datenschutz und Datensicherheit. Darüber hinaus sind die Geschäftsfelder und die Kundenbasis größer und heterogener als bei gewöhnlichen Dienstleistungsunternehmen.

Als Basistechnologien zur Unterstützung der Dienstleistungserbringung im Behördenumfeld nehmen besonders Workflow-Management-Systeme und Telekooperationssysteme eine wichtige Rolle ein. Workflow- und Telekooperationssysteme sind vor allem notwendig, um eine effiziente Koordination und Kooperation unter den Dienstleistungserbringern in den Behörden, wie auch zwischen den Dienstnutzern, also den Bürgern, und den Behörden zu realisieren. Dabei werden an die eingesetzten Systeme hohe Anforderungen bezüglich Verfügbarkeit, Interoperabilität, Sicherheit und Flexibilität gestellt.

Die Verfügbarkeit ist vor allem für die Bürger relevant, denn sie müssen die Dienste einfach, schnell und sicher erreichen können. Darüber hinaus ist eine intuitive Benutzerführung für den Dienstzugang notwendig, um eine erfolgreiche Einführung von IT-Technologien zur Unterstützung von Dienstleistungen im Behördenbereich zu erreichen.

Die Flexibilität und Interoperabilität ist zwingende Voraussetzung für eine erfolgreiche Einführung und Vernetzung der Dienstleistungen. Aufgrund der räumlichen Verteilung, der heterogenen existierenden IT-Infrastruktur und der spezifischen Aufgabenstellung von Behörden ist eine ein-

heitliche Technologieauswahl nicht realistisch. Zur Gewährleistung der erforderlichen Koordination und Kooperation sind daher Technologien einzusetzen, welche die Interoperabilität innerhalb und zwischen Behörden und dem Bürger sicherstellen.

Diese Fragen, in Bezug auf die zu beobachtende Ausrichtung von Behörden als Dienstleister, zeigen, daß neue Technologien und Architekturen zur Unterstützung von Vorgängen und Kooperation eingesetzt werden müssen. Besonders die gewünschte Bürgernähe und Kooperation zwischen Behörden erfordern innovative Lösungsansätze. Im Rahmen des Projekts PoliFlow wurden solche Lösungsansätze bei der OFD Berlin, dem Anwender, erarbeitet und eingeführt.

## 1.2 Zielsetzung und Ergebnisse

Das Konsortium beteiligte sich an dem Forschungsvorhaben “Telekooperation” mit einem Beitrag zum Thema Vorgangsbearbeitung mit dem Ziel, auf der Basis bereits vorhandener Technologien und Prototypen allgemeine und übertragbare Telekooperationstechniken zu konzipieren, zu entwickeln und am konkreten Anwendungsfall zu erproben. Diese Prototypen führten im Umfeld einer geographisch verteilten Verwaltungsorganisation zu signifikanten Effizienzsteigerungen in den Verwaltungsvorgängen.

Im Rahmen des Projekts wurde auf die Sicherheit, Verfügbarkeit und Flexibilität verteilter, asynchroner Verwaltungsprozesse sowie deren Integration mit synchronen Arbeitsformen besondere Rücksicht genommen. Bei der Konzeption des konkreten Anwendungssystems wurden arbeitswissenschaftliche und software-ergonomische Erkenntnisse bezüglich Aufgabenangemessenheit, Anpaßbarkeit und Handlungsorientierung berücksichtigt. Das Projekt hat neueste wissenschaftliche Erkenntnisse und Methoden in einsetzbaren Anwendungssystemen umgesetzt.

Aufgrund des komplexen Umfelds und der Projektdauer wurde das Projekt in zwei Phasen gegliedert. In der ersten Projektphase wurden verschiedene Telekooperationstechniken und -verfahren erprobt, im Feldversuch eingesetzt und deren effektiver Einsatz demonstriert. Die Planung und die Ziele der zweiten Phase wurde aufgrund der Ergebnisse der ersten Phase angepaßt und verfeinert.

Im Einzelnen wurden in der ersten Phase des Projekts folgende Ziele verfolgt:

- Geschäftsprozeßoptimierung: Analyse der Geschäftsprozesse mit dem Ziel, Optimierungspotentiale zu erkennen und umzusetzen und somit die Verwaltungsproduktivität zu erhöhen.
- Methoden und Werkzeuge zur Vorgangmodellierung: Konzepte, Notationen und Regeln zur Erfassung und Beschreibung von Vorgängen (Analyse, Modellierung) zu erstellen.

- Integration unterschiedlicher Arbeitsformen: Entwicklung software-ergonomischer Benutzeroberflächen und technologischer Grundlagen zur Flexibilisierung der asynchronen Bearbeitung strukturierter Vorgänge in Workflow-Management-Systemen.
- Anwendungsentwicklung: Ziel der Anwendungsentwicklung im Rahmen des Projekts war es, eine möglichst umfangreiche und aussagekräftige Basis für die durchzuführenden Feldversuche zur Verfügung zu stellen.
- Einführungsstrategien: Neue organisatorische Maßnahmen sowie die Einbettung der Vorgangsteuerungssysteme in die zukünftigen organisatorischen Abläufe waren durch geeignete Personalentwicklungsmaßnahmen zu vermitteln.
- Benutzerorientierung: Handlungsorientierung und Erfahrungswissen der Benutzer bilden die Grundlage für die Anwendungsentwicklung und die Gestaltung neuer Arbeitsabläufe. Eine aufgaben- und benutzerorientierte Gestaltung wurde durch Berücksichtigung und Umsetzung software-ergonomischer und arbeitswissenschaftlicher Anforderungen erreicht.
- Verfügbarkeit und Zuverlässigkeit: Untersuchungen und Erprobung von Methoden zur Erreichung von Revisionsfähigkeit, Konsistenz und hoher Verfügbarkeit bei lang laufenden, verteilten Vorgängen.
- Wiederverwertbarkeit und Interoperabilität: Konzeption und prototypische Realisierung einer vorgangsgesteuerten Anwendung auf der Basis objektorientierter Technologien. Integration bereits verfügbarer Dienste in einen Object Request Broker (ORB), der auf bekannten Standards (CORBA) aufsetzt.
- Sicherheit: Einsatz und Integration bestehender und neuer Konzepte und Technologien in die projektierte Anwendung und das allgemeine Workflow-Management-System.
- Skalierbarkeit: Untersuchung und Beschreibung der besonderen Anforderungen, die sich für die Komponenten einer vorgangsgesteuerten Anwendungsumgebung aus der angestrebten großen Anwenderzahl beim IVBB ergeben.
- Übertragbarkeit: Um eine möglichst hohe Breitenwirkung der Projektergebnisse zu erzielen, wurden die Ergebnisse verallgemeinert und hinsichtlich ihrer Übertragbarkeit geprüft.

Die Zielsetzung für die zweite Phase wurde anhand der Ergebnisse der ersten Phase des Gesamtprojekts neu ausgerichtet. Diese Weiterentwicklung sollte den Erwartungen des Anwenders bezüglich der Unterstützung des Kompetenzzentrums der OFD besser gerecht werden. Das Transparentmachen der komplexen Struktur bisheriger Prozesse in Phase I hat auf Anwenderseite den Bedarf entstehen lassen, die Prozesse nicht nur informationstechnisch optimiert und telekooperativ unterstützt abzuwickeln, sondern grundsätzlich im Rahmen der gegebenen Möglichkeiten, neu und überschaubarer, in Form von kleineren, vereinfachten Teilprozessen mit erhöhter Handlungsautonomie zu gestalten. Die Zielsetzungen für die zweite Phase wurde von der Konzeption zur Realisierung weiterentwickelt:

- Es wurde eine flächendeckende Basistechnologie zur Telekooperation eingeführt.
- Die Modellierung innovativer behördenübergreifender Zusammenarbeit wurde mit einer neuen Methodik für kooperative Prozeßformen dargestellt.
- Es wurde eine bessere Integrationsfähigkeit technischer Konzeptionen in das organisatorische Umfeld der öffentlichen Verwaltung ermöglicht.
- Telekooperative Prozesse der Informationslogistik werden schwerpunktmäßig und prozeßabdeckend unterstützt.
- Durch die Integration von Workflow mit Technologien für synchrone Telekooperation sowie mit der funktionalen Erweiterung hinsichtlich Flexibilität und Sicherheit ist eine Infrastruktur zu entwickeln, mit der verwaltungsspezifische Vorgänge einheitlich und umfassend unterstützt werden können.
- Auf der Basis der bekannten Defizite der bisher eingesetzten Unterstützungstechnologien sowie des erhobenen Unterstützungsbedarfs, wurde eine Lösung konzipiert.

Neben dem starken Realisierungsaspekt wurde die technologische Weiterentwicklung weiterverfolgt und nach Möglichkeit wurden neue verbesserte Technologien ins Anwendungsfeld übernommen. Entwicklungs- und Forschungsarbeit wurde vor allem in den Bereichen Anpassungsfähigkeit, Zuverlässigkeit und Sicherheit von Workflow-Management-Systemen durchgeführt.

### **1.2.1 Arbeitsbereiche und Ergebnisse am IPVR**

Das IPVR arbeitete im Rahmen des Gesamtprojekts PoliFlow intensiv mit. Es wurde sowohl in Arbeitsgruppen von PoliFlow wie auch der Förderinitiative POLIKOM mitgearbeitet. Der Schwerpunkt der Arbeiten am IPVR lag in den wissenschaftlichen/technologischen Arbeitsbereichen die im Projektplan definiert waren. Die Arbeitsbereiche, deren Inhalt und Ergebnisse sind im folgenden kurz aufgeführt.

- Technologie Evaluierung (TE): In PoliFlow war ein weit verteiltes Anwendungsszenario gegeben, aufgrund dessen waren Technologien zu untersuchen, die helfen die räumliche Trennung der Arbeitsplätze zu überbrücken und die Vorgangsbearbeitung effizienter zu gestalten. Zu den Technologien, die das weit verteilte Anwendungsszenario möglichst optimal unterstützen, zählen Systemdienste zur Unterstützung von verteilten Systemen, Multimedia-Technologien zur Vermeidung von Medienbrüchen und Telekooperationstechniken zur Gestaltung effektiver Gruppenarbeit.
- Konsistenzsicherende Mechanismen (CAM): Der Arbeitsbereich hatte Verfahren zur Konsistenzsicherung für Workflows zum Inhalt. Er befaßte sich mit der Anforderungsanalyse an die Verfahren zur Konsistenzsicherung sowie der Auswahl und der Identifikation eines Basismo-

dells. Das in CAM definierte und ausgewählte Basismodell zur Sicherstellung der Konsistenz von langandauernden Vorgängen ist grundsätzlich sehr gut geeignet. Bevor das Basismodell jedoch in der Anwendungsumgebung erfolgreich eingesetzt werden kann, ist noch die Integration in das gewählte Workflow-Management-System erforderlich.

Die Arbeitsbereiche TE und CAM wurden in der ersten Phase von PoliFlow bearbeitet.

- Gemischte Arbeitsformen (MWM): Im Rahmen des Arbeitsbereichs MWM wurden die Anforderungen an eine Integration von Workflow-Systemen und Telekooperationswerkzeugen zur Unterstützung gemischter Arbeitsformen untersucht. Es wurde im Bereich gemischte Arbeitsformen der Schwerpunkt auf die nahtlose Integration von Applikations- und Telekooperationskomponenten gelegt. Eine Referenzarchitektur, eines Systems zur Unterstützung gemischter Arbeitsformen, wurde entwickelt. Darauf aufbauend wurde ein Prototyp entwickelt, in dem die Kernaspekte realisiert sind.
- Adaptive Workflow Systeme (AWS): Flexibilität und Anpassungsrechte stellen wichtige Leistungsmerkmale für Workflow-Management-Systeme dar. Um diese Anforderungen zu erfüllen, wurden im Arbeitsbereich AWS Konzepte und Verfahren entwickelt, mit deren Hilfe Workflows während der Ausführung verfeinert, erweitert oder verändert werden können. Spezielle Schwerpunkte bildeten dabei die Modellierung dynamischer Anpassungsrechte und die Berücksichtigung struktureller Integritätsbedingungen. Durch einen graphischen Workflow-Editor wurde dem Anwender die Definition bzw. Integration von Ad-hoc-Workflows und -Modifikationen ermöglicht. Der Arbeitsbereich AWS wurde während der gesamten Laufzeit von PoliFlow bearbeitet.

Die Arbeitsbereiche MWM und AWS wurden während der gesamten Laufzeit von PoliFlow bearbeitet.

- Workflow Realisierung (WR): Im Arbeitsbereich "Workflow Realisierung" lag die Problemstellung darin Vorgangsbearbeitung in Behörden durch Workflow-Management zu unterstützen. In der ersten Phase des Projekts zeigten sich dabei unterschiedliche Anforderungen des Umfelds, die die Einführung von Workflow-Technologie beeinflussen, wie der verteilte Zugriff auf das Workflow-System, die Unterstützung heterogener IT-Infrastruktur, den einfachen, mehrgestaltigen Zugang zum Workflow-System und eine offene und erweiterbare Architektur. Unter Verwendung modernster Technologien wurde ein System entworfen, das kooperatives Arbeiten im Kontext von Vorgangsbearbeitung in weit verteilten Umgebungen unterstützt, und zwar insbesondere weitgehend unabhängig von der zugrundeliegenden Plattform auf Client-Seite. Als Basis-Infrastruktur dient das Intra- bzw. das Internet. Die klientenseitige Applikationsentwicklung erfolgte in Java.

- **Sichere Workflow Systeme (SWS):** Geschäftsvorgänge in Behörden und Unternehmen stellen fast ausnahmslos Anforderungen an die Sicherheit der Daten und des Datenzugriffs. Diese sicherheitsrelevanten Anforderungen werden im allgemeinen bei der Geschäftsprozessanalyse erfaßt. Bisher war eine direkte Modellierung dieser Anforderungen in Workflow-Definitionen nicht oder nur unzureichend möglich. In dem Arbeitsbereich SWS sollten Modelle und Konzepte entwickelt werden, um die Sicherheitsanforderungen bei der Workflow-Definition zu berücksichtigen.

Es wurden solche Modelle entwickelt, die die Sicherheitsanforderungen in Workflows einerseits organisationsübergreifend beschreiben, andererseits aber auch die Belange einzelner Sicherheitsdomänen berücksichtigen. Auf der Basis des “Sicherheitsmodells für interorganisationelle Workflows” und des “Aktivitätenbasierten Sicherheitsmodells” wurde ein Prototyp entwickelt, bei dem von der Modellierung über die Vorgangsausführung bis zur Aktivitätsbearbeitung die Sicherheitsanforderungen integral und durchgängig unterstützt werden.

Die Arbeitsbereiche WR und SWS wurden in der zweiten Phase von PoliFlow bearbeitet.

## **1.3 Struktur des Berichts**

Der Bericht gliedert sich in drei Teile. In diesem ersten Teil wurde ein Überblick über die Ziele und Ergebnisse des Projekts PoliFlow gegeben. Außerdem wird im folgenden Kapitel auf den Projektverlauf, das Projektmanagement und die Öffentlichkeitsarbeit eingegangen, wobei Kennzahlen und Informationen zu allen Bereichen enthalten sind.

Im zweiten Teil des Berichts wird über die technisch/wissenschaftlichen Arbeitspakete, die am IPVR bearbeitet wurden, berichtet. Zu diesen Arbeitspaketen zählen die Pakete: Workflow Realisierung (WR), Adaptive Workflow Systeme (AWS), Sichere Workflow Systeme (SWS), Gemischte Arbeitsformen (MWM), Konsistenzsichernde Mechanismen (CAM) und Technologie Evaluierung (TE).

Im dritten Teil werde die wichtigsten Ergebnisse nochmals herausgestellt und der Bericht endet mit einer abschließenden Zusammenfassung.

## 2 Projektverlauf

### 2.1 Projektmanagement

#### 2.1.1 POLIKOM

Damit die Ergebnisse der einzelnen POLIKOM-Projekte optimal verwertet werden konnten, war eine besonders gute Abstimmung der Konsortien erforderlich. Die inhaltliche Zusammenarbeit wurde daher intensiv in den Bereichen gefördert, bei denen ähnliche Anforderungen vorlagen oder Überschneidungen in den Arbeitspaketen vorhanden waren. Neben fachlichen Themen wie den Sicherheitsaspekten oder dem organisatorischen Handlungsbedarf im Anwendungsfeld stand auch eine koordinierte Öffentlichkeitsarbeit im Vordergrund.

Die POLIKOM-Koordinierungsgruppe sorgte für die nötige Abstimmung zwischen den Projekten, dem Projektträger (DLR) und dem Förderer (BMBF). Durch die konstante Mitarbeit in der Koordinierungsgruppe und maßgebliche inhaltliche Arbeiten im Arbeitskreis *Sicherheit* beteiligte sich das IPVR intensiv an den projektübergreifenden Aktivitäten; der Arbeitskreis *Organisation* wurde durch aktive Mitarbeit an Berichten, Präsentationen und Sachfragen unterstützt.

Die Ergebnisse der projektübergreifenden Arbeiten können abschließend als erfolgreich beurteilt werden, insbesondere da die Zusammenarbeit durch projektspezifische Zielsetzungen und eine gewisse Konkurrenzsituation der Industriepartner doch in einigen Situationen die Arbeit erschwerte.

#### 2.1.2 POLIFLOW

PoliFlow hatte wie die anderen POLIKOM-Projekte eine Verbundstruktur und setzte sich aus einem Industriepartner, Forschungseinrichtungen und einem Anwender zusammen. Durch diesen heterogenen Verbund konnten einerseits die gewünschten Synergieeffekte erzielt und der Zyklus zur Erstellung der innovativen Anwendungssysteme verkürzt werden, andererseits stellte diese Struktur auch außerordentliche Anforderungen an das Projektmanagement der beteiligten Partner. Hierbei waren die Koordination aller Teilaufgaben und eine gute Kooperation Voraussetzungen zur Erfüllung der gemeinsamen Ziele.

Organisatorisch wurde diese Zusammenarbeit durch mehrere Maßnahmen unterstützt:

- Ein abgestimmtes Berichtswesen - das Termine, Strukturen und Umfänge von unterschiedlichen Berichten festlegte - unterstützte die Projektkontrolle, förderte die interne Kommunika-

tion und war Hauptbestandteil der Rechenschaftslegung gegenüber dem Projektträger. Neben organisatorischen Statusberichten (in der ersten Projektphase jeden zweiten Monat) wurden inhaltliche Zwischenberichte (halbjährlich) und Fachberichte (bei Beendigung der Arbeitspakete oder im Zusammenhang mit Publikationen) erstellt.

- Eine interne Fortschritts- und Qualitätskontrolle wurde durch Reviews auf verschiedenen Ebenen erzielt. Neben den Reviews aller Dokumente (durch alle Projektpartner) wurden die Ergebnisse in regelmäßigen *Core-Team-Meetings* der Projektleiter überprüft, wobei auch die weitere Planung abgestimmt wurde. Halbjährlich fanden *Management-Reviews* statt, in denen der aktuelle Stand des Projektes und das weitere Vorgehen dem Top-Management der Projektpartner vorgestellt wurde. In einer ähnlichen Form wurden dann auch *Projekt-Reviews* mit dem POLIKOM-Beiratsmitglied Herrn Prof. Reichwald durchgeführt. Hierdurch wurden kontinuierlich Anregungen und Zielsetzungen der Förderinitiative aufgegriffen und berücksichtigt.
- Der fachliche Austausch, der erforderlich war um die gemeinsamen Aufgaben zu bewältigen, wurde durch formale *Team-Meetings* (mit allen Projektbeteiligten) und informale Arbeitstreffen in geeigneten Kleingruppen unterstützt.

Ähnlich wie im POLIKOM-Kreis hat sich auch hier gezeigt, daß bei dieser Projektstruktur ein enormer Kommunikations- und Kooperationsbedarf besteht. Durch Projektrahmenpläne werden Aufgabenbereiche definiert und Zielsystem gebildet, in denen die kumulierten Teilziele der Arbeitspakete und Arbeitsbereiche das Gesamtziel des Projektes ergeben sollen. Durch ungeplante Ereignisse, zusätzliche Aufwände oder sich ändernde Prioritäten bzw. veränderte Individualziele einzelner Partner entstehen dann immer wieder Problemsituationen, die teilweise gemeinsam gelöst werden können, die teilweise jedoch auch zu einer Reduzierung des Gesamterfolgs führen können.

Insbesondere die wissenschaftlichen Begleiter sind in solchen Projektstrukturen häufig stärker an den ursprünglichen Projektplan gebunden, da deren konzeptuellen Arbeiten entweder bestimmte Vorbedingungen oder auch eine etwas längere und kontinuierliche Bearbeitungszeit erfordern.

### **2.1.3 PoliFlow am IPVR**

Die Aufgabenstellung des IPVR in PoliFlow war die wissenschaftliche Begleitung im Technologiebereich Workflow-Management. Um den Anforderungen aus dem Anwendungsfeld gerecht zu werden, mußten grundlegende Erweiterungen an der bestehenden Workflow-Technologie vorgenommen werden. Wegen der fachübergreifenden Problemstellungen bei der Entwicklung und beim Einsatz von WFMS brachte das IPVR Spezialwissen von zwei Abteilungen (*Anwendungssoftware* und *Verteilte Systeme*) in das Projekt ein und paßte sich auch mit der internen Organisation an die Projektstruktur an.



Das Top-Management wurde von den beiden Professoren der Abteilungen gebildet, und die Verwaltung des Projektes erfolgte durch eine dritte Abteilung (*Infrastruktur*) des Instituts.

Zu Beginn wurden die vier bis fünf Mitarbeiter/innen zusätzlich von zwei Projektleitern unterstützt. In der zweiten Phase wurden diese Aufgaben dann von einem Projektleiter übernommen, zu dessen Aufgabenbereich die Koordination, Kontrolle und Qualitätssicherung der Arbeiten am IPVR sowie administrative Aufgaben, Gremienarbeit und Öffentlichkeitsarbeit zählten.

Die Arbeitspakete wurden durch die Projektmitarbeiter/innen fachkundig und selbständig bearbeitet, wobei eine kontinuierliche Personalpolitik - mit wenig Wechsel auf den Arbeitsbereichen - zum Gesamterfolg beigetragen hat.

Die Arbeiten wurden in vielen Bereichen durch das Mitwirken von Studenten/innen als wissenschaftliche Hilfskräfte oder als Studien- bzw. Diplomarbeiter unterstützt.

Da die Arbeitspakete des IPVR neuartige, komplexe Konzepte und deren prototypische Implementierung auf der Basis bestehender Produkte beinhalteten, bestand eine Vielzahl von Abhängigkeiten zu anderen Arbeitspaketen - am IPVR und bei den Partnern. Durch diese Abhängigkeiten und durch immer wieder erforderliche Änderungen der Vorgehensweise im Anwendungsfeld, der verwendeten Technologien (z.B. Intranet-Technologien) und der strategischen Produkte (z.B. Workmanager vs. Changengine) mußte die Arbeiten häufig unter erschwerten Umständen geleistet werden.

Durch geeignete Maßnahmen wie der Überarbeitung des Rahmenplans für die zweite Phase, der Aufstockung der Fördermittel für zusätzlichen Implementierungsaufwand und einer kostenneutralen Verlängerung, wurde diesen Änderungen Rechnung getragen. Trotz dieser Umstände und trotz zusätzlicher ungeplanter Aufgaben konnten alle Arbeitspakete ausgesprochen erfolgreich beendet werden, was erheblich zum Gesamterfolg des Verbundprojektes beigetragen hat.

Im Rahmen der oben genannten Maßnahmen wurden der Arbeits-, der Zeit- und der Kostenplan des IPVR eingehalten.

## 2.2 Öffentlichkeitsarbeit

Die Öffentlichkeitsarbeit hatte im Projekt einen besonders hohen Stellenwert, wodurch auch hohe Arbeitsaufwände erforderlich wurden. Das große Interesse der Öffentlichkeit an der Konzeption, dem Verlauf und den Ergebnissen des Projektes ist darauf zurückzuführen, daß verschiedene Adressatengruppen durch unterschiedliche innovative Schwerpunkte angesprochen wurden.

Durch die Struktur und den Umfang von POLIKOM war das politische Interesse der Medien und der Bevölkerung geweckt, weshalb neben den fachlichen Veröffentlichungen auch Pressearbeit und allgemeine Öffentlichkeitsarbeit zu leisten war. Resultierende Maßnahmen am IPVR waren

verschiedene Artikel in der regionalen und internen Presse (Stuttgarter Zeitung, UNI-Kurier, Jahresberichte des Instituts) sowie bürgernahe Präsentationen (Tag der offenen Tür) und umfangreiche Informationen im *World Wide Web* (WWW).

Durch die Verwendung neuester Technologien und den starken Anwendungsbezug wurden verschiedene Benutzer angesprochen. Neben Führungskräften und DV-Spezialisten interessierten sich auch Endanwender aus der öffentlichen Verwaltung für die Chancen und Risiken der elektronischen Vorgangsbearbeitung. Das wurde besonders bei Veranstaltungen wie den POLIKOM-Konferenzen, Präsentationen beim Anwender, aber auch auf der CeBIT deutlich.

Da die Hauptaufgabe des IPVR in der wissenschaftlichen Begleitung des Projektes lag, wobei wiederum verschiedene sehr aktuelle Forschungsschwerpunkte (Flexibilität, Sicherheit, Kooperation) bearbeitet wurden, bestand ein großes Interesse von Seiten anderer Forschungseinrichtungen, Systemherstellern und Fachgremien an den Ergebnissen des IPVR. Aus diesem Grund entstand eine Vielzahl von Kontakten und Kooperationen zu universitären Forschungsgruppen, Facharbeitsgruppen und Standardisierungsgremien.

### **2.2.1 Präsentationen**

Diesem großen Interesse wurde im Projekt Rechnung getragen, indem eine Vielzahl von Präsentationen durchgeführt wurde. Das Spektrum der Präsentationen reicht von konzeptionellen, wissenschaftlichen Vorträgen (z.B. auf Konferenzen oder an anderen Forschungseinrichtungen) über anwendungsorientierte Vorträge und Seminare (z.B. IAO-Forum) bis hin zu Systempräsentationen von Workflow- und Internet-Technologien.

Bei den beiden POLIKOM-Konferenzen lag der Schwerpunkt der Präsentationen auf den anwendungsspezifischen Lösungen für die öffentliche Verwaltung. Hierbei standen besonders die Vorgangsteuerung über Inter-/Intranet, die Integration mit anderen PoliFlow-Anwendungen und die Übertragbarkeit der Ergebnisse im Vordergrund.

In zwei aufwendigen Ausstellungen auf der CeBIT - in den Jahren 1996 (Integration von Workflow und synchronen Telekooperationsdiensten) und 1998 (Innovative Workflows im Internet) - standen neben der Architektur auch die verwendeten Technologien und Konzepte im Vordergrund der Präsentationen. Die CeBIT'96, mit der Ausstellung auf dem BMBF-Stand, war insbesondere aufgrund der verwendeten Hochtechnologien (Videokonferenzen, Shared Editing, Multimediaanwendungen über ATM) und neuartigen Ideen (Workflow und synchrone Telekooperation) ein Erfolg. Noch wertvoller waren die Erfahrungen und die Ergebnisse der CeBIT'98, da hier doch die vollständigen Lösungskonzepte und deren Realisierungen einem breiten Publikum vorgestellt wurden. Der Erfolg dieser Ausstellung ist besonders hoch einzuschätzen, da alle Teile der Präsentation (Workflow über Internet, Sicherheit in WFMS und Adaptive WFMS) ein extrem heteroge-

nes Publikum überzeugen konnten. Dies äußerte sich durch konkretes Interesse und gleichermaßen hohe Anerkennung von Anwendern, Systemherstellern und Forschungsgruppen.

Die folgende Übersicht gibt einen Überblick über die wesentlichen Systemdemonstrationen und Vorträge. Darüber hinaus wurden alle Publikationen (vgl. Kapitel 2.2.2) durch Vorträge auf den entsprechenden Konferenzen begleitet.

- Synchroner Rückfragen in Workflow-Management-Systemen, Demonstration auf der CeBIT (BMBF-Stand). Hannover, 14.-21.3.1996.
- Workflow via Internet, Vortrag und Demonstration auf dem IAO-Forum Internet & Intranet. Stuttgart, 12.-14.11.1996.
- Workflow via Internet, Demonstration auf der POLIKOM-Statuskonferenz. Bonn, 28.1.1997.
- Stuttgart's Workflow and Telecooperation System - SWATS, Vortrag und Demonstration in den HP Research Labs. Palo Alto (USA), 3.-4.3.1997.
- Workflow in der öffentlichen Verwaltung, Demonstration auf dem IAO-Forum Software-Technologien in der Praxis. Stuttgart, 22.4.1997.
- Innovatives Workflow-Management, Demonstration beim Tag der offenen Tür der Universität Stuttgart, 21.6.1997.
- Workflow-Management Technologie im Verbundprojekt PoliFlow, Vortrag an der Uni Erlangen, 8.7.1997.
- Innovative Vorgangunterstützung auf dem Internet, Vortrag auf dem BMBF-Forschungsforum. Leipzig, 19.9.1997.
- Innovative Vorgangssteuerung für behördliche Dienstleistungen, Vortrag auf dem Workshop Computerunterstützte Gruppenarbeit und Verwaltungsreform. Hamburg, 29.9.1997.
- Flexible Workflows im Verbundprojekt PoliFlow, Vortrag an der Universität Münster, 31.10.1997.
- Elektronische Vorgangsbearbeitung im Internet, Demonstration beim Uni-Tag der Universität Stuttgart, 12.11.1997.
- Workflow-Systeme in der öffentlichen Verwaltung, Seminar auf dem IAO-Forum Innovative öffentliche Dienstleister. Stuttgart, 19.11.1997.
- Anpassungsfähigkeit von WFMS: Ursachen, Chancen und Grenzen, Vortrag auf dem an der Universität Stuttgart ausgerichteten Treffen des GI-Arbeitskreises Workflow. Stuttgart, 19.2.1998.
- Innovative Workflows im Internet, Ausstellung auf dem Hochschulstand Baden-Württemberg auf der CeBIT 1998. Hannover, 18.-25.3.1998.
- Flexibilität und Sicherheit in WFMS, Präsentation auf der POLIKOM-Abschlußkonferenz. Bonn, 24.11.1998.

## 2.2.2 Veröffentlichungen

Neben zahlreichen internen Fachberichten aus den Arbeitsbereichen und offiziellen Projektberichten (Status- und Zwischenberichte), konnten aus dem Projekt wissenschaftliche Beiträge in namhaften nationalen und internationalen Zeitschriften bzw. Tagungsbänden veröffentlicht werden. Besondere Aufmerksamkeit fanden dabei die konzeptuellen Arbeiten aus den Forschungsbereichen *Anpassungsfähige Workflow-Systeme* und *Sichere Workflow-Systeme*.

Die folgende Liste gibt eine Übersicht über ausgewählte Publikationen aus dem Projekt:

- Kindler T., Soyez T. : Modelling Security for Integrated Enterprise Workflow and Telecooperation Systems. In Proceedings of the Fifth IEEE Workshops of Enabling Technologies : Infrastructure for Collaborative Enterprises. Stanford (USA), 1996.
- Siebert, Reiner : Adaptive Workflow for the German Public Administration. In Proc. of the 1st Int. Conf. on Practical Aspects of Knowledge Management (PAKM'96) : Workshop on Adaptive Workflow. Basel (Schweiz), 1996.
- Siebert R., Kindler T., Soyez, T. : Integrated Workflow and Telecooperation Support for the German Government. In Proc. of the 1997 ACM Symposium on Applied Computing (SAC'97). San Jose (USA), 1997.
- Kindler T., Kulendik O., Siebert R., Soyez T.: PoliFlow. In Informatics Cybernetics and Computer Science (ICCS-97): Collected Volume of Scientific Papers. Donetsk State Technical University. Donetsk (Ukraine), 1997.
- Siebert, Reiner : Anpassungsfähige Workflows zur Unterstützung unstrukturierter Vorgänge. In EMISA Forum, Band 1/1998. Proceedings des EMISA-Fachgruppentreffens 1997: Workflow-Management-Systeme im Spannungsfeld einer Organisation. GI, 1997.
- Siebert, Reiner : Adaptive Workflows im Verbundprojekt PoliFlow. Tagungsband zur Fachtagung Workflow-Management in Geschäftsprozessen im Trend 2000. Schmalkalden, 1997.
- Kindler T., Kulendik O., Siebert R., Soyez T. : Innovative Vorgangssteuerung für behördliche Dienstleistungen. Tagungsband zum GI-Workshop 'Computerunterstützte Gruppenarbeit und Verwaltungsreform'. Hamburg, 1997.
- Kindler, Thomas : Activity-Based Security in Intranet and Internet Workflows. In Proceedings of the Workshop on Information Technologies and Systems (WITS). Atlanta (USA), 1997.
- Siebert Reiner : An Open Architecture For Adaptive Workflow Management Systems. In Proceedings of the 3rd biennial World Conference on Integrated Design and Process Technology (IDPT): Issues and Applications of Database Technology. SDPS, Austin (USA) 1998.
- Siebert R., Weske M. : Flexibilität und Kooperation in Workflow-Management-Systemen: Tagungsband zum Workshop der D-CSCW 1998. Schriftenreihe Angewandte Mathematik und Informatik der Universität Münster, Band 18/98-1. Münster, 1998.

Darüber hinaus wurden während des Projektes immer wieder Artikel für die regionale/überregionale Presse verfaßt (Uni-Kurier, Stuttgarter Zeitung) und an gemeinsamen Artikeln für Fachzeitschriften mitgewirkt (Behördenspiegel, Computerwoche etc.).

Um die Demonstrationen und Präsentationen durch Informationsmaterial zu unterstützen wurden verschiedene PoliFlow-Informationsblätter mitgestaltet und ausführliche Broschüren zu den folgenden IPVR-Arbeitsbereichen erstellt und gedruckt:

- PoliFlow (Deutsch und Englisch): 03/96
- PoliFlow am IPVR 1-3 (Deutsch und Englisch): 03/96, 01/97 und 12/97
- SWATS 1-3 (Deutsch und Englisch): 02/97, 03/98 und 10/98
- Adaptives Workflow-Management (Deutsch): 10/98
- IPVR-Jahresberichte (Deutsch und Englisch): 05/95, 05/96, 05/97, 05/98

### 2.2.3 Das PoliFlow-Infonet am IPVR

Eine Maßnahme um den hohen Anforderungen im Bereich Öffentlichkeitsarbeit gerecht zu werden, war die Erstellung des *Infonet* zum Projektteil des IPVR. In über 500 Dateien (ca. 30 MB Daten) findet der Besucher im World Wide Web hier ausführliche Informationen über die Arbeiten, Ergebnisse und Technologien in PoliFlow. Das Infonet wurde 1996 eingerichtet und seitdem ständig erweitert und aktualisiert. Es wird von vielen anderen Workflow-Seiten referenziert (GI, AK Workflow, Konferenzseiten etc.) und enthält Informationen über:

- Das Verbundprojekt PoliFlow  
mit Beschreibungen des Projektes und der Förderinitiative POLIKOM, einer Sammlung aller offiziell verfügbaren Projektinformationen und mit Verweisen auf die Projektpartnern, Nachbarprojekte, BMBF etc.
- Die Arbeiten am IPVR  
mit detaillierten Beschreibungen der Arbeitsbereiche (Aufgabenstellung, Ergebnisse etc.) und Verweisen auf alle am Projekt beteiligten Personen.
- Studentische Arbeiten in PoliFlow  
mit Stellenausschreibungen, Übersichten über geleistete Tätigkeiten als wissenschaftliche Hilfskräfte sowie Studien- und Diplomarbeiten im Umfeld von PoliFlow.
- Publikationen und Dokumente  
mit Zusammenfassungen und Volltexten von Veröffentlichungen in PoliFlow, Studien- und Diplomarbeiten sowie Verweisen auf interne Berichte, weitere Veröffentlichungen etc.
- Präsentationen in PoliFlow  
mit fast 20 Hinweisen auf die wichtigsten Präsentationen des IPVR in PoliFlow. Diese beinhalten Informationen zu den Veranstaltungen oder Verweise auf die Veranstaltungsseiten, Ver-

weise auf die Exponate mit Screenshots, Filmen oder Demos sowie Verweise auf die entsprechenden Publikationen oder Vortragspräsentationen (-folien).

- Das System SWATS mit Hintergrundinformationen über Anwendungsfeld, Anforderungen und Motivation sowie ausführlichen Beschreibungen zur Architektur und zu der Benutzerschnittstelle von SWATS.
- Workflow World, eine international angesehene Sammlung von wichtigen Informationen zum Thema Workflow. Die Seiten enthalten Hinweise auf alle wesentlichen Organisationen, auf 40 Projekte/Institutionen und über 60 Produkten/Firmen sowie die wichtigsten Veranstaltungshinweise und Workflow-Literaturverzeichnisse.

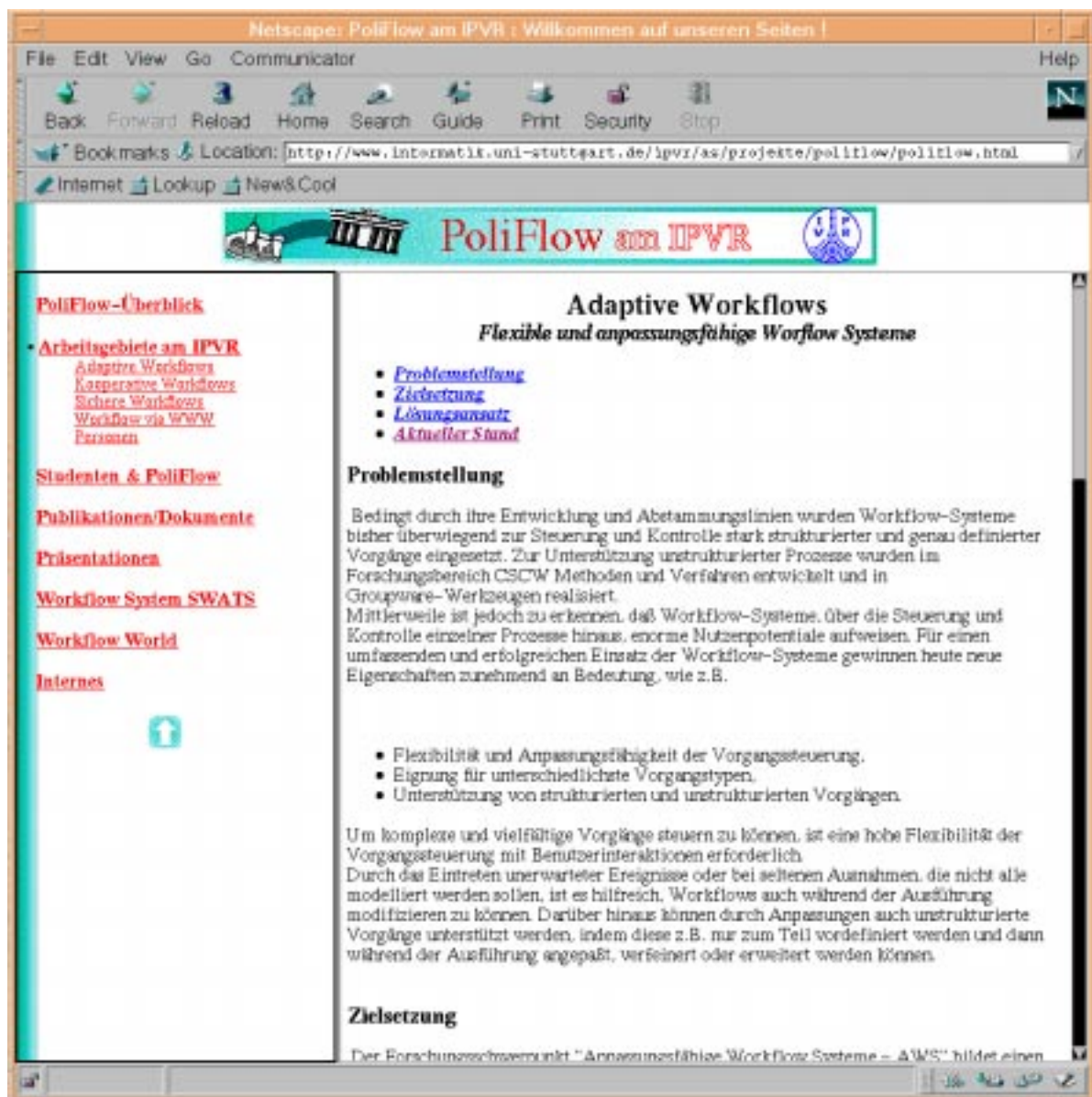


Abbildung 2.1: Das PoliFlow-Infonet am IPVR

## 2.3 Organisation und Kennzahlen

### 2.3.1 Studentische Arbeiten

Im Rahmen von PoliFlow wurden von den Mitarbeitern eine Vielzahl von Studien- und Diplomarbeiten betreut, deren Ergebnisse in die Projektarbeit mit eingeflossen sind. Die folgende Liste gibt einen Überblick über einige relevante Arbeiten aus den Themenbereichen AWS, SWS, MWM und WR:

- Schreyjak S., Siebert R. (DA) : Anforderungsanalyse für Workflow-Management-Systeme
- Bachner E., Kindler T. (SA) : Entwurf einer graphischen Oberfläche zur Unterstützung innovativer Wege interaktiver Benutzerführung in einem Client-Server basierten Kongress-Anmeldesystem
- Kogel E., Kindler T., Soyez T. (SA) : Entwurf und Implementierung eines Monitoring-Werkzeugs für ein verteiltes Workflow-System
- Köcher M., Kindler T., Soyez T. (SA) : Integration von Telekooperationswerkzeugen in ein Workflow-Management-System
- Ott A., Siebert R. (SA) : Modellierung von Anpassungsrechten für Workflows
- Bachner E., Kindler T. (DA) : Entwurf und Realisierung einer sicheren Kommunikationslösung im heterogenen Umfeld der POLIKOM Projekte
- Muth G., Kindler T., Soyez T. (DA) : Entwurf und Implementierung eines User-Managers für eine weit verteilte Workflow-Umgebung
- Pattermann F., Kindler T., Soyez T. (DA) : Entwurf und Implementierung eines Login-Managers für eine weitverteilte Workflow- und Telekooperations-Umgebung
- Aldinger J., Siebert R. (DA) : Erstellung eines Workflow-Editors
- Schröder R., Siebert R. (DA) : Realisierung von Diensten zur Anpassung von Workflows während der Laufzeit
- Vogel A., Kindler T. (DA) : Entwurf und Realisierung einer WWW-Anbindung für Datenbanken zur Gebäudeverwaltung
- Kußmaul T., Kindler T. (DA) : Konzeption und Realisierung eines verteilten WLH
- Kluger R., Kindler T. (DA) : Entwurf und Realisierung einer generischen WUI Toolbox
- Bayer H., Siebert R. (DA) : Realisierung eines Kontrolldienstes für Workflow-Anpassungen
- Hummel M., Siebert R., Kulendik O. (SA) : Erstellung eines netzbasierten Gruppenkalenders
- Hofmann R., Siebert R. (DA) : Entwurf und Realisierung eines anpassungsfähigen Workflow-Modells für SWATS

- Blum E., Siebert R. (DA) : Reimplementierung eines Workflow-Editors für das Internet
- Herzberger T., Siebert R. (SA) : Integration und Erweiterung eines User-Managers in SWATS
- Dudler A., Siebert R. (SA) : Realisierung adaptiver Workflow-Konzepte im Anwendungsfeld von PoliFlow
- Horster O., Kulendik O. (SA) : Dokumentenmanagement in einem WWW-basierten Workflowsystem
- Pataric Z., Kindler T. (DA) : Erweiterung eines Workflow-Management-Systems um Sicherheitsaspekte bei Modellierung und Ausführung von Vorgängen
- Dusel A., Kindler T. (DA) : Entwicklung eines Sicherheitsmodells für Vorgangssteuerungssysteme
- Spang M., Kindler T. (DA) : Entwurf einer Architektur zur Gewährleistung von Sicherheit in Vorgangssteuerungssystemen
- Herzberger T., Siebert R. (DA) : Konsistenz und Integrität in Workflows als Kontrollmechanismen dynamischer Änderungen.

### **2.3.2 Mitarbeiter/innen**

Im Laufe des Projektes haben folgende Personen auf der Seite des IPVR am Projekt - temporär oder dauerhaft - mitgearbeitet:

#### **Management**

Prof. Dr. Ing. Dr. hc. Andreas Reuter

Prof. Dr. rer. nat. Kurt Rothermel

#### **Verwaltung und Sekretariat**

Dr. Ing. Dipl.-Inform. Gerhard Strommer

Stephanie Klöpf

Dipl.-Inform. Matthias Matthiesen

Manfred Rasch

#### **Projektleiter und Mitarbeiter/innen (kontinuierlich 4-5)**

Dipl.-Inform. Steffen Fischer

Dipl.-Inform. Friedemann Schwenkreis

Dipl.-Inform. Thomas Kindler

Dipl.-Inform. Kerstin Schneider

Dipl.-Inform. Ernö Kovacs

Dipl.-Inform. Reiner Siebert

Dipl.-Inform. Ottokar Kulendik

Dipl.-Inform. (FH) Werner Sinz

Dipl.-Inform. Alexander Leonhardi

Dipl.-Inform. Tobias Soyez

Dipl.-Inform. Nikolaos Radouniklis



**Wissenschaftliche Hilfskräfte (kontinuierlich 4-5)**

Thomas Albrecht	Gerhard Muth
Heinrich Brunhmeier	Daniela Nicklas
Christian Bühler	Roland Ortloff
Oliver Frank	Christoph Pfisterer
Otto Dämpfle	Bernhard Riedhofer
Gerald Grau	Ralf Schröder
Armin Grieshaber	Dieter Schüle
Michael Grollmuss	Johannes Schützner
Lars Holowko	Georgios Skempes
Rainer Hofmann	Alexander Till
Zhengyu Jiang	Georgios Tzizlis
Jens Müsle	Man Yang

**2.3.3 Externe Kontakte**

Zu den folgenden Firmen, Forschungseinrichtungen oder Organisationen wurden im Rahmen von PoliFlow Kontakte oder Kooperationen unterhalten:

- Workflow Management Coalition (WFMC): aktive Mitgliedschaft.
- Gesellschaft für Informatik (GI), Arbeitskreis Workflow: aktive Beteiligung
- Software Ley GmbH (heute COSA Solutions GmbH): Produkt-Evaluation
- CSE Österreich: Produkt-Evaluation
- Hewlett-Packard Labs, Palo Alto, USA: Kooperation und fachlicher Austausch
- debis Systemhaus: Kooperation und fachlicher Austausch
- BERKOM - Beta Test: Multi Media Services
- Universität Münster: Kooperation und fachlicher Austausch (AWS)
- Universität Oldenburg: Fachlicher Austausch (AWS)
- Universität Erlangen: Fachlicher Austausch (AWS)
- Universität Ulm: Fachlicher Austausch (AWS)

### 3 Realisierung eines WFMS (WR)

Workflow-Systeme bieten Vorteile wie Produktivitätssteigerung durch Vermeidung von Transport- und Liegezeiten und durch Parallelarbeit, Erhöhung der Auskunftsfähigkeit durch Verfügbarkeit des Workflow-Zustands, Qualitätssicherung durch das Verhindern vergessener Aktivitäten und die Dokumentation ausgelassener Aktivitäten, und Nachweisbarkeit durch die Dokumentation von Abläufen aus rechtlichen Gründen oder wegen Normen [Böhm95]. Es sollte bestimmt werden, ob und wie man diese Vorteile für das Anwendungsfeld des Projekts, in dem die Behörde OFD operiert, nutzbar machen könnte.

Die Untersuchung des Anwendungsfelds in der ersten Phase des Projekts ergab, daß es unter anderem durch eine räumliche Verteilung der Behörde, durch eine heterogene IT-Infrastruktur und durch Benutzer mit unterschiedlichen IT-Kenntnissen und Vorgangsbeteiligungen gekennzeichnet ist. Für ein solches Anwendungsfeld waren die existierenden Standard-Workflow-Systeme zunächst ungeeignet, da sie nur im lokalen Netzwerk verfügbar waren, beim Benutzer lokal installierte Software für eine bestimmte Plattform voraussetzten und den Benutzer oftmals mit komplexen Benutzungsschnittstellen überforderten. Damit würden sie eine Einführung von Workflow-Management in einem solchen Umfeld unnötig erschweren, da der Einsatzbereich unnötig eingeschränkt, der Aufwand für die Installation und den Betrieb des Systems und für die Schulung der Mitarbeiter überflüssig groß wäre.

Das Ziel des Arbeitsbereichs 'Workflow Realisierung' war somit die Konzipierung einer Architektur für ein integriertes Workflow-Management-System, das Erleichterungen bei der Einführung bieten könnte, und mit dem bestehende Verwaltungsvorgänge im eben dargestellten Problemfeld bei der Anwenderbehörde bearbeitet werden könnten. Die Architektur sollte mit einem Prototyp, möglichst unter Verwendung kommerziell verfügbarer Technologie, realisiert werden, um dieses Konzept zu verifizieren und zu demonstrieren. Ein pragmatisches Teilziel war dabei außerdem die Bereitstellung eines Basissystems zur Integration der innovativen Konzepte aus den PoliFlow-Arbeitsbereichen des IPVR AWS, MWM und SWS.

Im folgenden werden zunächst die Grundlagen von Internet-Technologien und ein verwendetes kommerzielles Workflow-System dargestellt. Danach wird die Problemstellung umrissen und Anforderungen und Entwurfsziele für die Lösungsarchitektur 'Stuttgarter Workflow And Telecooperation System', abgekürzt SWATS, abgeleitet, zu der anschließend Konzeption und Aufbau präsentiert werden. Anschließend wird der realisierte Prototyp mit einigen Komponenten und Beispielanwendungen beschrieben. Eine Zusammenfassung beschließt den Bericht über diesen Arbeitsbereich.

## 3.1 Grundlagen

### 3.1.1 WWW- / Internet-Technologien

Das Internet besteht vereinfacht aus miteinander verbundenen Rechnern, die alle die Protokolle TCP/IP unterstützen. Auf diesen Protokollen setzen dann Anwendungen wie das World Wide Web (WWW), e-Mail oder News auf. Diese Anwendungen sind meist in Client/Server-Architektur realisiert: Der Benutzer der Anwendung interagiert mit einer Benutzungsschnittstelle, die auf seinem Netzwerk-Endgerät abläuft, während der Server auf einem anderen Rechner im Netzwerk die Anwendungslogik und Datenhaltung der Anwendung realisiert. Durch die standardisierte Kommunikation lassen sich die verschiedenen heterogenen lokalen Netzwerke wie Ethernet, ATM oder Tokenring gemeinsam und transparent, das heißt, unabhängig von ihren technologischen Eigenheiten, übergreifend verwenden. Diese Technologie kann dazu verwendet werden, ein behördeninternes Netz (Intranet), das nach außen hin abgeschirmt ist, oder ein Netzwerk (Extranet) für bestimmte Partnerrechner bzw. Partnernetze zu realisieren.

#### **HTML, HTTP, WWW-Server, WWW-Browser, Formulare, CGI**

Die Basistechnologie des WWW bildet die Seitenbeschreibungssprache Hypertext Markup Language HTML [Musc96], mit dessen Standardisierung und Weiterentwicklung sich das World Wide Web Consortium [WWWC] beschäftigt. Mit ihr lassen sich über das Hypertext Transfer Protokoll (HTTP) [IETF96], das wie die anderen Internet-Standards von der Internet Engineering Task Force [IETF] standardisiert wird, der Inhalt und teilweise auch das Layout von Informationsseiten beschreiben, die Text, Bilder, Tabellen, Verknüpfungen zu anderen Seiten (sogenannte Hyperlinks) und vieles mehr enthalten.

WWW-Server wie Apache von der Apache Group [Apache], dessen Implementierung frei verfügbar ist, gestatten den Netzwerkzugriff auf diese Seiten, indem sie HTTP implementieren. Sie liefern auf die Anfragen eines WWW-Browsers dem Client, mit dem man durch die Seiten navigieren kann, die bei ihnen lokal gespeicherten Seiten zurück. Zu den bekanntesten und gebräuchlichsten Browsern gehören Communicator von Netscape und Internet Explorer von Microsoft.

Um eine gewisse Interaktion mit dem Benutzer des WWW zu erhalten, kann man in HTML Formulare gestalten. Solche Formulare gestatten dem Benutzer Eingaben in Form von Menüs, Texteingabefeldern und dem Betätigen von Schaltflächen. Diese werden zur Verarbeitung an ein dem Formular zugeordnetem Programm auf dem WWW-Server zurückgesendet, das vom WWW-Server ausgeführt wird und die Eingaben verarbeitet. Das Programm, ein sogenanntes CGI-Skript, kann weitere Aktionen vornehmen, wie etwa Manipulationen von Datenbanken als Resultat der

Eingabe oder ähnliches, und generiert schließlich wieder eine Ausgabeseite in HTML, die an den Browser zurückgesendet und damit dem Benutzer als Resultat präsentiert wird. Diese Art der Verarbeitung von Benutzereingaben ist allerdings nicht besonders interaktiv in dem Sinne, daß die Reaktion des Rechners auf Eingaben immer erst nach einem Zugriff auf den Server erfolgt.

### **Java, Network Computing, JDBC**

Für weiterreichende Interaktionen mit dem Benutzer, die mehr Logik im Client benötigen, um schneller auf Benutzereingaben reagieren zu können, kann die objektorientierte Programmiersprache Java [Flan97] von Sun Microsystems [Java] verwendet werden. Java ist ein vom Hersteller Sun veröffentlichter de facto-Standard. Durch die Verfügbarkeit von Java-Interpretern auf zahlreichen Plattformen, wie insbesondere UNIX-Derivaten und Windows 95/NT, kann man bereits heute von einer plattformunabhängigen Programmiersprache sprechen. Sun ist darüberhinaus bestrebt, Java beim Standardisierungsgremium ISO als PAS (Public Accessible Specification, zu deutsch: öffentlich verfügbare Spezifikation) einzubringen, was Java zu einem internationalen de jure-Standard machen würde.

Durch die Plattformunabhängigkeit von Java sind Java-Programme nicht nur portabel, sondern sogar in übersetzter Form unverändert unmittelbar auf anderen Plattformen ablauffähig, sofern keine plattformspezifischen Bibliotheken oder nur auf einem Rechner lokal zugreifbaren Ressourcen verwendet werden.

Java gestattet die Implementierung von Programmen, die über das Netzwerk durch HTTP zum Client transportiert werden können und dort in einem WWW-Browser ablaufen, die sogenannten Applets. Dies ermöglicht das sogenannte Network Computing: Anwendungen müssen nicht auf Client-Systemen installiert werden, sondern werden bei Bedarf von Servern geladen, auf denen die Software zentral gespeichert und administriert wird, und sind damit prinzipiell auf allen Knoten des Netzwerks verwendbar. Durch Caching auf dem Client, also dem temporären Speichern bereits zugegriffener Anwendungen, entfallen Ladevorgänge bei wiederholt benötigten Anwendungen, die Aktualität der Anwendungen bei Versionswechseln bleibt gewahrt. Da somit die benötigte Software beim Client sich prinzipiell auf einen WWW-Browser beschränkt, wird dieser Ansatz auch als 'Thin Client'-Ansatz bezeichnet, da er nur geringe Ansprüche an die Hardware des Clients stellt.

Die Definition und Standardisierung weiterer Schnittstellen vergrößert das Einsatzgebiet von Java und seine Möglichkeiten zur Integration von Hardware und Software. So bietet beispielsweise die Schnittstelle JDBC (Java DataBase Connectivity) den Zugriff auf Datenbanken, wenn diese einen JDBC-Treiber bereitstellen [Java97]. Die standardisierte Schnittstelle erleichtert dabei den Austausch der Datenbank-Realisierung. Weitere wichtige Schnittstellen betreffen e-Mail, Smart

Cards und viele mehr. Dadurch, und durch seine Einfachheit im Vergleich zu Sprachen wie C++, nimmt die Verbreitung von Java ständig zu. Weiterhin verfügt Java über Sicherheitskonzepte, um die Privilegien der Applets beim Ablauf auf dem Client einschränken.

Java wird gewöhnlich interpretiert verwendet: Zur Laufzeit wird der aus dem Quellcode generierte Zwischencode vom Interpreter übersetzt und ausgeführt. Es gibt allerdings Just-In-Time-Compiler, die direkt vor der Ausführung den ganzen Zwischencode der Klassen in den Code der zugrundeliegenden Maschine übersetzen.

## **CORBA**

CORBA ist ein Standard der Object Management Group (OMG) [OMG], einem Herstellerkonsortium, das Standards für Software erarbeitet und veröffentlicht, für die kommerzielle Implementierungen verfügbar sind. Das Ziel der OMG ist dabei, die Interoperabilität von Software zu erreichen, das bedeutet, das Zusammenspiel von Software, die in unterschiedlichen Programmiersprachen und auf unterschiedlichen Plattformen entwickelt wurde.

Die Common Request Broker Architecture (CORBA) standardisiert objektorientierte Client-Server-Kommunikation auf dem Internet. Implementierungen dieser Architektur mit dem Kernstück ORB (Object Request Broker), die mittlerweile zahlreich kommerziell und frei verfügbar sind, ermöglichen es, als objektorientierte Komponenten realisierte verteilte Anwendungen über ein Netzwerk wie das Internet zugreifbar zu machen.

Der besondere Vorteil von CORBA gegenüber anderen sogenannten Middleware-Systemen wie etwa Java RMI ist, daß diese Komponenten in unterschiedlichen Programmiersprachen entwickelt sein können, wie etwa C++, Java, Ada und weitere. Für jede dieser Sprachen wurde eine Abbildung auf die Schnittstellenbeschreibungssprache (IDL) von CORBA definiert. Die Objektdienste (object services) wie beispielsweise Namensdienst, Sicherheitsdienst uvm., die von der OMG standardisiert werden, ergänzen die Kommunikationsfunktionalität des ORB und ermöglichen die Erweiterung eines Systems um Dinge wie Objektamen, Sicherheit und so weiter.

Außerdem gestattet CORBA, Altsoftware durch das Wrapper-Konzept mit einer Softwarehülle zu umgeben und so in ein erweiterbares System einzubinden. Da CORBA ein offener Standard ist, ist ferner eine gewisse Unabhängigkeit von den Herstellern der ORBs gewährleistet, deren Interoperabilität untereinander durch das sogenannte Internet Inter-ORB Protokoll (IIOP) ebenfalls standardisiert ist. (So interoperieren im SWATS-Prototyp beispielsweise ein ORB von Iona und ein ORB von Hewlett Packard.)

## HP Changengine

Changengine ist eine Workflow-Engine von Hewlett Packard [HPa][HPb], also eine Kernkomponente eines Workflow-Management-Systems, in der Workflow-Definitionen und -Instanzen verwaltet und bearbeitet werden. Nachdem eine Evaluierung von Workflow-Systemen ergab, daß die Architektur und die Technologien von Changengine genau in die Konzeption von SWATS paßte, wurde Changengine für die Verwendung im SWATS-Prototyp ausgewählt und konnte im Rahmen einer Forschungskoooperation mit den HP Labs, Palo Alto, in PoliFlow verwendet werden.

Hinweis: Die in diesem Bericht verwendeten Beschreibungen zu Changengine beziehen sich auf die zum Zeitpunkt der Implementierung des Prototyps verwendeten Versionen von Changengine [HP97a][HP97b][HP97c] (bis zur Version A.01.00.03), vormals mit dem Namen Montana bezeichnet. In der Zwischenzeit wurde das Produkt erweitert, auch durch die Resultate des Projekts. Siehe dazu [HPb].

Changengine besitzt einen modularen Aufbau und verwendet zur internen Kommunikation der Komponenten einen CORBA-basierten Nachrichten-Bus. Diese Architektur erleichtert die Erweiterung des Systems um weitere Komponenten, ohne daß Eingriffe in bestehende, mitgelieferte Komponenten nötig werden. Die Verwendung von CORBA ermöglicht den Betrieb in einer heterogenen Hardware-Umgebung. Changengine sorgt weiterhin für eine persistente Speicherung der Workflow-Definitionen und -Instanzen.

Changengine ermöglicht die Definition von Workflows. Die Kontrollstruktur, das heißt, die Beschreibung der möglichen Abläufe eines Workflows, wird durch einen gerichteten Graph gegeben. Knoten des Graphen können Arbeitsknoten sein, die auf Aktivitäten verweisen, oder Verzweigungsknoten, die Regeln enthalten, die den Ablauf des Workflows beeinflussen. Ereignisse können in Regelknoten erzeugt und behandelt werden. In einer Workflow-Definition können weiterhin Kontrolldaten deklariert und definiert werden, auf die in den Aktivitäten und in den Regelknoten zugegriffen werden kann. Aktivitäten schließlich enthalten eine Rollendefinition, die den oder die möglichen Bearbeiter spezifiziert. Weitere technische Aspekte der Workflow-Definition betreffen die Version des Workflows und weiteres.

Changengine gestattet die Einbindung selbstimplementierter Workflow-Komponenten. So kann die Rollenbeschreibung einer Aktivität durch einen beliebigen, selbstimplementierte Rollenverteilungskomponente, einen sogenannte Ressourcenmanager, aufgelöst werden. Die Aktivitäten werden von der Kernkomponente, der Workflow-Engine, an sogenannte Business-Objekte geschickt, die ebenfalls beliebig spezifiziert werden können, was wiederum eine Verteilung der Aktivitäten an selbstimplementierte, eventuell über das Netzwerk verteilte, Arbeitslisten-Manager oder an verarbeitende Agenten ermöglicht.

## 3.2 Problemstellung und Anforderungen

### 3.2.1 Problemstellung

Ausgangspunkt des Arbeitsbereichs "Workflow Realisierung" war das Problem, die Vorgangsbearbeitung in Behörden durch Workflow-Management zu unterstützen. In der ersten Phase des Projekts zeigten sich dabei die folgenden Charakteristika des Umfelds, die die Einführung von Workflow-Technologie beeinflussen.

Behörden sind räumlich über mehrere Standorte verteilt, wie die OFD, speziell im Kontext des Regierungsumzugs. Außerdem kooperieren sie mit weiteren Behörden und anderen externen Partnern: So sind bei der OFD beispielsweise Architekturbüros an den HU-Bau-Vorgängen beteiligt. Dies und das evolutionäre Wachstum der Informationstechnologie - in Behörden wie in anderen Organisationen - bedeutet, daß Teilnehmer der zu unterstützenden Vorgänge insgesamt keine homogene, sondern eine heterogene IT-Infrastruktur haben: Sie besitzen verschiedene Hardware- und Software-Plattformen, wie etwa Windows-PCs für einfache Textverarbeitung und UNIX-Workstations für CAD-Aktivitäten, wie sie häufig in Architekturbüros Verwendung finden.

An den zu unterstützenden Vorgängen nehmen unterschiedliche Arten von Benutzern teil: Es gibt zum einen die routinierten, EDV-erfahrenen Sachbearbeiter, die bestimmte Vorgänge mit häufiger Wiederkehr bearbeiten, und die primär durch bisherige Workflow-Management-Systeme unterstützt werden. Allerdings gibt es auch Benutzer, die zum Beispiel wie die Kooperationspartner nur gelegentlich an Vorgängen der Behörde teilnehmen und deshalb möglichst nicht mit der Standard-Schnittstelle eines Workflow-Systems ausgerüstet und konfrontiert werden sollten. Es gibt sogar EDV-unerfahrene Benutzer, die mit dem Konzept und der mächtigen Standard-Benutzerschnittstelle eines Workflow-Management-Systems und mächtigen Anwendungen zur Bearbeitung von Teilschritten zunächst überfordert wären. Die Einführung von Workflow-Management in einem solchen Umfeld muß diese Faktoren berücksichtigen, um nicht durch unnötige Kosten für die Schulung der Teilnehmer und für die Entwicklung, Installation und Wartung nötiger Hardware und Software das Rationalisierungspotential, das mit Workflow-Management erreichbar ist, zu schmälern oder gar langfristig zu verhindern.

### 3.2.2 Anforderungen und Entwurfsziele

Aus den beobachteten Problemen lassen sich folgende Anforderungen an eine Architektur für ein Workflow-Management-System in diesem Kontext ableiten:

1. Verteilter Zugriff auf das Workflow-System

Um den behördenweiten Einsatz und die Einbindung von Partnern in die Vorgänge sicherzu-

stellen, soll das Workflow-System dezentral über das Netzwerk zugreifbar sein. Das heißt, es soll netzwerkweit möglich sein, Vorgänge zu starten, ihren Zustand zu verfolgen, sich Aktivitäten anzeigen und sie bearbeiten zu lassen.

## 2. Unterstützung heterogener IT-Infrastruktur

Um bestehende Netzwerke, Hardware und Software möglichst weiterverwenden zu können, soll die Architektur unterschiedliche lokale Netzwerke, wie Ethernet, Tokenring und ähnliches, und die unterschiedlichen Rechner-Plattformen, wie Windows-PCs und Unix-Workstations, einbinden können.

## 3. Einfacher, mehrgestaltiger Zugang zum Workflow-System

Um auch gelegentliche Workflow-Benutzer und unerfahrene Benutzer leicht an Vorgängen teilnehmen zu lassen, die Einarbeitungszeit kurz zu halten und eine gute Benutzerakzeptanz zu erreichen, soll die Benutzungsschnittstelle einfach gestaltet und leicht ersetzbar sein. Außerdem soll es unterschiedliche Formen von Zugängen zum Workflow-System geben, um etwa die nur an einzelnen Aktivitäten von Vorgängen Beteiligten nicht der gesamten Komplexität des Workflow-Managements auszusetzen.

## 4. Offene und erweiterbare Architektur

Die Verwendung veröffentlichter Standards bei der Realisierungstechnologie soll die Abhängigkeit der Anwenderbehörde von einzelnen Herstellern minimieren und die Zukunftssicherheit der Architektur erhöhen. Die Architektur soll durch geeignete Auswahl von Technologien ferner die Einbindung bestehender Informationssysteme und Altsoftware erleichtern. Dieses Entwurfsziel sollte weiterhin die Erreichung des pragmatischen Teilziels der Einbindung der Ergebnisse aus AWS, MWM und SWS sicherstellen.

# 3.3 Konzepte und Lösungsansätze

## 3.3.1 Stuttgarter Workflow And Telecooperation System (SWATS)

### Konzeption der Architektur

Die Konzipierung der Lösungsarchitektur zur eben beschriebenen Problemstellung bestand im wesentlichen in der Wahl der Technologien und der Wahl eines Kern-Workflow-Managementsystems, das dann für die Projektzwecke erweitert werden sollte.



Zur Realisierung des dezentralen Zugriffs bei gleichzeitiger Wahrung der Konsistenz der Datenbestände wurde das System in Client-Server-Architektur entworfen. Dabei enthalten die Client-seitigen Komponenten Präsentationslogik und Teile der Anwendungslogik, der Großteil der Anwendungslogik und die Datenspeicherung erfolgt auf der Server-Seite.

Als gemeinsame Basis-Infrastruktur für die heterogene und räumlich verteilte IT-Landschaft und zur Überwindung der Netzwerk-Heterogenität wurden die Internet-Technologien gewählt; dabei ist eine Nutzung dieser Technologien in einem Intranet, Extranet oder im Internet möglich. Das WWW bietet darüberhinaus einen einfachen und dezentralen Zugang zum Workflow-System: Das einfache Konzept von HTML-Seiten und -Formularen und die einfache Bedienung von WWW-Browsern und deren niedrige Kosten versprechen eine Herabsetzung der Zugangsschwelle zum System. Außerdem sind die WWW-Browser Integrationspunkte für die unterschiedlichen Internet-Technologien, wie Email und News.

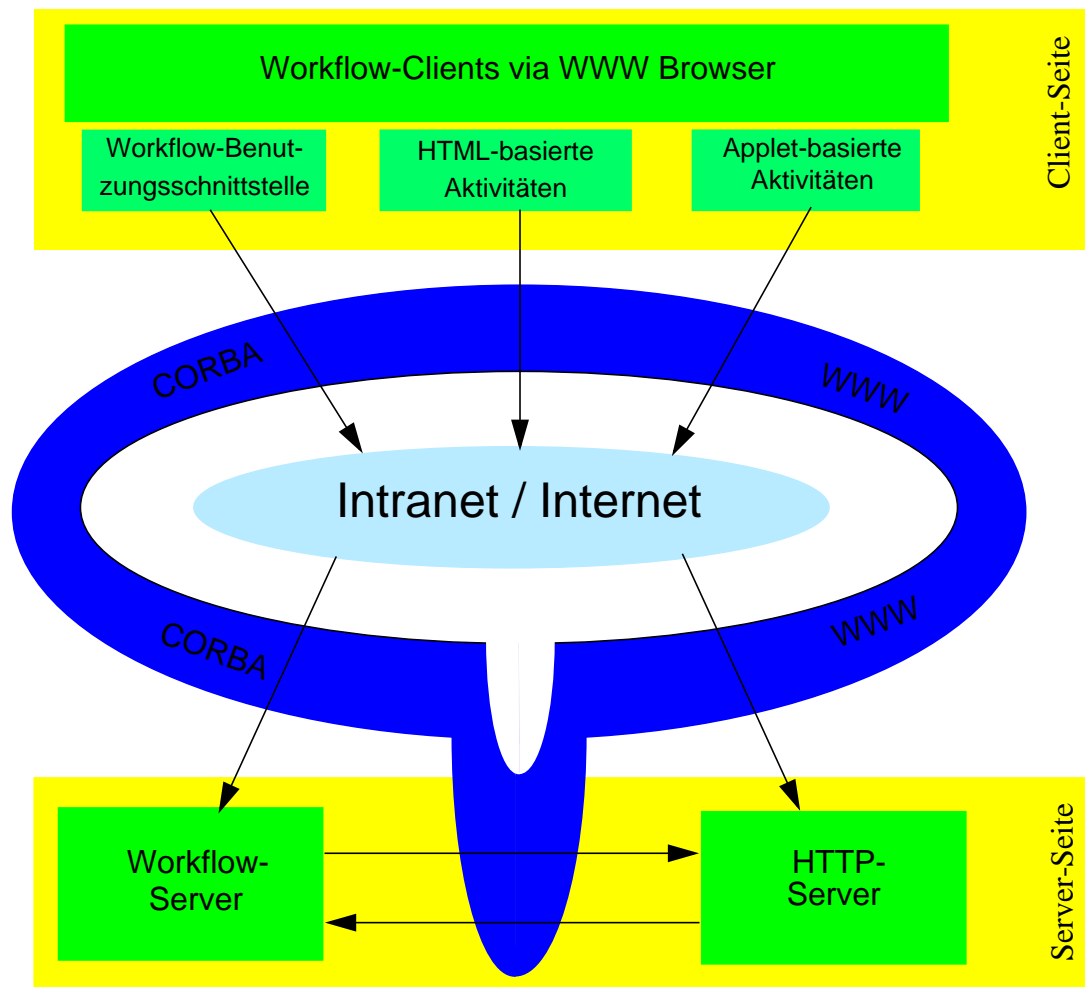
Zur weitergehenden Überwindung der Client-seitigen Heterogenität und zur Implementierung von komplexerer Präsentationslogik wurde die Programmiersprache Java ausgewählt. Da sie ebenfalls in WWW-Browsern einsetzbar ist, gestattet sie die Steuerung des Workflow-Systems und die Arbeit mit Anwendungen an einer Stelle. Außerdem ermöglicht Java die Verwendung des Network-Computing-Paradigmas. Dadurch genügt die Verfügbarkeit beziehungsweise Installation von WWW-Browsern auf den Client-Systemen zur Steuerung des Workflow-Systems und zur Bearbeitung der Aktivitäten; es muß keine weitere Software dezentral installiert werden.

Zur Überwindung Server-seitiger Heterogenität, die sich in Serverprogrammen äußert, die in unterschiedlichen Programmiersprachen, für unterschiedliche Betriebssysteme und Hardware-Plattformen realisiert sind, wurde CORBA ausgewählt. Es ermöglicht den dezentralen Zugriff auf Workflow- und anwendungsspezifische Server. Außerdem läßt sich CORBA leicht in Java-Programme integrieren.

Mit HP Changengine wurde schließlich ein Workflow-Management-Kern-System des Industriepartners gewählt, das zwar im Hinblick auf einen unmittelbaren Einsatz in einer Behörde unvollständig war, sich aber hervorragend in die Architektur einfügte, da es zum einen eine CORBA-Schnittstelle und zum anderen eine erweiterbare Struktur besitzt.

### **Die Architektur im Überblick**

Im folgenden Bild (vgl. Abb. 3.1) ist die Architektur von SWATS grob dargestellt. Sie zerfällt in die Client-Seite und die Server-Seite, die durch die erwähnten Internet-Technologien verbunden sind.



**Abbildung 3.1:** Architektur von SWATS

### Die Client-Seite

Die Benutzungsschnittstelle wurde in Form eines Java-Applets in eine HTML-Seite integriert. Die Funktionsweise der Benutzungsschnittstelle (vgl. Abb. 3.2) soll hier nur in groben Zügen beschrieben werden. Eine detailliertere Beschreibung davon findet sich in Kapitel über MWM.

Zuerst muß sich der Benutzer beim Workflow-System anmelden, indem er sich mit Name und Paßwort dem System gegenüber identifiziert. Wurde die Verbindung erfolgreich aufgebaut, so kann sich der Benutzer die Aktivitäten anzeigen lassen, die in seinem Eingangskorb und auf seinem Schreibtisch liegen. Weiterhin hat er auch die Möglichkeit, neue Vorgänge, die in einer Vorgangsliste zusammengefaßt sind, zu starten. Die Aktivitäten im Eingangskorb können vom Benutzer zur Bearbeitung übernommen werden. Aktivitäten auf dem Schreibtisch eines Benutzers (vgl. Abb. 3.2) können zur Bearbeitung geöffnet werden.



**Abbildung 3.2:** Benutzungsschnittstelle des Prototyps

Für die Realisierung der Aktivitäten können, wie zur Workflow-Steuerung, Java-Applets verwendet werden. Durch ein speziell entwickeltes Konzept können die zu bearbeitenden Aktivitäten aber auch als HTML-Formulare dargestellt und bearbeitet werden. Dabei können einem Bearbeiter (Aktor) die HTML-Formulare auf zwei verschiedenen Wegen zugeführt werden:

1. Die Aktivitäten werden über das User-Interface ausgewählt, und die entsprechenden HTML-Formulare können in einem eigenen Browser-Fenster dargestellt und bearbeitet werden.
2. Die Aktivitäten werden via e-Mail zugestellt, und die entsprechenden HTML-Formulare werden direkt in einem HTML-fähigen e-Mail-Client dargestellt und bearbeitet.

Die Mitteilung an den User-Manager, daß die Bearbeitung einer Aktivität abgeschlossen wurde, erfolgt in beiden Fällen direkt aus dem HTML-Formular über CGI.

## Die Server-Seite

Die Server-Seite besteht aus folgenden beiden Komponenten:

1. einem verteilten Workflow-Server, der Workflow-Definitionen und -Instanzen, das Organisationsmodell und die Arbeitslisten der Benutzer verwaltet und deren Bearbeitung gestattet. Diese Funktionalität stellt er den anderen Komponenten durch eine objektorientierte CORBA-Schnittstelle zur Verfügung.
2. einem HTTP-Server, der für Workflow-Clients die graphischen Benutzungsschnittstellen und Workflow-Aktivitätsimplementierungen in Form von HTML-Formularen und Java-Applets bereithält. Außerdem dient er zur Vermittlung der CORBA-Objektreferenz des User-Managers, die per HTTP vom Client abgefragt werden.

Der Workflow-Server wiederum besteht im wesentlichen aus drei Teilen:

### 1. Workflow-Engine

Die Workflow-Engine entscheidet aufgrund des aktuellen Zustands eines Vorgangs, welche Aktivitäten als nächstes zu bearbeiten sind. Die Zuweisung der Aktivitäten an die betreffenden Ressourcen erfolgt mit Hilfe des Resource-Managers.

### 2. Ressourcenmanager / Organisationsmanager

Der Ressourcenmanager ist für die Auflösung von Rollen zu Ressourcen zuständig. Durch den Zugriff auf Information über die Aufbauorganisation, die der Organisationsmanager verwaltet, liefert er auf Anfragen der Workflow-Engine, die in der Workflow-Beschreibung spezifiziert werden, und auf Anfragen des User-Managers einen oder mehrere mögliche Bearbeiter für eine Aktivität zurück.

### 3. User-Manager

Diese Komponente dient als Zugang für die Endbenutzer-Clients. Er bietet über CORBA eine reduzierte, spezialisierte Schnittstelle zum Workflow-System und ermöglicht so eine Vereinfachung der Benutzungsschnittstelle und einen erleichterten Austausch der Workflow-Engine. Der User-Manager ist zuständig für die Verwaltung der Arbeitslisten, die Aktivitäten eines Anwenders enthalten. Jeder Anwender kann dabei aus dem User-Interface heraus auf die eigene Arbeitsliste im User-Manager zugreifen und seine Aktivitäten manipulieren, das heißt, sie sich zur Bearbeitung zuweisen lassen, sie abschließen, delegieren und ähnliches.

Die nachfolgende Tabelle gibt Aufschluß über die in SWATS Release 2.3 verwendeten Software-Module. Der Server-Teil dieses beschriebenen Systems war auf einer HP Workstation *HP 9000 / s777* unter *HP-UX 10.20* installiert und betrieben. Als Clients wurden Workstations vom selben Typ, PCs vom Typ *HP Vectra XU 5/90* mit *Windows 95* und *Windows NT* und Unix-Terminals vom Typ *HP EnvizeX a Series* an den Workstations verwendet.

## Verwendete Komponenten und Hardware:

<i>Komponente, Version</i>	<i>Hersteller</i>	<i>Funktion; ggfs. spezielle Verwendung im SWATS-Prototyp</i>
Changengine A.01.00.03	Hewlett Packard	Workflow-Management; Verwaltung und Manipulation von Workflow-Instanzen
OrbPlus	Hewlett Packard	C++-ORB; Integration von Changengine
Oracle 7 Server 7.3.2.2	Oracle	Datenbank; Datenspeicherung für Workflow-Management, Workflows und Anwendungen
JDBCThin	Oracle	vollständig in Java implementierter JDBC-Treiber; clientseitig und serverseitig verwendet
Apache 1.2b4	Apache Group	WWW-Server; Verwaltung von Informationsseiten, Server-Informationen, Client-Software, Formularbearbeitung
Swing 0.5.1	Sun	Oberflächenbibliothek; Verwendung in Client-Software
OrbixWeb 3.0	Iona	Java-ORB; Verwendung in Client-Software und serverseitig in WFA
JDK 1.1.6 B.01.13.04	Hewlett Packard	Java-Interpreter für HP-UX Verwendung in WFA
WFA 1.3	IPVR	HTML-Formularverarbeitung; Server-seitige Formularverarbeitung
Communicator 4.06	Netscape	WWW-Browser für Unix und Windows95/NT; generischer Zugriff auf Workflow-System, Anwendungen und Informationen
WUI 2.3	IPVR	Benutzungsschnittstelle; Bearbeitung von Arbeitslisten und Aktivitäten
WUM 2.3	IPVR	Verwaltung und Manipulation von Arbeitslisten etc.
WRM 1.3	IPVR	Integration des WOM in die Changengine-Architektur
WOM 1.2	IPVR	Verwaltung und Auskunft über Organisation und Rollen- auflösung
WSS 1.2	IPVR	Start der Komponenten des Prototyps

**Tabelle 3.1:** Verwendete Komponenten in SWATS Prototyp Release 2.3

### 3.3.2 Ausgewählte eigenimplementierte Komponenten der Architektur

#### Der Workflow User Manager (WUM)

Der Entwurf des WUM zielt auf die Realisierung der Abstraktionen Arbeitssitzung, Prozeßdefinitionsliste, Prozeß, Arbeitsliste und Aktivität, die verwaltet werden und vom Benutzer manipuliert werden können. Wichtig ist dabei der Dienst der Arbeitslistenverwaltung, die Realisierung eines Zugriffsschutzes auf die Aktivitäten und die Authentisierung der Benutzer. Eine Arbeitsliste ist mit einem Benutzer assoziiert und enthält alle Aktivitäten, die diesem Benutzer zur Bearbeitung zugewiesen sind. Zugriffsschutz wird vom WUM in dem Sinne realisiert, daß er Teilnehmern lediglich den Zugriff auf solche Aktivitäten gestattet, zu deren Bearbeitung sie autorisiert sind. Dazu führt der WUM ebenfalls die Authentisierung seines Benutzers mittels eines Passworts durch. Insgesamt orientiert sich der Entwurf an der von der WfMC standardisierten Schnittstelle 2 [WfMC97] für Workflow-Client-Applikationen, diese ist aber objektorientiert umgesetzt [Muth96][Kußm97].

Die Realisierung des WUM implementiert Schnittstellen für Arbeitssitzungen, Prozeßdefinitionslisten, Prozesse, Arbeitslisten und Aktivitäten. Sie ist dabei als Business-Objekt in die Change-Engine-Architektur eingebunden, das mit der Workflow-Engine kommuniziert. Die einzelnen Instanzen der Arbeitslisten, Aktivitäten et cetera sind eigenständige CORBA-Objekte, was nach deren Auffindung den direkten Zugriff auf sie ermöglicht, ohne den Umweg über ein zentrales Objekt. Der WUM verwendet den WOM, um Rollenbeschreibungen aufzulösen und mögliche Bearbeiter zu bestimmen. Außerdem übernimmt er das Anlegen von Formularen beim Starten eines Workflows, indem er von Formularvorlagen Instanzen für die entsprechende Workflow-Instanz anlegt.

#### Der Workflow Organisation Manager (WOM)

Bei der Verteilung von Arbeitseinheiten in einem Workflow-Management-System werden zumeist Rollen oder Gruppen definiert, die für die Ausübung bestimmter Tätigkeiten geeignet sind (zum Beispiel die Rolle Sachbearbeiter Wohnungsvergabe). Personen oder künstliche Aktoren (Agenten) gehören diesen Rollen oder Gruppen an und stellen den konkreten Bearbeiter dar (beispielsweise spielen die Personen Maier und Müller die Rolle Sachbearbeiter Wohnungsvergabe). Oft sind auch Beziehungen zwischen den Personen bei bestimmten Aufgaben relevant, etwa wenn ein Urlaubsantrag vom Vorgesetzten zu genehmigen ist.

Größtenteils lassen sich diese Informationen aus der Aufbauorganisation gewinnen. Dazu muß die Aufbauorganisation aber zunächst in computerlesbarer Form gespeichert sein, und es müssen Schnittstellen zur Abfrage existieren. Viele kommerzielle Workflow-Management-Systeme lösen dieses Problem, indem sie ein bestimmtes Metamodell für die Aufbauorganisation vorschreiben und bieten dementsprechend nur vordefinierte Abfragen an. In Metamodellen kommen zum Beispiel Personen, Stellen, Rollen, Gruppen und Organisationseinheiten vor. Auch die Beziehungen zwischen diesen sind in Metamodellen festgelegt.

Bei der Untersuchung existierender Metamodelle stellte sich heraus, daß von speziellen angepaßten Metamodellen der größte Nutzen zu erwarten ist [Wege97][Mühl96][Gall95]. Aus diesem Grund wurde eine Organisationskomponente geschaffen, der jedes beliebige Metamodell unterlegt werden kann. Die Schnittstelle zu dieser Organisationskomponente (WOM, Workflow Organisations Manager) besteht aus einer Abfragesprache, die nicht an ein spezielles Metamodell gebunden ist. Zusätzlich zu den Daten der Aufbauorganisation lassen sich in dieser Sprache auch ablauforganisatorische Abfragen formulieren. Diese Fähigkeit benötigt man etwa, wenn eine Arbeitseinheit von derselben Person ausgeführt werden soll, die auch bereits eine vorangegangene Tätigkeit ausgeführt hat. Damit kann man sicherstellen, daß eine Person für einen ganzen Arbeitsablauf zuständig ist, obwohl dieser in verschiedene Arbeitseinheiten aufgeteilt ist.

Der WOM ist als Server realisiert. Er arbeitet mit einer relationalen Datenbank, die das Metamodell und die Daten der Aufbauorganisation enthält. Der Abfragedienst des WOM wird über CORBA angesprochen. Für die Abfragen wurde eine SQL-ähnliche Sprache entworfen. Die Resource Query Language (RQL) genannte Abfragesprache ermöglicht es, auf einfache Art und Weise Anfragen entsprechend dem Metamodell zu formulieren. Bei entsprechender Modellierung der Aufbauorganisation ließe sich beispielsweise eine Anfrage stellen wie: "Wer ist der Chef von Herrn Mueller?" In RQL sieht das wie folgt aus:

```
SELECT person a FROM [person a] <ist_vorgesetzter_von> [person b]
WHERE person b IS 'Mueller'
```

In diesem Beispiel kommen die wesentlichen Elemente von Metamodellen vor: Entitäten und Relationen. Entitäten sind zum Beispiel Person, Stelle oder Rolle. Relationen sind etwa *ist\_vorgesetzter\_von* oder *ist\_stellvertreter\_von*. Orientiert man sich am Metamodell, legt man in RQL im Grunde einen Pfad fest, der an bestimmte Bedingungen geknüpft ist. Bedingungen an diesen Pfad sind, daß bestimmte Entitäten über bestimmte Pfade miteinander verbunden sind. Zusätzlich lassen sich über die Attribute der Entitäten weitere Bedingungen festlegen.

Ein Beispiel: Gesucht sind alle Mitarbeiter, die in der Rolle Sachbearbeiter involviert sind. Desweiteren sollen die Mitarbeiter in Berlin wohnen und mehr als 32 Jahre alt sein:

```
SELECT person FROM [person] <plays> [role]
WHERE role IS 'Sachbearbeiter' AND person FITS
{ alter > 32 AND wohnort = 'Berlin' }
```

RQL-Anfragen können nun in Workflow-Definitionen aufgenommen werden. Das Workflow-Management-System muß diese Anfragen dann evaluieren. Als Ergebnis erhält es eine Menge von identifizierten Ressourcen (hier Personennamen), die den Anforderungen entsprechen. Die Komponente eines Workflow-Systems, die passenden Ressourcen auswählt und adressiert heißt Resource Manager. Der WOM wurde bisher prototypisch implementiert, um die Funktionsfähigkeit der Konzepte zu zeigen. Als Kommunikationsarchitektur wurde CORBA ver-

wendet. Clients können daher in beliebigen Programmiersprachen geschrieben werden (der Resource Manager ist z.B. in C++ geschrieben). Der Server wurde in Java geschrieben und beinhaltet im wesentlichen den RQL-Parser, der RQL-Anfragen auf Standard-SQL abbildet. Mit Hilfe von JDBC werden die SQL-Statements dann auch einer Oracle-Datenbank ausgeführt. Der Parser wurde mit einem Java-Parsergenerator erzeugt.

### 3.3.3 Prototypische Anwendungen in der Architektur

Der im Arbeitsbereich MWM verwendete Workflow Urlaubsantrag ist im Abschnitt über MWM beschrieben. Ebenso findet sich dort eine Beschreibung des WWW-basierten Gruppenkalenders.

#### Der Workflow Wohnungsvergabe

Im folgenden wird ein Workflow mit seiner prototypischen Realisierung beschrieben, der bislang in ähnlicher Form bei der OFD Berlin durchgeführt wird, allerdings nicht in elektronischer Form.

Im Wohnungsvergabe-Workflow werden Wohnungen aus Berlin an interessierte Behördenmitarbeiter in Bonn vermittelt und vermietet. Der Workflow ist in Aktivitäten aufgeteilt, die von verschiedenen Rollen wie "Sachbearbeiter Wohnungsvergabe", "Bewerber", et cetera bearbeitet werden. Bei der Aktivität *Bewerber auswählen* wird ein Java-Applet verwendet, die übrigen Aktivitäten sind als HTML-Seiten realisiert.

Bevor der Sachbearbeiter bei der Aktivität *Wohnung anbieten* einen Workflow zur Vergabe einer Wohnung anbietet (vgl. Abb. 3.3), kann er sich Daten wie Adresse, Größe, und so weiter, und multimediale Informationen, wie Bilder, Animationen oder Videos, von freien und bereits angebotenen Wohnungen anzeigen lassen. Die Daten werden dabei aus einer Wohnungsdatenbank gelesen. Zusätzlich können bei den freien Wohnungen Werte wie der Mietpreis, die Bewerbungsfrist und ähnliches ergänzt werden; sie werden in der Wohnungsangebotsdatenbank gespeichert.

Nach Ablauf einer Bewerbungsfrist könnten Interessenten benachrichtigt werden, die zuvor über das WWW ihr Interesse für bestimmte Wohnungen registriert hatten. Während der Bewerbungsfrist können sich dann Interessenten für die Wohnung bewerben. In der Aktivität *Bewerber auswählen* bestimmt der Sachbearbeiter zu der Menge der Bewerber eine Bewerberliste, indem er in dem Applet (vgl. Abb. 3.4) einen Bewerber der linken Liste selektiert und der rechten Liste hinzufügt. Dabei kann er sich die Angaben der Bewerber anschauen. Mit dem ersten ausgewählten Bewerber wird dann der Wohnungsvergabe-Workflow fortgesetzt; falls es mit ihm nicht zu einer Vergabe der Wohnung kommt, könnte mit dem nächsten Bewerber der Liste verhandelt werden. Zu allen Bewerbern und zu der Wohnung kann weiterhin eine WWW-Seite aufgerufen werden, in der sich nähere Informationen finden. Die Liste der ausgewählten Bewerber wird ebenfalls in einer Datenbank geführt.



**POLIFLOW - Wohnungsvergabe**

**Freie Wohnungen**  
 1 Am Steinbruch 1  
 Wohnung über anzeigen

**Angestrebte Wohnungen**  
 1 Bachweg 5  
 Wohnung über anzeigen

**auswahl**

**Folgende freie Wohnung wurde ausgewählt:**

Ort: Wohnort  
 Strasse: Am Steinbruch 1  
 Größe: 150  
 Zimmer: 4  
 Wert: 350.000,-  
 Zustand: gehobener Standard  
 Beschreibung: Eine Wohnung mit allem Komfort.

[Anzeige über Beschreibung der Wohnung](#)

**Anbieten der Wohnung**

Zum Anbieten dieser Wohnung müssen noch folgende wohnvergabe Daten eingegeben werden:

Bewerbungsbeitrag:   
 Mietpreis (pro Monat):   
 Frei ab (Datum):

Abbildung 3.3: HTML-Formular zur Bearbeitung der Aktivität *Wohnung anbieten*.

**POLIFLOW - Wohnungsvergabe**

**Bewerberauswahl**

Bewerberliste			Engere Wahl			
Nachname	Vorname	Wohnort	Rang	Nachname	Vorname	Wohnort
Siedemann	Siegfried	12345 Berlin	1	Siebert	Reiner	70565 Stutt...
			2	Sinz	Werner	70565 Stutt...

**Details zur aktuellen Wohnung anzeigen**

**Details zum ausgewählten Bewerber**

Nachname: Siebert  
 Vorname: Reiner  
 Wohnort: 70565 Stuttgart  
 Strasse: Breithwiesenstrasse 20-22

Status

Abbildung 3.4: Applet zur Bearbeitung der Aktivität *Bewerber auswählen*

Der Sachbearbeiter vereinbart anschließend mit dem Bewerber einen Termin für die *Besichtigung*. Dies kann er mittels e-Mail oder Telefon absprechen, alternativ ist bei entsprechender Unterstützung durch den Browser auch eine Telekonferenz aus der Aktivität startbar. Der ausgewählte Bewerber kann sich nun in der Aktivität *Entscheidung* für oder gegen das Mieten der Wohnung entscheiden. Falls er mieten möchte, bereitet ein - eventuell anderer - Sachbearbeiter in *Vertrag vorbereiten* den Vertrag mit dem Bewerber vor. Er hat noch die Möglichkeit, Daten wie den Mietpreis und -datum zu ändern und Vereinbarungen mit dem Bewerber zu vermerken.

Der Vertrag kann in der Aktivität *Vertrag abschließen* vom Bewerber abgeschlossen werden. Dazu könnte seine elektronische Unterschrift verwendet werden. Er könnte hier ebenfalls Änderungswünsche äußern oder den Vertrag ablehnen. Der Abteilungsleiter schließt den Workflow mit der Aktivität *Wohnungsvergabe beenden* ab, wobei er u.a. den Mietvertrag mit seiner elektronischen Unterschrift gegenzeichnet.

## 3.4 Zusammenfassung, Bewertung und Ausblick

Unter Verwendung modernster Technologien wurde ein System entworfen, das kooperatives Arbeiten im Kontext von Vorgangsbearbeitung in weit verteilten Umgebungen unterstützt, und zwar insbesondere weitgehend unabhängig von der zugrundeliegenden Plattform auf Client-Seite. Als Basis-Infrastruktur dient das Intra- bzw. das Internet. Die Client-seitigen Applikationsentwicklung erfolgt in Java. Dieser Ansatz hat insbesondere für den Endbenutzer äußerst entscheidende Vorteile, wie z.B.:

- **Plattformunabhängigkeit, Herstellerunabhängigkeit**  
Durch die Verwendung von Internettechnologien und veröffentlichten Standards, insbesondere von Java und CORBA, können unterschiedliche Plattformen auf Client- wie auf Server-Seite verwendet werden. Außerdem besteht keine Abhängigkeit von einem Hersteller, da mehrere Hersteller Produkte anbieten.
- **Reduktion von Installations- und Wartungsaufwand auf Client-Seite**  
Durch Network Computing kann der Aufwand für die Installation und Wartung auf Client-Seite auf den WWW-Browser beschränkt werden.
- **Integration von Workflow, anderen Anwendungen und Informationsdiensten**  
Anwendungen in Form von Java-Applets können auf natürliche Weise in die Umgebung des Benutzers eingebettet werden, wodurch eine weitgehend homogene Schnittstelle zwischen Mensch und Maschine erreicht wird.
- **Einfachheit der Benutzungsschnittstellen**  
Das auf einfachen Metaphern und wenigen Funktionselementen basierte Workflow User-Interface, die Formular-basierten Aktivitäten und die Einbindung ins e-Mail-System vereinfachen den Zugang zum Workflow-System erheblich und öffnen die Technologie somit auch gelegentlichen oder unerfahrenen Benutzern. Dies wurde insbesondere durch Mitarbeiter der Anwenderbehörde und anderen Benutzern bei den Präsentationen des SWATS-Prototyps bestätigt. Das Konzept der WWW-Integration von Workflow durch Formulare und in Java implementierte Benutzungsschnittstellen ist bei HP zurück in die Produktentwicklung zur Changengine geflossen und findet mittlerweile auch bei anderen Workflow-Herstellern zunehmend Verbreitung.
- **Erweiterbarkeit der Architektur**  
Die Einbindung der Ergebnisse der anderen Arbeitsbereiche AWS, MWM und SWS zeigt bereits, wie weitere Komponenten eingebracht und das System um neue Funktionalität erweitert werden kann.

## 3.5 Literatur

- [Apache] Apache Group. The Apache Server Project. <http://www.apache.org>
- [Böhm95] Böhm, Markus und Schulze, Wolfgang (1995). Grundlagen von Workflow-Managementsystemen. Wissenschaftliche Beiträge zur Informatik, 8(2):50-65, 1995
- [Flan97] Flanagan, David (1997). Java in a Nutshell. Second Edition. O'Reilly & Associates
- [Gall95] Galler, J. (1995). Metamodelle des Workflow-Managements. [http://www.iwi.uni-sb.de/research/contact/cont\\_53.html](http://www.iwi.uni-sb.de/research/contact/cont_53.html)
- [IETF] Internet Engineering Task Force (IETF). <http://www.ietf.org>
- [IETF96] Internet Engineering Task Force (IETF). Hypertext Transfer Protocol -- HTTP/1.0. <http://www.ics.uci.edu/pub/ietf/http/rfc1945.html>
- [Java] Sun Microsystems. Java. <http://java.sun.com>
- [Java97] Javasoft (1997). JDBC: A Java SQL API, Version 1.2. <http://www.javasoft.com/products/jdbc/index.html>
- [HPa] Hewlett Packard. <http://www.europe.hewlett-packard.com>
- [HPb] Hewlett-Packard. HP Changengine V3.0. <http://www.hp.com/>
- [HP97a] Hewlett-Packard (1997). HP Process Manager. Technical Overview. Edition 1.0
- [HP97b] Hewlett-Packard (1997). HP Process Manager. Process Definition Model and Language. Edition 1.0
- [HP97c] Hewlett-Packard (1997). HP Process Manager. Developer Reference Guide. Edition 1.0
- [Kußm97] Kußmaul, Timo 1997. Konzeption und Realisierung eines verteilten Worklist-Handlers zum Einsatz in Intranet und Internet Workflow-Lösungen. Diplomarbeit 1482, IPVR, Universität Stuttgart
- [Mühl96] zur Mühlen, Michael 1996. Der Lösungsbeitrag von Metadatenmodellen beim Vergleich von Workflowmanagementsystemen. <http://www-wi.uni-muenster.de/is/lehre/diplom/ismizu/Index.htm>
- [Musc96] Musciano, Chuck und Kennedy, Bill 1996. HTML- The Definitive Guide. O'Reilly & Associates
- [Muth96] Muth, Gerhard 1996. Entwurf und Implementierung eines User-Managers für eine weitverteilte Workflow-Umgebung. Diplomarbeit 1344, IPVR, Universität Stuttgart.
- [OMG] The Object Management Group. <http://www.omg.org>
- [Wege97] Wegener, Dierk 1997. Entwurf und Realisierung einer Workflowsystem-Komponente zur Verwaltung der Aufbauorganisation eines Unternehmens. Diplomarbeit 1453, IPVR, Universität Stuttgart.
- [WfMC97] Workflow Management Coalition 1997. The Workflow Reference Model. <http://www.aiai.ed.ac.uk/project/wfmc/DOCS/refmodel/rmv1-16.html>
- [WWWC] World Wide Web Consortium. HTML Homepage. <http://www.w3.org/MarkUp/>

# 4 Flexibilität und Anpassungsfähigkeit in WFMS (AWS)

## 4.1 Motivation und Einleitung

Zur Steuerung und Kontrolle von Geschäftsprozessen können Workflow-Management-Systeme (WFMS) eingesetzt werden. Diese Systeme leisten eine Teil- oder Vollautomatisierung der Abwicklung von Prozessen in einem Unternehmen. Die Durchführung erfolgt durch die systematische und effektive Kombination von personellen und maschinellen Ressourcen in der Ablauforganisation des Unternehmens [JaBöSc97].

Bislang werden Workflow-Management-Systeme in Büroumgebungen von Unternehmen wie z.B. Banken oder allgemein in Verwaltungsorganisationen eingesetzt. In diesen Umgebungen sind einzelne Büroabläufe zusammen mit der Verwaltung von beteiligten Personen sowie (elektronischen) Dokumenten im Zuge der Büroautomatisierung offengelegt und dokumentiert. Bei einer fortschrittlichen Organisation kann bereits eine Optimierung der Büroprozesse durchgeführt worden sein, wobei verschiedene Kriterien für die Optimierung zur Anwendung kommen können. Je nach Zielsetzung kann hierbei u.a. die zeitliche oder personelle Optimierung bei der Bearbeitung von Abläufen ein Kriterium darstellen. In diesen Einsatzbereichen handelt es sich bei den Prozessen um stark strukturierte Vorgänge, deren Modellierung durch ein WFMS einfach zu realisieren ist [GeorEA95] [Jablon95a] [LeymEA94].

Der Einsatz in einer weit verteilten Anwendungsumgebung wie z.B. innerhalb der öffentlichen Verwaltung [MayKop95] [Sieber95] erfordert weitergehende Ansätze. Daher muß ein geeignetes System in der Lage sein, unterschiedlichste Vorgangstypen zu unterstützen [KelRom91] [Sieber96] [SiKiSo97]. Neben den bereits erwähnten stark strukturierten Vorgängen, bei denen alle erforderlichen Details bereits während der Modellierung bekannt sind, existieren auch sehr unstrukturierte Vorgänge, bei denen wichtige Kontrollinformationen erst während der Durchführung ermittelt werden können oder bei denen die Berücksichtigung aller möglichen Sonderfälle nur mit enormem Aufwand - zum Teil auch auf Kosten der Übersichtlichkeit eines Prozesses - oder überhaupt nicht möglich ist. Dies führt dazu, daß das WFMS so erweitert werden muß, daß auch hinreichende Unterstützung von unstrukturierten Prozessen gewährleistet werden kann [Sieber97b], was durch die Möglichkeit der Durchführung von Anpassungen an Workflow-Instanzen oder der flexiblen Modellierung von Workflows erreicht werden soll. Im Fall der Anpassungsfähigkeit ist zusätzlich zu beachten, daß prinzipiell alle Workflow-Anwender - falls erforderlich -

in der Lage sein sollten, Anpassungen durchzuführen. Damit erspart man sich lange Anpassungsprozeduren durch Workflow-Spezialisten, da im anderen Fall selbst kleine Änderungen zu langen Anpassungszeiten und hohen Kosten führen würde. Auch durch eine flexible Modellierung können Anpassungen vermieden werden, ohne jedoch auf die Forderung der Unterstützung unstrukturierter Abläufe zu verzichten.

Momentan nimmt die Bedeutung von einfach anpaßbaren und optimierten Geschäftsprozesse aufgrund der sich schnell wandelnden Marktbedingungen und der Konkurrenzsituation stark zu [ReiDad97] [Sheth96] [SieWes98] [WolRei96]. Die Kontrolle der strategischen Prozesse und die Flexibilität dieser Prozesse werden zu relevanten Faktoren für den Markterfolg - von Unternehmen ebenso wie von öffentlichen Dienstleistern.

Aus diesen Gründen wurde im Projekt der Forschungsschwerpunkt der *Adaptiven Workflow-Systeme (AWS)* bearbeitet. Dabei wurden Konzepte entwickelt, realisiert und evaluiert, die eine flexible und anpassungsfähige Beschreibung und Ausführung von Workflows ermöglichen [Sieber97b].

In der Literatur werden häufig Begriffe wie Flexibilität, Anpassungsfähigkeit, dynamische Änderungen oder ad-hoc-Workflows verwendet, ohne daß diese genau definiert werden. Insbesondere die Begriffe strukturiert und unstrukturiert werden im Kontext von Prozessen lediglich mit Beispielen unterlegt. Alle diese Beispiele von unstrukturierten Prozessen besitzen die Eigenschaft, daß bestimmte Details erst während der Ausführung ermittelt werden können, weshalb sie für traditionelle Workflow-Systeme eine Schwierigkeit darstellen [HageEA97].

Als *Struktur* eines Vorgangs kann somit der Grad bezeichnet werden, in dem relevante Details zum Zeitpunkt der Modellierung bekannt sind, bzw. auf welche Art die fehlenden Informationen während der Laufzeit ermittelt und berücksichtigt werden können. In diesem Sinne kann *Anpassungsfähigkeit* im Zusammenhang mit Workflow-Systemen als Fähigkeit definiert werden, die Durchführung auch unstrukturierter Vorgänge unterstützen zu können.

Die wichtigsten Ergebnisse des Arbeitsbereichs sind in diesem Teil des Berichtes zusammengefaßt. Im folgenden Kapitel 4.2 wird die Problemstellung nochmals genauer erörtert und die wichtigsten Anforderungen werden dargelegt, welche den Verlauf der Arbeit beeinflußt haben. Ausgehend von diesen Anforderungen werden in Kapitel 4.3 ein allgemeiner Lösungsansatz und in Kapitel 4.4 die konkret entworfene und realisierte Systemarchitektur beschrieben. Die Benutzerschnittstelle wird im folgenden Kapitel 4.5 genau erläutert, bevor dieser Teil des Berichtes mit einer Bewertung (Kapitel 4.6) der erzielten Ergebnisse und einem Einblick in die weiterführende Fachliteratur (Kapitel 4.7) endet.

## 4.2 Problemstellung und Anforderungen

### 4.2.1 Problemstellung

Bedingt durch ihre Entwicklung und Abstammungslinien wurden Workflow-Systeme bisher überwiegend zur Steuerung und Kontrolle stark strukturierter und genau definierter Vorgänge eingesetzt. Mittlerweile ist jedoch zu erkennen, daß Workflow-Systeme, über die Steuerung und Kontrolle einzelner Prozesse hinaus, enorme Nutzenpotentiale aufweisen. Für einen umfassenden und erfolgreichen Einsatz der Workflow-Systeme gewinnen heute neue Eigenschaften zunehmend an Bedeutung [GeorEA95] [Jablon95b], wie z.B.

- Flexibilität und Anpassungsfähigkeit der Vorgangssteuerung,
- Eignung für unterschiedlichste Vorgangstypen,
- Unterstützung von strukturierten und unstrukturierten Vorgängen,
- Einsatzfähigkeit in einer weit verteilten Anwendungsumgebung.

Um komplexe und vielfältige Vorgänge steuern zu können, ist eine hohe Flexibilität der Vorgangssteuerung mit Benutzerinteraktionen erforderlich. Durch das Eintreten unerwarteter Ereignisse oder bei seltenen Ausnahmen, die nicht alle modelliert werden sollen, ist es notwendig, Workflows auch während der Ausführung modifizieren zu können. Darüber hinaus können durch Anpassungen auch unstrukturierte Vorgänge unterstützt werden, indem diese z.B. nur zum Teil vordefiniert werden und dann während der Ausführung angepaßt, verfeinert oder erweitert werden können. Dadurch wird auch eine bessere Wirtschaftlichkeit erreicht, da in die Modellierungsphase nur die wichtigsten Aspekte einfließen müssen, wohingegen die Feinheiten des Ablaufes durch Anpassungen integriert werden können. Der endgültige Prozeß entwickelt sich also im Laufe der Bearbeitungszeit und steht für jeden weiteren Start eines solchen Prozesses zur Verfügung. Die Anpassungsfähigkeit erlaubt auch unstrukturierten Vorgängen die Steuerung durch ein WFMS, wodurch eine bessere Kontrolle dieser schlecht überschaubaren Prozesse erreicht wird [WainEA96].

Ein Problem bei der Durchführung von Anpassungen ist die Darstellung von Instanzen [Weske 98b]. In der Regel sind Prozesse durch eine Beschreibungssprache dargestellt, die vergleichbar mit einer Programmiersprache ist. Wenn nun aber potentiell allen Workflow-Anwendern, die nicht alle über ein spezielles Expertenwissen verfügen, die Möglichkeit eröffnet werden soll, Anpassungen durchführen zu können, so wird ein Werkzeug benötigt, das eine intuitive, übersichtliche und graphische Repräsentation eines Prozesses bietet, mit welcher der Benutzer die Anpassungen durchführen kann. Er soll dabei auf Standardanpassungen (Delegation, Rücksprache etc.) zurückgreifen, aber auch komplexe Modifikationen an einem Graphen, der den Workflow darstellt, ausführen können.

Dabei darf auch nicht vergessen werden, daß die Anpassungen durch verschiedene Kontrollmechanismen eingeschränkt werden müssen, da selbst einfachste Anpassungen weitreichende Konsequenzen auf den weiteren Ablauf haben können. So kann das Einfügen eines zusätzlichen Teilschrittes in den Ablauf den Prozeß so weit verzögern, daß dies nicht akzeptabel ist. Andererseits kann es aber auch notwendig sein, daß gerade dieser neue Teilschritt eingebracht werden muß, z.B. aufgrund veränderter Gesetzgebung, so daß der Verantwortliche des Prozesses trotzdem die Möglichkeit besitzen muß, diese Anpassung durchführen zu können. Es bedarf also systeminterner Konsistenzprüfungen zur Sicherung der syntaktischen Korrektheit und benutzermodellierter Anpassungsrechte, um die semantische Korrektheit der Workflow-Instanzen sicherstellen zu können [EllisEA95].

Die Durchführung von Anpassungen an einer Workflow-Instanz ist aber nicht der einzige Ansatz, um die verschiedenen Vorgangstypen durch ein WFMS steuern zu können. Ein weiterer Ansatz besteht in der Möglichkeit der flexiblen Modellierung und Ausführung des Vorganges. Bisher wurden starr strukturierte Vorgänge durch ein WFMS gesteuert, die sich dadurch auszeichnen, daß der Ablauf a priori bekannt war. Man benötigte also nur präskriptive Kontrollkonstrukte (wie z.B. Sequenz, bedingte Verzweigung, Parallelverzweigung etc), wie sie auch aus Programmiersprachen bekannt sind. Die Modellierung von teil-/unstrukturierten Prozessen ist mit diesen Konstrukten nur mit sehr hohem Aufwand (durch Modellierung aller möglichen Pfade oder der Durchführung vieler Anpassungen) oder überhaupt nicht möglich. Durch die Einführung deskriptiver Kontrollkonstrukte erreicht man eine flexiblere Modellierungsmöglichkeit, die es erlaubt, speziell teilstrukturierte Vorgänge zu beschreiben [Jablon95a]. Dadurch kann man auch Anpassungen vermeiden, da mittels der beschreibenden Modellierung bereits verschiedene Pfade im Prozeß verankert sind, die nicht oder nur durch Anpassungen möglich wären. Die Vermeidung von Anpassungen hat auch noch den Vorteil, daß zur Laufzeit keine weiteren Konsistenz-, Integritäts- und Anpassungsrechte-Prüfungen erforderlich werden, was enorm zur Stabilität, Sicherheit und Ausführungsgeschwindigkeit der Prozesse beiträgt. Als weiteren Vorteil erhält man zusätzlich die Einbeziehung des Anwenders in die Steuerung des Ablaufes; so ermöglichen deskriptive Kontrollkonstrukte z.B., daß der Anwender Entscheidungen über die Reihenfolge oder die Ausführung von Aktivitäten treffen kann, die dann im weiteren Verlauf des Prozesses, in erforderlicher Weise berücksichtigt werden können. Neben der kontrollierten Ausführung von Prozessen wird damit auch die Eigenständigkeit und Mitbestimmung der Aktoren gefördert.

Durch deskriptive Modellierungskonstrukte muß aber auch die Vorgangsteuerung erweitert werden, da diese Konstrukte eine zusätzliche Funktionalität beinhalten, die durch die rein präskriptiven Konstrukte nicht abgedeckt werden können.

Ein weiteres Problem besteht im Einsatz des WFMS in weit verteilten Anwendungsumgebungen. Die Schwierigkeit dabei ist sicherzustellen, daß Vorgänge in heterogenen technischen Umgebungen und mit Anwendern verschiedenster Benutzerprofile ausgeführt werden können. Aus techni-



scher Sicht müssen hohe Anforderungen an die Integrierbarkeit, Skalierbarkeit und Erweiterbarkeit von WFMS gestellt werden. Da diese Systeme auf die Neugestaltung oder Verbesserung betrieblicher Abläufe abzielen, ist die Forderung nach Integration bestehender Systemkomponenten fundamental. Hierunter fallen nicht nur Anwendungsprogramme sondern vor allem existierende Datenbestände.

## 4.2.2 Anforderungen

In der ersten Projektphase wurde eine ausführliche Analyse des Anwendungsfeldes durchgeführt, bei der sowohl die Prozesse als auch die Infrastruktur untersucht wurde. Die Ergebnisse wurden anschließend verallgemeinert und bildeten die Basis, aus der dann konkrete Anforderungen an ein flexibles WFMS abgeleitet wurde.

Für den Entwurf des Gesamtsystems SWATS mußten, neben den spezifischen Anforderungen zur Flexibilisierung noch weitere Anforderungen aus den Arbeitsbereichen *Sicherheit* und *Gemischte Arbeitsformen* sowie allgemeine Anforderungen an WFMS aus dem Bereich *Workflow Realisierung* berücksichtigt werden. Da diese auch den Entwurf der *Adaptiven Workflow Schicht* (vgl. auch Kapitel 4.4) beeinflußt haben, werden die wichtigsten Anforderungen im folgenden kurz aufgeführt.

### Allgemeine Anforderungen an innovative WFMS

- Verfügbarkeit an allen Arbeitsplätzen
- Verteilter Zugriff auf das System und Unterstützung heterogener Rechnersysteme
- Ergonomie und Variabilität der Benutzerschnittstellen
- Offenheit und Erweiterbarkeit
- Sicherheit und Vertraulichkeit
- Integrationsfähigkeit mit anderen Systemen (z.B. Groupware-Anwendungen, Videokonferenzen) und Altanwendungen, Interoperabilität
- Skalierbarkeit und Zuverlässigkeit

### Spezielle Anforderungen an Adaptive WFMS

- Dynamische Änderungen an Workflow-Instanzen zur Laufzeit  
Damit soll ermöglicht werden, daß in unerwarteten Situationen, bei Fehlern oder unvollständiger Modellierung der Workflows der Ablauf des Workflows während der Ausführung beeinflußt werden kann. Diese Änderungen betreffen zunächst eine spezielle Instanz, sollten aber auch auf andere ausgewählte Instanzen übertragbar sein [EllisEA95].
- Bereitstellen vordefinierter Anpassungen  
Um die Anwender von vorhersehbaren und Anpassungen zu entlasten sollen die Anpassungen

auch automatisch ausführbar sein. Damit soll die Möglichkeit der Modellierung erweitert werden, indem automatische Anpassungen auch durch eine Workflow-Instanz selbst vorgenommen werden können.

- Vermeidung von Anpassungen durch deskriptive Konstrukte  
Um manuelle Anpassungen, die immer mit einem personellen Aufwand verbunden sind, zu vermeiden, sollte die Modellierung der Workflows so flexibel zu gestalten sein, daß vorhersehbare Ausnahmen und Fehlerfälle auch in einer übersichtlichen und angemessenen Modellierung berücksichtigt werden können.
- Beteiligung aller Workflow-Partizipanten  
Anpassungen sollten nicht nur von privilegierten Personen (z.B. Administratoren) oder extremen Spezialisten (z.B. Prozeßmodellierern) vorgenommen werden können, sondern prinzipiell von allen Workflow-Beteiligten. Dies bedeutet auch, daß die Benutzerschnittstelle für alle betroffenen Aktoren geeignet sein muß.
- Kontrolle der Anpassungen an Workflow-Instanzen  
Wenn auch gefordert wird, daß die Workflows hochgradig anpassungsfähig sind, so muß dennoch dafür Sorge getragen werden, daß keine unerwünschten oder inkorrekten Anpassungen durchgeführt werden können.

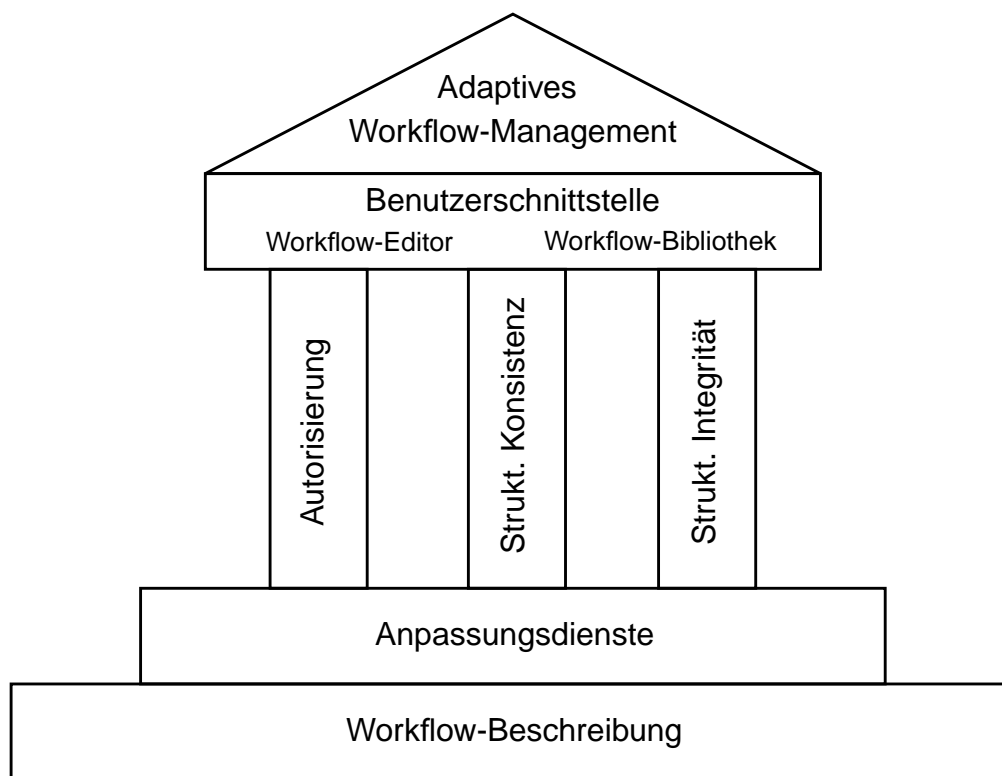
## 4.3 Lösungsansatz

Um die beschriebenen Anforderungen an ein adaptives WFMS erfüllen zu können, reichen isolierte Ansätze wie z.B. die Änderbarkeit von Workflow-Instanzen oder die Integration von unstrukturierten Gruppenaktivitäten nicht aus. Vielmehr ist ein ganzheitlicher Ansatz erforderlich, der Mittel zur Flexibilisierung enthält, ohne die Stärken der bestehenden Workflow-Technologie wieder zu mindern. Hierfür wurde ein integrierter Lösungsansatz entwickelt, dessen Konzepte im folgenden vorgestellt werden (vgl. auch Abb. 4.1).

### **Flexible Vorgangsteuerung**

Um unnötige Anpassungen zu vermeiden, wird auf einem sehr mächtigen und flexiblen Workflow-Modell aufgesetzt, mit dem z.B. komplizierte Kontrollflüsse durch die Verwendung interner und externer Ereignisse oder Schleifen modelliert und ausgeführt werden können.

Durch die Modellierung deskriptiver und komplexer Kontrollkonstrukte können Ausnahmesituationen berücksichtigt werden, ohne die Übersichtlichkeit des Modells zu gefährden. Darüber hinaus können komplexe Kontrollflußabhängigkeiten durch Konstrukte, die über eine Bibliothek zur Verfügung gestellt werden, auf einfache Art und Weise berücksichtigt werden. Durch eine angemessenere Modellierung können u.U. bereits einige unnötige Anpassungen vermieden werden.



**Abbildung 4.1:** Integrierter Lösungsansatz: Anpassungsfähige Workflow-Systeme

### Workflow-Anpassungsdienste

Um die einzelnen Instanzen ändern zu können, werden sie zur Ausführungszeit unabhängig von ihren Definitionen verwaltet. Über definierte Schnittstellen können die Instanzen dann geändert werden, wobei eine hierfür geeignete Zwischenrepräsentation der Workflow-Instanzen verwendet wird. Die Anpassungsschnittstelle verfügt über elementare Änderungsoperationen, die alle Aspekte der Workflow-Beschreibung betreffen und darüber hinaus in komplexen Transaktionen zusammengesetzt werden können. Besondere Schwerpunkte liegen dabei auf den funktionalen, verhaltensorientierten, informationalen, organisatorischen und zeitlichen Aspekten.

Durch diese Anpassungen können zur Laufzeit alle Workflows erzeugt werden, die auch mit Hilfe der Modellierungswerkzeuge erstellt werden können.

### Modellierung von Anpassungsrechten

Insbesondere bei manuellen Anpassungen durch den Benutzer stellt sich die Frage, welche Anpassungen von wem und unter welchen Umständen (Zustand des Workflows, zusätzliche erforderliche Aktionen etc.) durchgeführt werden dürfen. Durch die Erweiterung der Workflow-Definition um Anpassungsrechte (Teil der Anpassungsaspekte) kann dies beliebig genau spezifiziert

werden. Dabei können sowohl elementare als auch komplexe Anpassungen berücksichtigt werden; bei der Spezifikation des Zustandes kann ebenfalls wieder auf unterschiedliche Aspekte des Ausführungszustandes und des weiteren Ablaufplanes zurückgegriffen werden.

### **Kontrolle und Gewährleistung der Konsistenz**

Bei der Durchführung der Anpassungen ist darauf zu achten, daß die Konsistenz des Workflows nicht verletzt wird. Dabei muß zum einen der aktuelle Zustand des Workflows (Ausführungsmodell) und zum anderen das Definitionsmodell berücksichtigt werden. Die Konsistenzbedingungen beziehen sich zum Teil auf einen Aspekt (z.B. müssen Daten, die gelesen werden auch irgendwo geschrieben worden sein), zum Teil aber auch auf Abhängigkeiten zwischen Aspekten (z.B. Kontrollfluß und Datenfluß).

Durch das Aufstellen geeigneter, systemimmanenter Konsistenzbedingungen, die bei der Durchführung von Anpassungen überprüft werden, können die Instanzen zu jeder Zeit konsistent gehalten werden. Das gilt auch bei Anpassungen im Mehrbenutzerbetrieb, bei dem die Workflow-Instanzen durch herkömmliche Sperren gesichert werden.

### **Kontrolle und Gewährleistung struktureller Integritätsbedingungen**

Durch die Definition der Anpassungsrechte wird lediglich der Zustand vor der Anpassung überprüft, und die konsistenzhaltenden Maßnahmen beschränken sich lediglich auf die syntaktische Korrektheit der Instanzen. Die Semantik der angepaßten Instanz bzw. der Änderungen wird damit noch nicht berücksichtigt.

Hierfür können in der Workflow-Definition Integritätsbedingungen frei definiert werden, die dann bei der Durchführung von Änderungen zusätzlich kontrolliert und damit auch nach der Änderung garantiert werden. Die Integritätsbedingungen können sich ebenfalls auf unterschiedliche Aspekte beziehen und bilden zusammen mit den Anpassungsrechten einen umfangreichen Schutz der Instanzen vor unerwünschten Anpassungen.

### **Anwendergerechte Benutzerschnittstelle zur Durchführung der Anpassungen**

Mit den bisher beschriebenen Maßnahmen sind kontrollierte Anpassungen technisch durchführbar. Um sie allerdings praxistauglich zu gestalten, müssen sie zum einen von allen Anwender durchführbar sein, und zum anderen muß der Aufwand in einem vernünftigen Verhältnis zum Nutzen stehen.

Damit der Aufwand für die Definition von Anpassungsrechten und Integritätsbedingungen reduziert werden konnte, wurden verschiedene Konzepte eingeführt, wie z.B. die Definition von Anpassungsrechten und Verboten, die Typisierung und Gruppierung von Workflows und Anpassungen und die Vererbung der Definitionen eines Workflows an seine Subworkflows.

Weitere Maßnahmen erleichtern auch nicht spezialisierten Anwendern den Umgang mit den Anpassungen. Hierfür wurde ein Editor erstellt, der die Workflow-Instanzen in mehreren Sichten darstellt und über einfache graphische Operationen Änderungsdienste anbietet. Das Verständnis

der Prozesse wird u.a. auch durch die Darstellung definierter Kontrollflußkonstrukte erleichtert. Durch Bibliotheken von vordefinierten Objekten (Aktivitäten, Dokumenten, Aktoren etc.) und komplexen Anpassungen (z.B. Delegation, Wiedervorlage, Verschieben von Aktivitäten, Parallelisierung von Aktivitäten etc.) wird die Handhabung auch für weniger geschulte Anwender erleichtert [Sieber97a].

Durch diesen integrierten Lösungsansatz können einfache ad hoc-Modifikationen durch alle Anwender vorgenommen werden und somit auch unstrukturierte Prozesse unterstützt werden, deren Ausführung vor dem Start noch nicht genau festgelegt ist. Auf diese Weise besteht die Möglichkeit, explorative Vorgänge zu unterstützen oder teilweise vordefinierte Prozesse durch Erweiterungen oder Verfeinerungen zur Laufzeit anzupassen.

## 4.4 Architektur

Im folgenden Abschnitt wird die Architektur des entwickelten Systems vorgestellt. Dazu werden zuerst die verwendeten Technologien beschrieben. Anschließend wird die der Entwicklung zugrundeliegende Referenzarchitektur angeführt, der dann die erweiterte Architektur folgt.

### 4.4.1 Technologien

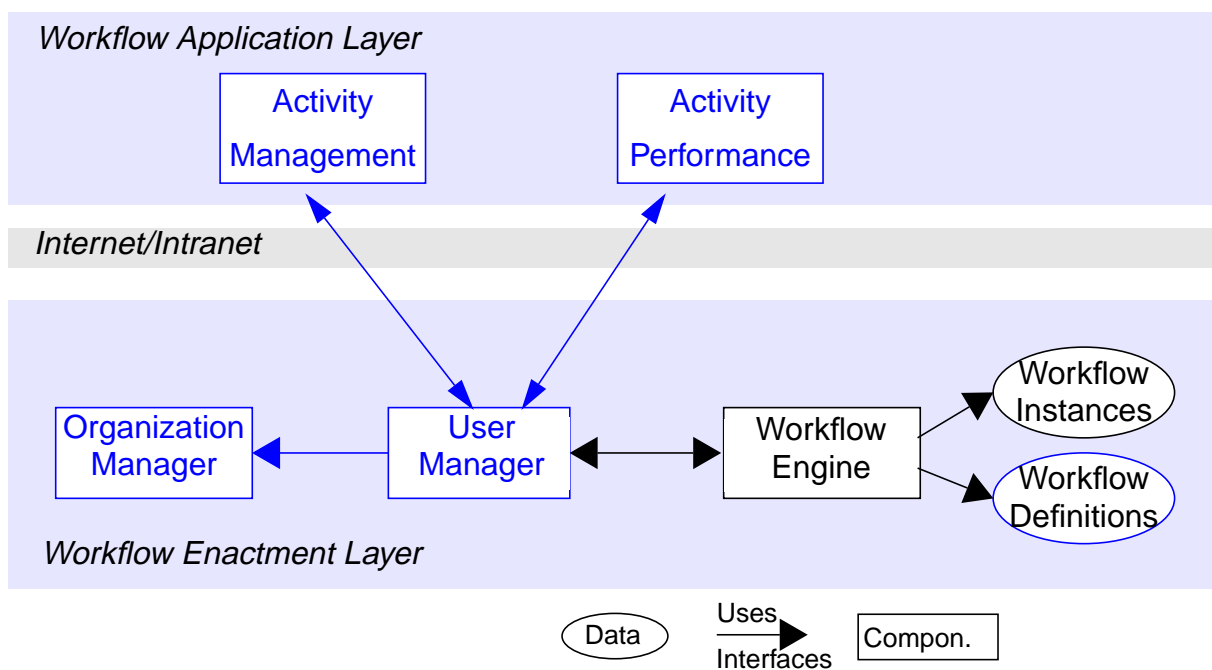
Aufgrund der bestehenden Anforderungen an das System und der Anforderungen, die aus anderen Teilbereichen des Projekts herrühren, wurde eine Client-Server-Architektur entworfen, die auf folgende Technologien aufsetzt (vgl. auch Kapitel 3):

- die Intra-/Internet-Technologie mit dem HTTP-Protokoll bildet die Basis-Infrastruktur und dient der Kommunikation zwischen den Benutzer-Clients und dem Workflow-Server.
- HTML, unter anderem mit den Möglichkeiten zur Gestaltung von Informationsseiten und einfachsten graphischen Benutzeroberflächen in WWW-Browsern (mit dem Datenaustausch via CGI), gestattet den Zugang zu Workflow-Clients und Aktivitäten.
- JAVA wird für Applets eingesetzt, mit denen komplexe Aktivitätsimplementierungen und Workflow-Clients realisiert und in HTML-Seiten eingebettet werden. Dadurch kann aus dem WWW auf CORBA-Objekte zugegriffen werden.
- CORBA wird für anspruchsvolle Kommunikationsschnittstellen zwischen den SWATS-Komponenten benutzt. Es erlaubt die plattform- und programmiersprachenunabhängige Integration von Softwarekomponenten in das System.

- RMI wird ebenfalls wie CORBA für Kommunikationsschnittstellen zwischen rein in JAVA implementierten SWATS-Komponenten benutzt. Mit der Weiterentwicklung von JAVA, die eine direkte Integration von CORBA in das JDK mit sich bringt, können diese Schnittstellen in Zukunft ebenfalls in CORBA umgesetzt werden.

#### 4.4.2 Referenzarchitektur für traditionelle WFMS

Innerhalb der SWATS-Architektur (vgl. Abb. 4.2), die als Referenzarchitektur betrachtet werden kann, können zwei Schichten identifiziert werden: eine serverseitige Workflow-Ausführungsschicht (Workflow Enactment Layer) und eine clientseitige Applikationsschicht (Workflow Application Layer).



**Abbildung 4.2:** Basisarchitektur und Schichten

Die *Workflow-Engine* entscheidet aufgrund des aktuellen Zustands eines Vorgangs, welche Aktivitäten als nächstes zu bearbeiten sind. Die Zuweisung der Aktivitäten an die betreffenden Ressourcen erfolgt mit Hilfe des *Organisations-Managers*. Dieser ist für die Auflösung von Rollen zu Ressourcen zuständig. Durch den Zugriff auf Information über die Aufbauorganisation liefert er auf Anfragen des *User-Managers* - gemäß der in der Workflow-Beschreibung spezifizierten Rollenbeschreibung - einen oder mehrere mögliche Bearbeiter für eine Aktivität zurück.

Der *User-Manager* dient als Zugang für die Endbenutzer-Clients, die Teil der Applikationsschicht sind. Er bietet eine reduzierte, spezialisierte Schnittstelle zum Workflow-System und ermöglicht so eine Vereinfachung der Benutzerschnittstelle und einen erleichterten Austausch mit der

Engine. Der User-Manager ist zuständig für die Verwaltung der Arbeitslisten, welche die Aktivitäten eines Anwenders enthalten. Jeder Anwender kann dabei, aus dem User-Interface der Applikationsschicht heraus, auf die eigene, dynamische Arbeitsliste im User-Manager zugreifen und diese durch Aktionen beeinflussen, wie z.B. durch Annahme der Aktivität oder Start bzw. Abschluß der Bearbeitung.

In der Applikationsschicht wird die Benutzerschnittstelle umgesetzt. Die Komponente Aktivitäten-Management dient dabei als zentrale Zugangsschnittstelle für den Endbenutzer. Über diese erhält er Arbeitslisten sowie Funktionen zur Nutzung der Dienste der Ausführungsschicht. Neben diesen administrativen Aufgaben wird durch die Applikationsschicht die Ausführung der Aktivitäten übernommen, dargestellt durch die logische Komponente Aktivitäten-Ausführung.

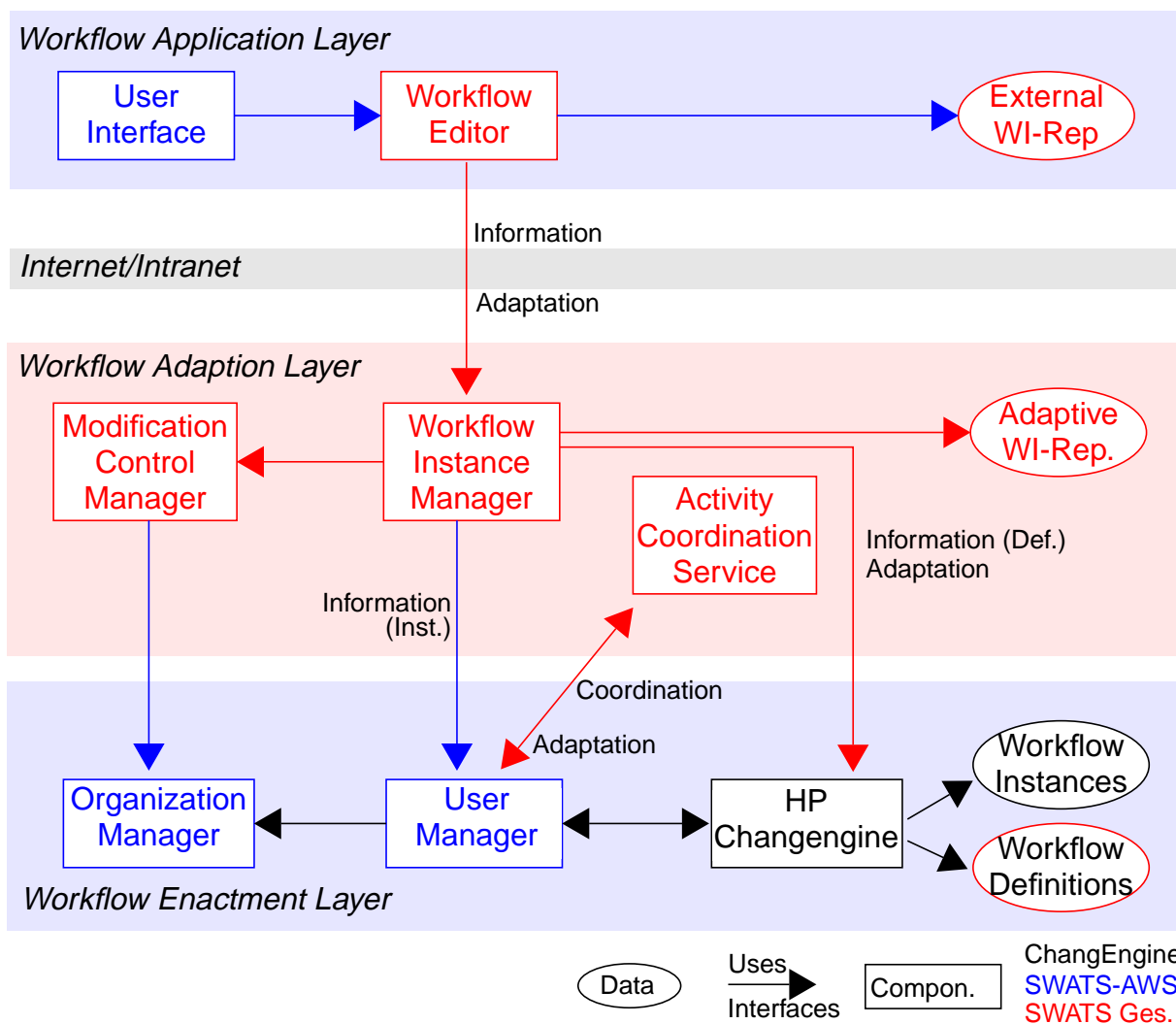
Bei der Realisierung dieser Architektur wurde als Workflow-Engine das Produkt *Changengine* von Hewlett-Packard verwendet, das eine auf CORBA basierende Kommunikations-Infrastruktur zur Verfügung stellt. Alle anderen Komponenten wurden im Rahmen des Projektes PoliFlow am IPVR entwickelt (vgl. Kapitel 3).

### **4.4.3 Erweiterte Architektur für adaptive WFMS**

Die Referenzarchitektur aus dem vorherigen Abschnitt wurde aufgrund der bestehenden Anforderungen aus dem Bereich der anpassungsfähigen WFMS (AWS) und der weiteren Projektschwerpunkte erweitert. Dabei wurde auf der Server-Seite eine zusätzliche Schicht, die Workflow-Anpassungsschicht eingeführt (siehe Abb. 4.3). Diese Schicht beinhaltet verschiedene logische Komponenten, mit denen traditionelle Workflow-Systeme um Funktionalität zur Flexibilisierung und Anpassungsfähigkeit von WFMS erweitert werden können. Dabei stand die Übertragbarkeit und Anwendbarkeit auf verschiedene, bereits existierende Systeme im Vordergrund [Sieber98].

Für die Ausführung flexibler, anpassungsfähiger Workflows wurden Erweiterungen am User Manager notwendig, die es erlauben, mit dem neu entwickelten, anpassungsfähigen Workflow-Modell (s.u.) zusammenzuarbeiten und gleichzeitig das starre Workflow-Modell der zugrundeliegenden Engine zu unterstützen. Diese Erweiterungen betreffen folgende Eigenschaften:

- Integration des *Workflow Instance Managers* (WIM) und des *Activity Coordination Services*
- Verwaltung von Nachrichtenkanälen auf Basis des CORBA-Event-Services, um die integrierenden Dienste des User Managers für die zusätzlichen Komponenten bereitzustellen.
- Erweiterung der eigenen Schnittstelle so, daß die Komponenten der zusätzlichen Schicht bedingten Einfluß auf Aktivitäten nehmen können.



**Abbildung 4.3:** SWATS-Architektur

Der Workflow Instance Manager spielt in der Gesamtarchitektur für anpassungsfähige Workflows eine zentrale Rolle. Zum einen nutzt er Instanzen des flexiblen, anpassungsfähigen Workflow-Modells (Adaptive WI-Representation), die ein Abbild der von der Engine gehaltenen, originären Instanzen darstellen (Workflow-Instances), zum anderen erlaubt er die Durchführung von Anpassungen. Dafür stellt er zum Workflow-Editor eine Anpassungsschnittstelle mit fest definierten Anpassungen zur Verfügung. Die Schnittstelle erlaubt einfache Anpassungen ebenso, wie komplexe, transaktional durchzuführende Anpassungen.

Die Instanzen werden über eine Schnittstelle zur Workflow-Engine erzeugt und im WIM verwaltet. Sämtliche Anpassungen werden dann zuerst auf den internen Instanzen des WIM ausgeführt und anschließend an die Engine über eine ebenfalls wohldefinierte Schnittstelle propagiert. Über die Schnittstelle zum User-Manager können evtl. zusätzliche aktuelle Instanzinformationen empfangen werden, die dann in die jeweiligen Workflow-Instanzen eingebracht werden.



Über die Informationsschnittstelle zum Workflow-Editor ist auch ein Monitoring einzelner Instanzen möglich. Im Editor werden die Instanzen graphisch dargestellt, wodurch den Anwendern ein guter Überblick über den aktuellen Bearbeitungsstand eines Prozesses ermöglicht wird.

Innerhalb des Schwerpunktes AWS wurde ein anpassungsfähiges Workflow-Modell entwickelt (Adaptive WI-Rep), das auf dem Modell der ausführenden Engine basiert [Hofman98]. Um die Anpassungsfähigkeit zu gewährleisten, wurden einige Erweiterungen eingeführt.

Zum einen wurde das Modell modular aufgebaut, so daß es auf Bedarf einfach erweitert und damit an anwendungs- oder systemspezifische Gegebenheiten angepaßt werden kann. Der Aufbau des Kontrollflusses wurde dabei strukturiert gestaltet, wodurch Anpassungen am Kontrollfluß einfacher zu überprüfen sind. Des weiteren wurden deskriptive Kontrollkonstrukte mit aufgenommen, die eine flexiblere Modellierung gerade im Hinblick auf teilstrukturierte Workflows ermöglichen.

Zum anderen wurden in das Modell neue (Anpassungs-) Aspekte aufgenommen, die zur Kontrolle von Anpassungen erforderlich sind; so können zu einzelnen Kontrollflußkonstrukten flexibel definierbare Anpassungsrechte für Benutzer(-gruppen) spezifiziert werden. Zusätzlich wurden Workflow-Typisierungen eingeführt, die verschiedene Aspekte eines Workflows betreffen, wie z.B. zeitliche- und Sicherheitsaspekte.

Die Verifikation einer Anpassung an einer Instanz erfolgt durch den *Modification Control Manager*, der durch den WIM im Zuge einer Anpassung angesprochen wird. Er kann auf Instanzinformationen lesend zugreifen und anhand der in der Definition spezifizierten Anpassungsrechte entscheiden, ob die Modifikation durchgeführt werden kann oder nicht. Das Ergebnis wird wiederum vom WIM verarbeitet. Bei manchen Ergebnissen können durch den WIM auch weitere Aktionen ausgelöst werden, die zur endgültigen Durchführung der Anpassung notwendig sind. Dazu zählt z.B. die Benachrichtigung einer verantwortlichen Person des Prozesses über die Anpassung. Die Informationen, welche Aktionen ausgelöst werden sollen, werden ebenfalls zur Modellierungszeit innerhalb der Anpassungsaspekte des Workflows festgelegt.

Um die Ausführung der komplexen Kontrollkonstrukte durch die Engine zu ermöglichen, wurde der *Activity Coordination Service* eingeführt. Er ermöglicht, in Zusammenarbeit mit dem User Manager, die Ausführung und Steuerung komplexer Kontrollkonstrukte. Diese können durch vordefinierte Makros in der originären Workflow-Definition modelliert werden. Der Koordination der Aktivitäten erfolgt damit für den Benutzer transparent - mit Hilfe des Activity Coordination Service.

In die Workflow-Applikationsschicht wurde ebenfalls eine neue Komponente integriert: der Workflow-Editor [Blum 98]. Er wird über das User-Interface des Benutzers aufgerufen und erlaubt das Monitoring eines Prozesses ebenso, wie die benutzergerechte Durchführung von Anpassungen. Dazu werden die (RMI-) Schnittstellen des WIM benutzt (*Information/Adaption*),

über die aktive Prozesse ausgelesen und modifiziert werden können. Aus Gründen der Performance und Unabhängigkeit (von dem internen Modell des WIM) arbeitet der Editor mit einer eigenen Instanzrepräsentation (External WI-Rep), die für unterschiedliche Ansichten des Editors optimiert wurde. Eine detaillierte Beschreibung des Editors findet sich im nächsten Abschnitt.

Das User-Interface erfuhr ebenfalls eine Erweiterung. So lassen sich in der erweiterten Version Aktivitäten durch den Benutzer ablehnen, falls diese als optional ausführbar modelliert wurden. Damit kann ein Benutzer über das User-Interface jetzt direkt die Ausführung einer solchen Aktivität ablehnen, wenn es z.B. nicht mehr notwendig sein sollte, diese zu bearbeiten.

## 4.5 Benutzerschnittstellen

Die Akzeptanz und tatsächliche Anwendung eines anpassungsfähigen WFMS wird stark durch die Benutzerschnittstelle beeinflusst. An dieser Stelle wird die im Schwerpunkt AWS entwickelte Benutzerschnittstelle, der Workflow-Editor [Blum 98] beschrieben. Ziel des Workflow-Editors ist es, ein Werkzeug bereitzustellen, mit dem zum einen Workflow-Instanzen dargestellt und zum anderen Anpassungen an laufenden Instanzen durchgeführt werden können. Des Weiteren ist es möglich, das Werkzeug zur Modellierung von Workflows einzusetzen. Der Editor stellt das Workflow-Modell geeignet dar und für die Durchführung von Anpassungen geeignete Interaktionsmöglichkeiten bereit.

Es kann davon ausgegangen werden, daß die Benutzer des Editors über durchschnittliche Erfahrungen mit dem Umgang mit Rechnern verfügen und mit einzelnen Prozessen vertraut ist. Die Bedienung des Editors ist einfach und funktionell und wird eine integrierte Online-Hilfe (HTML) unterstützt; gleichartige Operationen werden auf die gleich Art und Weise ausgeführt. Der Editor ist sowohl für Administratoren als auch für Bearbeiter von Workflows konzipiert. Je nach Art der durchzuführenden Anpassung muß der Benutzer über mehr oder weniger tiefgreifendes Wissen der Workflow-Modellierung verfügen.

Bei der Gestaltung der Benutzerschnittstelle stand zum einen der einfache Wechsel zwischen den einzelnen Sichten und zum anderen die möglichst einfache Durchführung von Anpassungen im Vordergrund.

Workflow-Modelle sind in der Regel äußerst komplex und beinhalten eine Vielzahl von Informationen und einzelnen Aspekten. Da die wahrgenommenen Informationen auch verarbeitet werden müssen, wird die menschliche Wahrnehmungsfähigkeit durch die Leistungsfähigkeit menschlicher Denkvorgänge zur Speicherung und Verarbeitung von Informationen begrenzt [Stary94].

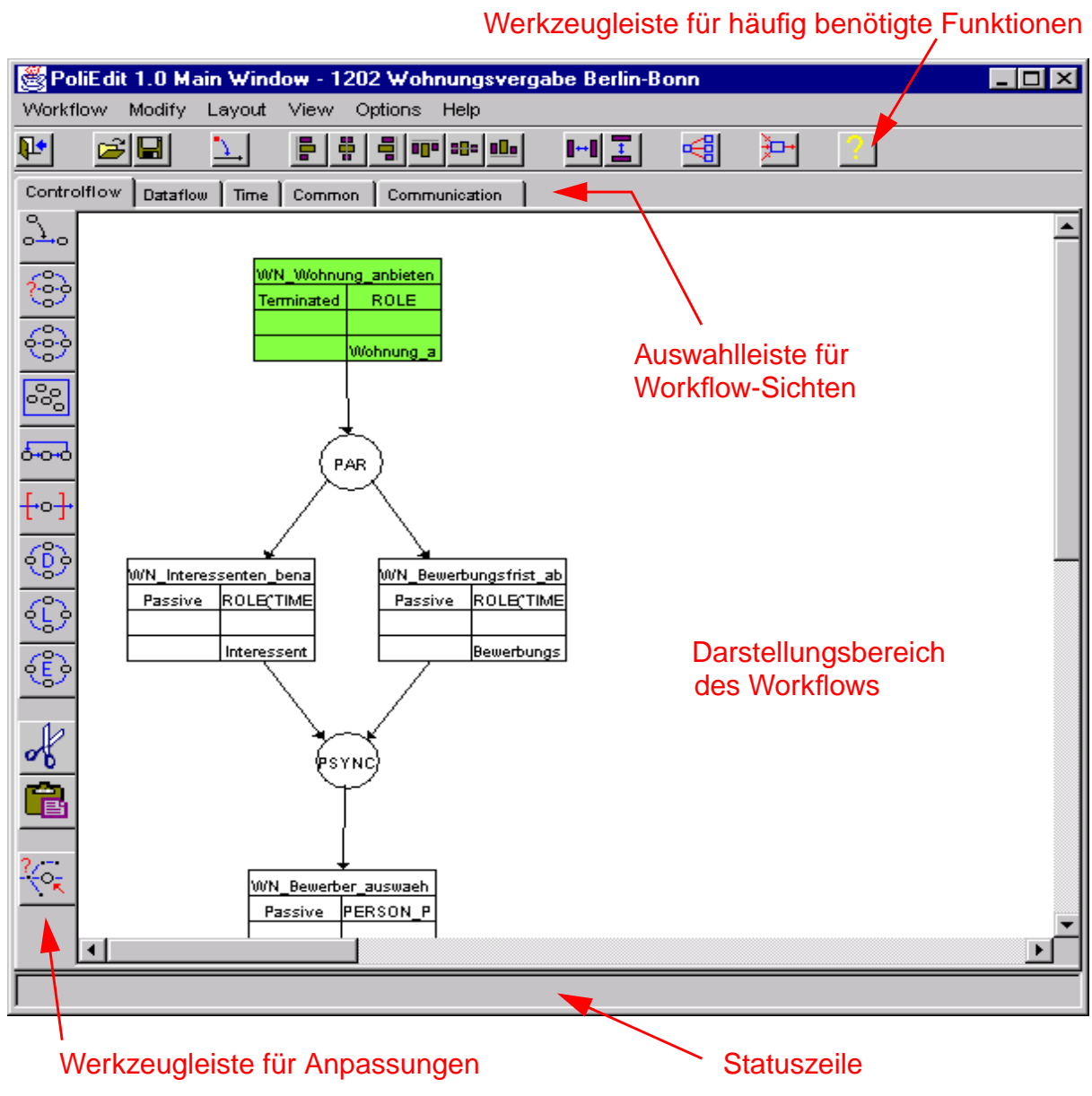
Damit das System jeweils nur die Informationen darstellt, die vom Benutzer zur Erfüllung seiner Aufgabe benötigt werden, wurde eine Aufteilung der Informationen auf verschiedene Sichten erforderlich.

Da mit dem Editor nicht nur Workflows dargestellt sondern auch manipuliert werden, müssen entsprechende Interaktionsmöglichkeiten bereitgestellt werden. Beim Design des Editors wurde daher auf die Bereitstellung intuitiver Mechanismen zur Durchführung von Anpassungen großen Wert gelegt.

Die Analyse der Anforderungen an den Editor ergab, daß mindestens vier Sichten zur optimalen Präsentation der Workflows bereitgestellt werden müssen. Durch weitere Anforderungen aus dem Projekt wurde eine weitere Sicht bereitgestellt, die - für mit dem Prozeß vertraute Personen - einen schnellen Überblick über die Workflows bietet, da lediglich die Aktivitäten und deren grobe Ablaufstruktur dargestellt wird. Die folgenden Sichten wurden entworfen und realisiert:

- **Allgemeine Sicht**  
Mit dieser Sicht wird es dem Benutzer ermöglicht werden, einen Überblick über den Workflow zu bekommen. In dieser Sicht wird nur der (vereinfachte) Ablaufgraph des Workflows dargestellt.
- **Kontrollfluß-Sicht**  
Diese Sicht dient dazu, den Kontrollfluß darzustellen und zu modellieren. Nur in dieser Sicht sind Manipulationen der funktionalen und verhaltensorientierten Aspekte möglich.
- **Datenfluß-Sicht**  
Diese Sicht dient dazu, den die Datenverwendung (lesend/schreibend) und die Datenflüsse darzustellen und zu modellieren. Der Kontrollfluß wird hierbei auf explizite Vorgänger-/Nachfolgerbeziehungen der Aktivitäten reduziert.
- **Zeitliche Sicht**  
Durch diese Sicht wird der zeitliche Verlauf der Workflows, in einer an Pert-Charts angelehnten Notation dargestellt und modelliert. Neben der Darstellung von Zeiten über Attribute, werden die Aktivitäten an dynamischen Zeitachsen ausgerichtet.
- **Kommunikations-Sicht**  
Diese Sicht hebt den organisatorischen Aspekt hervor und ermöglicht dem Anwender bei auftretenden Fragen möglichst unkompliziert mit den jeweiligen Gesprächspartner Kontakt aufzunehmen (vgl. Kapitel 6).

Für alle Sichten gilt, daß alle für den jeweiligen Aspekt nicht zweckdienlichen Informationen ausgeblendet werden. Es ist dem Benutzer jedoch freigestellt, welche Informationen er ein- und ausblenden will, d.h. die Darstellung verschiedener Informationen ist durch den Benutzer frei konfigurierbar. Zwischen den einzelnen Sichten kann jederzeit gewechselt werden.



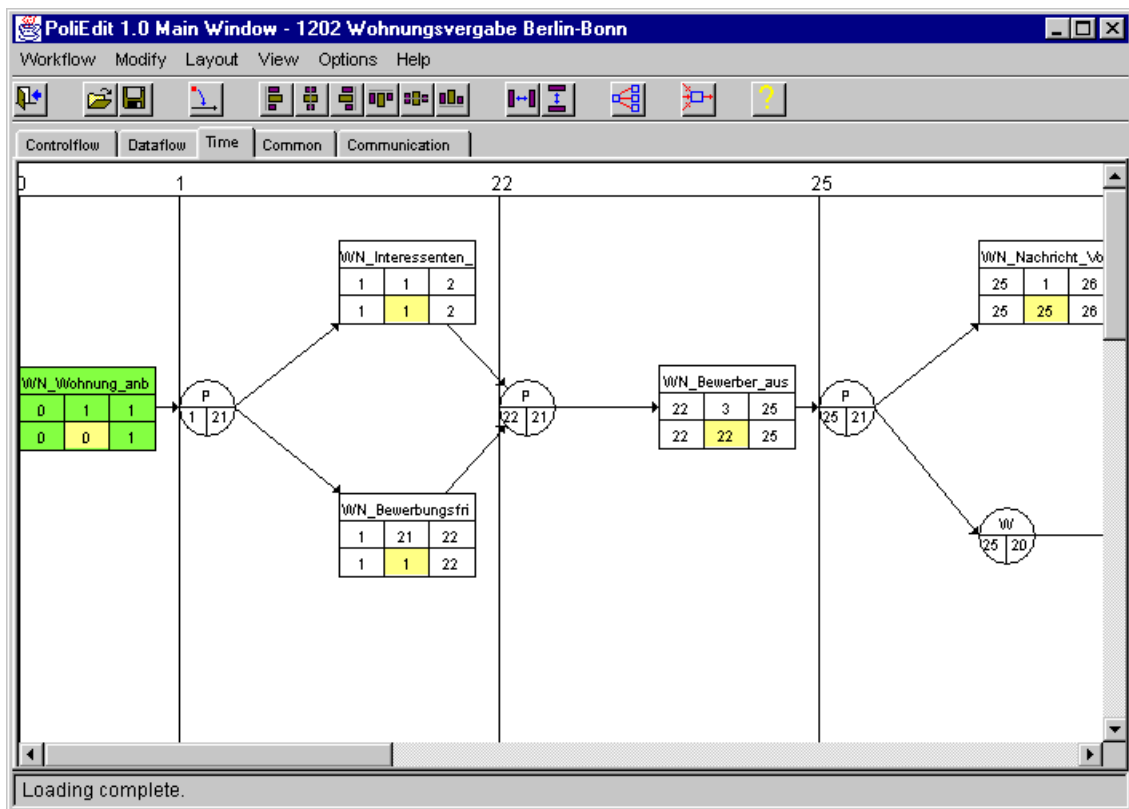
**Abbildung 4.4:** Kontrollfluß-Sicht

In Abb. 4.4 ist die Oberfläche des Editors in der Kontrollfluß-Sicht zu sehen. Die Werkzeuggeste für Anpassungen ist eine Sammlung von graphischen Bedienelementen, die das Ziel verfolgt, dem Benutzer über kurze Interaktionswege häufig benutzte Funktionen bereitzustellen. Der Editor benützt eine solche Werkzeuggeste zum möglichst einfachen und intuitiven Einfügen von neuen Aktivitäten oder Konstrukten in den Workflow per Drag&Drop. So kann in dieser Darstellung ein gewünschtes Objekt durch Ziehen aus der Werkzeuggeste auf eine Kante oder einen Knoten des Graphen eingefügt werden. Angewählte Zielorte verändern dabei ihre Farbe, so daß sofort ersichtlich ist, ob an einer Stelle des Graphen eine Anpassung am Kontrollfluß überhaupt möglich ist oder nicht. Nach dem Erreichen der Einfügeposition läßt man das Konstrukt fallen und erhält je nach eingefügtem Konstrukt weitere Dialoge, in denen z.B. der Namen des Kon-

struktus eingegeben werden muß. Nach der Angabe aller benötigten Informationen über die Dialoge wird sofort die Darstellung des Graphen aktualisiert. Die durchgeführte Anpassung wird dann an den WIM propagiert, wo eine Überprüfung der Anpassung erfolgt. Die erfolgreiche Ausführung einer solchen Anpassung wird in der Statuszeile angezeigt. Falls die Anpassung nicht durchgeführt werden darf, wird durch den Workflow-Editor der alte Stand des Prozesses vom WIM geholt und die Anzeige wieder entsprechend angepaßt. Man erhält also in jedem Fall eine visuelle Meldung über die Durchführung einer Anpassung.

Die Werkzeugleiste ist nur in der Kontrollfluß-Sicht verfügbar, da nur dort Anpassungen der Ablaufbeschreibung möglich sind. Zwischen den einzelnen Sichten kann zu jeder Zeit beliebig gewechselt werden.

Abb. 4.5 zeigt beispielhaft die zeitliche Sicht eines Workflows, mit der Zeitachse am oberen Rand des Darstellungsbereiches, an der die Aktivitäten nach unterschiedlichen Kriterien (frühester/spätester/erwarteter Anfangs-/Endetermin) ausgerichtet sind.



**Abbildung 4.5:** Zeitliche Sicht

In der Konstruktsicht (siehe Abb. 4.6) können weitergehende Informationen über Aktivitäten betrachtet oder geändert werden. Sie beinhaltet zusätzliche Informationen zu den Konstrukten, die am übersichtlichsten über Attributwerte dargestellt und einfach über Tastatureingabe angepaßt werden können. Auch diese Sicht enthält wieder Unterbereiche, die den größtenteils den Aspekten zuzuordnen sind.

Die Unterbereiche der Konstruktsicht enthalten folgende Informationen:

- Allgemeine Attribute (Name, Rolle, Akteur, Beschreibung etc.)
- Dokumentenfluß und Datenfluß (Datum, Typ, Zugriffsart etc.)
- Zeitliche Attribute (Termine, Dauer, Maßnahmen bei Terminüberschreitung etc.)
- Aktueller Zustand (angeboten, gestartet etc.)
- Ereignisse (Auslösen von oder Synchronisation mit externen/internen Ereignissen)
- Verzweigungsregeln (z.B. bei Alternative und Schleife)

The screenshot shows the 'Activity Properties' dialog box with the 'Time Attributes' tab selected. The dialog is divided into several sections:

- if construct starts after:** A text input field containing '01.04.1998'.
- if construct ends after:** An empty text input field.
- if duration exceeds maximum duration:** A group of radio buttons with 'proceed' selected.
- Duration fields:** Three input fields for 'minimum duration', 'average duration', and 'maximum duration', each containing the value '2'.
- Bottom section:** Three empty input fields for 'start date', 'duration', and 'end date'.
- Buttons:** 'Ok' and 'Cancel' buttons at the bottom center.

**Abbildung 4.6:** Konstruktsicht

Bei der Auswahl einer Aktivität (z.B. aus der zeitlichen Sicht) wird kontextabhängig der entsprechende Teil der Konstruktsicht (z.B. zeitliche Aspekte) vorausgewählt.

Über Eingabefelder und Dialoge können Anpassungen an den Konstrukten durchgeführt werden. Abb. 4.6 zeigt die Konstruktsicht mit dem Teilbereich für zeitliche Attribute. Hier können z.B. die minimale, maximale und durchschnittliche Ausführungszeit des Konstruktes durch Überschreiben der bisherigen Werte verändert werden.

## 4.6 Zusammenfassung und Bewertung

### 4.6.1 Die Ergebnisse im Überblick

Zur Steigerung der Qualität von Geschäftsprozessen werden in Unternehmen zunehmend Workflow-Management-Systeme eingesetzt. Neben sozialen Akzeptanzproblemen stellte sich für die Einführung solcher Systeme als hinderlich heraus, daß viele Geschäftsprozesse unstrukturierte Elemente enthalten, die sich nicht oder nur mit erheblichem Aufwand vorab in einer Workflow-Sprache modellieren lassen und daß in Ausnahmefällen nicht ausreichend (z.B. durch manuelle Maßnahmen) regulierend eingegriffen werden kann.

Die Möglichkeit des Einsatzes von Workflow-Management-Systemen wird daher stark unter den Gesichtspunkten der Anpassungsfähigkeit und Flexibilität erwogen. Durch Anpassungsfähigkeit können Prozesse zur Laufzeit von Benutzern geändert werden, um auf veränderte Rahmenbedingungen reagieren zu können. Ferner können grob modellierte Prozesse mit unbekanntem Teilen zur Laufzeit ergänzt oder verfeinert werden. Auch eine erhöhte Flexibilität durch ein mächtiges Workflow-Modell kann dazu beitragen, Anpassungen (zumindest teilweise) zu vermeiden. Durch Bibliotheken von Zusatzaktivitäten und vordefinierten Anpassungen (wie Delegation oder Entscheidungspunkten) können die Anwender bei der Durchführung manueller Anpassungen unterstützt werden.

Die Prozeßmodellierung ist eine komplexe und schwierige Aufgabe, so daß manuelle Anpassungen an Workflows nicht unkritisch betrachtet werden sollten; so sind Auswirkungen, speziell von zusammengesetzten Anpassungen nicht immer einfach zu überblicken. Andererseits kann dieselbe Anpassung in der einen Vorgangsinstantz hilfreich und erwünscht sein, in einer anderen jedoch strikt unerwünscht sein und fatale Folgen haben. Aus diesem Grund muß jede manuelle Anpassung kontrolliert und der entsprechende Vorgang dadurch geschützt werden. Hierfür wurde das Workflow-Modell um entsprechende Anpassungsaspekte erweitert, in denen z.B. Rechte oder Bedingungen definiert werden können, die einem integrierten Dienst zur Anpassungskontrolle die Überprüfung von Modifikationen an einem Vorgang ermöglichen. Die Ergebnisse der Anpassungskontrolle erlauben in diesem Zusammenhang auch noch die automatische Ausführung zusätzlicher Aktionen durch das System, die notwendig sein können, um eine Anpassung tatsächlich durchführen zu können.

Das erweiterte Workflow-Management-System SWATS enthält Komponenten zur Realisierung von Anpassungsfähigkeit und Flexibilität. Diese wurden im Architekturmodell als zusätzliche serverseitige Workflow-Anpassungsschicht auf die klassische Workflow-Ausführungsschicht gesetzt. Dazu wurde ein flexibles, anpassungsfähiges Workflow-Modell entwickelt, deren Instanzen in einer neuen Komponente, dem Workflow Instance Manager verwaltet werden. Zusätzlich

wird durch diese Komponente eine Schnittstelle definiert, die einen definierten Satz von Anpassungen zur Verfügung stellt über die der Benutzer des Workflow-Editors in einfacher Weise Anpassungen auf aktiven Instanzen durchführen kann.

Die Integration der zusätzlichen Schicht erfolgte über Erweiterungen des User-Managers. Zum einen können die Komponenten der Zusatzschicht dadurch Einfluß auf Aktivitäten nehmen und zum anderen wird die Ausführung und Steuerung komplexer Kontrollkonstrukte durch den zugrundeliegenden Workflow-Ausführungsdienst ermöglicht.

## 4.6.2 Weitere Ansätze zur Flexibilisierung von WFMS

Bereits vor einigen Jahren wurde das Thema der Anpassungsfähigkeit von Workflow-Systeme adressiert. Die Einschränkungen der bestehenden Systeme wurden dabei ebenso erkannt wie das Potential und die Notwendigkeit zur Flexibilisierung der Prozesse. Die ersten Arbeiten im Bereich AWS fielen noch in diese Phase. In den letzten Jahren hat der Themenbereich international einen enormen Aufschwung erhalten und es wurde eine Vielzahl von Projekten ins Leben gerufen, die teilweise mit sehr unterschiedlichen Ansätzen versuchen, die bestehende Problematik zu lösen; dabei können einige Forschungsarbeiten in Deutschland als führend bezeichnet werden.

Die Arbeiten der frühen Phase können in zwei Teilbereiche gegliedert werden. Neben Ansätzen zur Flexibilisierung von Prozessen durch Versionsmanagement der Workflow-Definitionen [Amberg96] und zur dynamischen Änderbarkeit von Workflow-Instanzen [ReiDad98] [Weske98a] wurde dabei versucht, unstrukturierte Tätigkeiten durch die Anbindung gekapselter Groupware-Applikationen (Konferenzen, gemeinsames Editieren etc.) zu integrieren.

Die Ansätze fanden zwar Berücksichtigung in Forschungsprototypen, wurden jedoch - auch bis heute - leider selten in produktfähige Systeme umgesetzt [McCSar93] [NasHil94] oder auf bestehende Produkte übertragen.

In stark wirtschaftswissenschaftlich orientierten Arbeiten wurden die Eigenschaften von strukturierten bzw. unstrukturierten Prozessen untersucht [EllisEA95] [HageEA97] [Heilmann94].

Mittlerweile hat sich gezeigt, daß beide Ansätze isoliert betrachtet, die bestehenden Probleme nicht allgemeingültig lösen können. Daher werden derzeit verstärkt integrierte Ansätze untersucht - wie auch der hier vorgestellte Ansatz aus dem Forschungsbereich AWS -, die auf einer flexiblen Modellierung und Versionskontrolle basieren, wobei die Instanzen dann zur Laufzeit kontrolliert angepaßt werden können. Auch die Bedeutung benutzergerechter Anwender-Schnittstellen wird allgemein anerkannt und verstärkt berücksichtigt.

Auch die aktive Beschäftigung einiger bedeutender Workflow-Organisationen (WFMC, OMG, GI-Arbeitskreis Workflow etc.), sowie eine Vielzahl namhafter Veranstaltungen zu diesem Problembereich belegen nicht nur das reale Anwendungsinteresse, sondern auch die interessanten Fortschritte, die auf diesem Gebiet gemacht werden (z.B. [SieWes98]).



### 4.6.3 Bewertung und Ausblick

Der modulare Aufsatz einer eigenen Schicht, der die Erweiterung des WFMS hinsichtlich Flexibilität und Anpassungsfähigkeit erlaubt, ist eine sinnvolle Konzeption, gerade im Hinblick auf die Tatsache, daß auch andere als das in diesem Projekt verwendete WFMS keine oder nur rudimentäre Anpassungen an aktiven Instanzen erlauben. So kann dieses Verfahren mit relativ wenig Änderungen an der Workflow-Ausführungsschicht an andere Systeme angepaßt werden und erlaubt somit allen Systemen die Durchführung dynamischer Anpassungen wie mit diesem System.

Ein weiterer Vorteil der Entwicklungen liegt in der Tatsache, daß an einer Instanz Anpassungen in allen enthaltenen Aspekten durchführbar sind und diese nicht auf einfachste Modifikationen von Teilbereichen eingeschränkt werden, wie z.B. die Änderung eines Aktors. Durch die Einführung von Anpassungsrechten, deren Detaillierungsgrad von sehr fein bis sehr grob reicht, und der zusätzlich eingeführten Workflow-Typisierung können alle Anpassungen auch durch das System kontrolliert werden, was angesichts der vielfältigen Auswirkungen von Anpassungen ein erheblicher Vorteil ist. So können Auswirkungen von Modifikationen zum Teil auch abgeschätzt werden und durch das System entsprechend definierte Maßnahmen eingeleitet werden, die eine unreflektierte Anpassung verhindern, sie jedoch auch zulassen, wenn dies nach entsprechender Prüfung zugelassen wird. Die Anpassungen können somit sehr flexibel kontrolliert werden, wobei der Modellierungsaufwand mit der Komplexität der Anpassungssituationen bzw. der Rechtevergabe steigt.

Durch die Einführung einer systeminternen Kontrolle der Anpassungen können Änderungen von allen Benutzer durchgeführt werden, da die syntaktische und semantische Korrektheit der Workflows nicht durch unsachgemäße Änderungen zerstört werden kann. Dies erforderte bisher besonders legitimierte Personen mit hoch spezialisiertem Fachwissen über die Prozesse und die Prozeßmodellierung.

Eine Stärke der implementierten Komponenten liegt in der Verwendung von innovativen Techniken wie Java und CORBA, die es erlauben, das System ohne großen Aufwand in verteilten, heterogenen Umgebungen einzusetzen. Diese Stärke wird erkauf mit einer Einschränkung der Performance, da die Ausführungsgeschwindigkeit von Java-Applikationen/-Applets noch nicht ganz konkurrenzfähig ist. Es ist jedoch abzusehen, daß diese Probleme durch die Weiterentwicklung von Java in der Zukunft zumindest reduziert werden können.

Der strukturierte Aufbau des Modells erleichtert die Durchführung von Anpassungen sowie die Zuweisung von Anpassungsrechten und Typisierungen. In die Modellierung der Workflows für spezifische Engines müssen jedoch u.U. Zusatzinformationen eingebracht werden, die eine Trans-

formation in das Modell der Anpassungsschicht erlauben. Ebenso müssen die entsprechenden Schnittstellen - insbesondere die Anpassungsdienste - vom Produkt unterstützt bzw. angeboten werden.

In weiteren Arbeiten an dem System könnten z.B. noch die Bibliotheken mit vordefinierten Aktivitäten und Anpassungen erweitert werden. Sie sollten allgemein verwendbare Elemente beinhalten, die für möglichst unterschiedliche Prozesse und Anwendungsbereiche einsetzbar sind (wie z.B. "Information des Vorgesetzten"). Für einen Einsatz mit vielen gleichzeitigen Änderungen an Instanzen müßte darüber hinaus das Sperrkonzept verfeinert werden, das bisher sehr grobgranular ist.

Weitere Forschungsarbeiten bzgl. anpassungsfähiger WFMS sind in den Bereichen der Nachvollziehbarkeit von Änderungen, den Benutzerschnittstellen und den transaktionale Konzepten erforderlich.

## 4.7 Literatur

- [Amberg96] Amberg, Michael : Modelling Adaptive Workflows in Distributed Environments. In Workshop on Adaptive Workflow: Proc. of the first Int. Conf. on Practical Aspects of Knowledge Management (PAKM). Basel, October 1996.
- [Blum 98] Blum, Ekkehard : Reimplementierung eines Workflow Editors für das Internet : Diplomarbeit. Universität Stuttgart, Fakultät Informatik, Stuttgart 1998
- [EllisEA95] Ellis C., Keddara K., Rozenbert G.: Dynamic Changes Within Workflow Systems. In Proc. Conference on Organizational Computing Systems (COOCS), Milpitas, CA, 1995, pp. 10-22.
- [GeorEA95] Georgakopoulos D., Hornick M., Sheth A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, 3, pp. 119-153, 1995.
- [HageEA97] Hagemeyer J., Herrmann Th., Just-Hahn K., Striemer R. : (1997) Flexibilität bei Workflow-Management-Systemen. Software-Ergonomie 97, 1997, S. 179-190.
- [Heilma94] Heilmann, Heidi : Workflow Management: Integration von Organisation und Informationsverarbeitung. In HMD Theorie und Praxis der Wirtschaftsinformatik: Workflow Management, S. 8-21. Forkel Verlag, 1994.
- [Hofman98] Hofmann, Rainer : Entwicklung eines anpassungsfähigen Workflow-Modells für SWATS: Diplomarbeit Nr. 1573. Universität Stuttgart, Fakultät Informatik, 1998.
- [Jablon95a] Jablonski, Stefan : Workflow-Management-Systeme: Motivation, Modellierung, Architektur. In Informatik Spektrum Nr. 18, 1995, S. 13-24.
- [Jablon95b] Jablonski, Stefan : Workflow-Management-Systeme: Modellierung und Architektur. Thomson's aktuelle Tutorien, Band 9. Berlin: Thomson, 1995.

- [JaBöSc97] Jablonski S., Böhm M., Schulze W. : Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie. dpunkt, Heidelberg, 1997.
- [KelRom91] Kellner M., Rombach D.H. : Comparison of Software Process Description - Session Summary. In Proc. of the 6th Int. Software Process Workshop (ISPW), pp. 7-18. New York, IEEE, 1991.
- [LeymEA94] Leymann F., Altenhuber W. : Managing Business Processes as an Information Resource. IBM Systems Journal 33, 1994, 326-347.
- [MayKop95] Mayer R., Kopperger D. : Schwachstellenanalyse HU-Bau. PoliFlow-Projektbericht am Institut für Arbeitswissenschaften und Technologiemanagement. Universität Stuttgart, November 1995.
- [McC Sar93] McCarthy D., Sarin S. : Workflow and Transactions in InConcert. IEEE Bull. of the Tech. Comp. on Data Engineering, Vol. 16, No. 2, 1993.
- [NasHil94] Nastansky L., Hilpert W. : The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation. In Wolfinger B. (Hrsg.) : Innovation bei Rechen- und Kommunikationssystemen. Springer, Informatik Aktuell, 1994.
- [ReiDad97] Reichert M., Dadam P. : A Framework for Dynamic Changes in Workflow Management Systems. In Proc. of the 8th Int. Workshop on Database and Expert Systems Applications (DEXA). France, 1997.
- [ReiDad98] Reichert M., Dadam P. (1998): Supporting Dynamic Changes of Workflows Without Loosing Control. To appear in: Journal of Intelligent Information Systems, Special Issue on Workflow and Process Management, Vol 10, No 2, 1998.
- [Sheth96] Sheth A. et al. : Report from the NSF Workshop on Workflow and Process Automation in Information Systems. Technical Report UGA-CS-TR-96-003, University of Georgia, Athens, GA, 1996.
- [Sieber95] Siebert, Reiner : Verbundprojekt PoliFlow, Anpassungsfähige Workflow-Systeme, Zwischenbericht 6/95. Modellierung und Beschreibung anpassungsfähiger WFMS - Kurzfassung. Poliflow-Zwischenbericht, Universität Stuttgart, IPVR, Juni 1995
- [Sieber96] Siebert, Reiner : Adaptive Workflow for the German Public Administration. In Proc. of the 1st Int. Conf. on Practical Aspects of Knowledge Management (PAKM'96) : Workshop on Adaptive Workflow. Basel (Schweiz), 1996.
- [Sieber97a] Siebert, Reiner : Anpassungsfähige Workflows zur Unterstützung unstrukturierter Vorgänge. In EMISA Forum, Band 1/1998. Proceedings des EMISA-Fachgruppentreffens 1997: Workflow-Management-Systeme im Spannungsfeld einer Organisation. GI, 1997.
- [Sieber97b] Siebert, Reiner : Adaptive Workflows im Verbundprojekt PoliFlow. Tagungsband zur Fachtagung Workflow-Management in Geschäftsprozessen im Trend 2000. Schmalkalden, 1997.
- [Sieber98] Siebert Reiner : An Open Architecture For Adaptive Workflow Management Systems. In Proceedings of the 3rd biennial World Conference on Integrated Design and Process Technology (IDPT): Issues and Applications of Database Technology. SDPS, Austin (USA) 1998.

- [SieWes98] Siebert R., Weske M. : Flexibilität und Kooperation in Workflow-Management-Systemen: Tagungsband zum Workshop der D-CSCW 1998. Schriftenreihe Angewandte Mathematik und Informatik der Universität Münster, Band 18/98-1. Münster, 1998.
- [SiKiSo97] Siebert R., Kindler T, Soyez T. : Integrated Workflow and Telecooperation Support for the German Government. In Proc. of the ACM Symposium on Applied Computing (SAC). San Jose 1997.
- [Stary94] Stary, Christian : Interaktive Systeme - Software-Entwicklung und Software-Ergonomie. Vieweg Verlagsgesellschaft mbH, Braunschweig/Wiesbaden 1994.
- [WainEA96] Wainer J., Weske M., Vossen G. Bauzer-Medeiros C.M. : Scientific workflow systems. In Proc. of the NSF Workshop on Workflow and Process Automation in Information Systems. Georgia, 1996.
- [Weske 98a] Weske, Mathias (1998): Flexible Modeling and Execution of Workflow Activities. In Proceedings of 31st Hawai'i International Conference on System Sciences, Software Technology Track (Vol VII), 713-722. IEEE Computer Society Press, 1998.
- [Weske 98b] Weske, Mathias : State-based Modeling of Flexible Workflow Executions in Distributed Environments. accepted for: International Workshop on Issues and Applications of Database Technology (IADT'98), Berlin, July 1998.
- [WolRei96] Wolf M., Reimer U. (Hrsg.) : Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM). Basel, 1996.

## 5 Sicherheit in WFMS (SWS)

Geschäftsvorgänge, in Behörden und Unternehmen, stellen fast ausnahmslos Anforderungen an die Sicherheit der Daten und des Datenzugriffs. Diese sicherheitsrelevanten Anforderungen können im allgemeinen bei der Geschäftsprozeßanalyse erfaßt werden. Heutzutage ist eine direkte Modellierung dieser Anforderungen in Workflow-Definitionen nicht oder nur unzureichend möglich. Bestehende Workflow-Management-Systeme bieten in ihren Modellen keine Mittel zur Umsetzung der Anforderungen, daher können die sicherheitsrelevanten Anforderungen nicht bei der Workflow-Definition berücksichtigt werden. Eine Umsetzung der Anforderungen außerhalb des Workflow-Management-Systems wird dadurch notwendig, dies erschwert die Sicherheitsadministration der gesamten DV-Infrastruktur erheblich, da die Querbezüge zwischen den Geschäftsvorgängen und der Sicherheitsadministration manuell verwaltet werden müssen. Dies ist ein unbefriedigender Zustand, dieses Ergebnis ergab auch die Technologie Evaluierung im Rahmen des PoliFlow Projekts (siehe Kapitel 8). Der Fokus des Arbeitsbereichs "Sichere Workflow Systeme" liegt somit auf der Erweiterung bestehender Workflow-Management-Systeme um den Sicherheitsaspekt. Im folgenden wird zunächst auf die Frage des Bedarfs an IT-Sicherheit, Sicherheitsstandards und der Sicherheit in Workflows eingegangen. Darauf aufbauend werden die Ergebnisse des Arbeitsbereichs "Sichere Workflow Systeme" dargestellt.

### 5.1 IT-Sicherheit und Workflow-Management

Die zunehmende Zahl der Berichte über Bedrohungen durch Viren, Hackern und von ihnen verursachte Schäden haben zu einer Sensibilisierung der Öffentlichkeit in Punkto IT-Sicherheit geführt. Tatsächlich sind all jene von Angriffen bedroht, die auf reibungslos funktionierende IT-Infrastrukturen zur Umsetzung ihrer Geschäftsprozesse angewiesen sind. Dies waren bisher meist Unternehmen der Wirtschaft, also Banken, Versicherungen und Handelsunternehmen [Glei89], [Ker95]. Durch die zunehmend automatisierte Steuerung (weitverteilter) Geschäftsprozesse in öffentlichen Einrichtungen und Behörden geraten aber auch diese zunehmend in das Schußfeld potentieller Angreifer. Als solche sind in erster Linie die Benutzer der IT Systeme anzusehen, die nicht selten ideale Voraussetzungen für erfolgreiche Angriffe vorfinden. Diese Erkenntnis unterstreicht die Forderung, daß - vor allem bei weitverteilten Geschäftsprozessen - Funktionalität und Sicherheit Hand in Hand gehen müssen. Nur so kann die bisherige Korrektheit und Qualität der Geschäftsprozesse auch bei automatischer Abwicklung garantiert werden.

## 5.1.1 Motivation zur IT-Sicherheit

Der zunehmende Einsatz von IT-Systemen in Unternehmen und Behörden, sowie deren rapide zunehmende Vernetzung, erhöhen die Effizienz und Qualität der Geschäftsprozesse. Gleichzeitig aber werden die Systeme und ihre Daten zunehmenden Bedrohungen ausgesetzt [Ker95], [Lin96]. Obwohl diese Tatsache allgemein bekannt ist, entspricht die Lage in der anbrechenden Informationsgesellschaft, trotz ausgereifter Sicherheitsstandards, selten den Erfordernissen. Somit besteht in der Praxis meist ein beträchtlicher Nachholbedarf in Punkto IT-Sicherheit [Ker95], [Fumy95].

Ein Grund für dafür ist, daß Sicherheit in der IT oft unterbewertet wird, und augenscheinlich mit hohem Aufwand und Kosten verbunden ist. Dabei wird jedoch vergessen, daß der nachträgliche Einbau von Sicherheitsfunktionen unvermeidbar ist, hohe zusätzliche Kosten verursacht und meist Lücken offen läßt [Eck95]. Ein weiterer Grund ist die lückenhafte Gesetzgebung in diesem Bereich. Dies gilt vor allem für externe Netzwerke, da sie nicht der direkten Kontrolle der Benutzer unterstehen, und zusätzliche Bedrohungen durch Außenstehende hinzukommen. Nicht zuletzt ist aber auch der hohe Kenntnisstand der Benutzer eine wichtige Voraussetzung für die ständig zunehmende Zahl erfolgreicher Angriffe auf IT-Systeme [Ker95], [Eck95], [Lin96]. Über den Erfolg solcher Angriffe lassen sich betroffene Unternehmen selten aus. Verständlich, ein ideeller Schaden könnte weitaus schwerer wiegen als ein direkt entstandener Schaden [Ker95], [Cer97].

## 5.1.2 Grundlagen der IT-Sicherheit

Wie in vielen anderen Bereichen der IT werden in der IT Sicherheit - trotz nationaler und internationaler Standardisierungsbemühungen - Fachbegriffe oft unterschiedlich definiert und verwendet. Um für die folgenden Überlegungen einen eindeutigen Hintergrund zu schaffen, werden hier zunächst die wichtigsten Begriffe zur IT-Sicherheit festgelegt und erläutert.

### **Ordnungsmäßigkeit und Sicherheit, Bedrohungen**

Der Begriff Ordnungsmäßigkeit (*safety*) beschreibt den korrekten, geplanten Ablauf der Prozesse eines IT Systems. Damit umfaßt Ordnungsmäßigkeit auch die Garantie der Funktionen, deren Qualität und die bereits angesprochene Rechtssicherheit [Ker95].

Unter IT-Sicherheit (*security*) verstehen man den Schutz von IT Systemen vor Angriffen auf das System selbst, darin ablaufende Prozesse, gespeicherte Daten und auf die Benutzer. Dabei ergeben sich jedoch Überschneidungen mit der Ordnungsmäßigkeit, so z.B bei Angriffen gegen die Funktionalität, bei der auch häufig Prozesse oder Benutzer betroffen sind. Die Auswirkungen solcher Angriffe lassen sich meist durch organisatorische und konstruktive Maßnahmen (z.B. funk-

tionelle Redundanz, Vorhalten von Ressourcen) verhindern oder zumindest einschränken [Beut90], [Ker95]. Derartige Angriffe werden deshalb im folgenden der Ordnungsmäßigkeit zugeordnet.

Das Gegenstück zum Begriff Sicherheit ist die Bedrohung (*threat*), worunter die Gefahr der Einschränkung oder des Verlusts von Sicherheitseigenschaften zu verstehen ist [Ker95]. Wegen der Überschneidung zwischen Ordnungsmäßigkeit und Sicherheit ist dabei eine Einteilung in Bedrohungstypen angebracht: a) Bedrohungen, die auf technisches Versagen oder höhere Gewalt zurückzuführen sind bzw. konzeptuelle (systemimmanente) Ursachen haben, können meist durch Maßnahmen zur Ordnungsmäßigkeit nahezu ausgeschlossen werden; b) Bedrohungen, die von Angreifern (Viren, Softwarefehler, Benutzer, Kommunikationspartner oder Außenstehende) ausgehen bedürfen i.d.R. spezieller Sicherheitsmaßnahmen [Beut90], [Ker95], [Pern97a]. Im folgenden soll nur letzterer Bedrohungstyp als Bedrohung der Sicherheit i.e.S. angesehen werden.

### **Sicherheit - Ziele, Anforderungen und Politiken**

Sicherheitsziele sind angestrebte Teilaspekte der IT-Sicherheit. In der Praxis unterscheidet man i.a. zwischen den Grundzielen eines IT-Systems (Vertraulichkeit, Integrität, Verfügbarkeit) und sonstigen Sicherheitszielen (Anonymität, Pseudonymität, Zuweisbarkeit, Verbindlichkeit, etc.). Letztere spielen vor allem in offenen Kommunikationssystemen eine wichtige Rolle [Eck95], [Fumy95]. Bemerkenswert ist, daß eine Verletzung der Sicherheitsziele oft unterschiedlich schnell erkannt wird. So fällt eine eingeschränkte Verfügbarkeit sehr schnell, verletzte Integrität eher spät, und ein Verlust der Vertraulichkeit oder Anonymität oft überhaupt nicht auf [Ker95].

Konkrete Ausprägungen der Sicherheitsziele werden in Form von Sicherheitsanforderungen spezifiziert. Relevante Sicherheitsziele, einschließlich der Anforderungen daran, werden zu Sicherheitsprofilen zusammengefaßt, die den spezifischen Bedarf eines IT-Systems an Sicherheit repräsentieren. Dieser Bedarf ist im staatlich-militärischen Bereich meist sehr hoch, in der freien Wirtschaft variabel, und im privaten Bereich (ohne Netzanbindung) dagegen eher gering [Ker95].

Der Sicherheitsbedarf eines IT-Systems orientiert sich an den mehr oder weniger strikt vorgegebenen Kausalitäten einer Organisation. Diese manifestieren sich i.d.R. in unternehmensspezifischen Sicherheitspolitiken, die die Regeln zur Auswahl der Maßnahmen (Sicherheitsdienste) und den Einsatz der Mittel (Sicherheitsmechanismen) zur Umsetzung der Sicherheit festlegen [Eck95].

Als Grundfunktionen sicherer IT-Systeme werden die Sicherheitsdienste zur Identifikation, Authentifikation, Autorisierung, Beweissicherung, Wiederaufbereitung, Verfügbarkeit und zur Sicherung der Kommunikation angesehen (Vgl. [Ker95]). Die dabei typischerweise eingesetzten Sicherheitsmechanismen sind Mechanismen zur Identifikation, Authentifikation, Verschlüsselung, Integritätssicherung und zur Beglaubigung (Notariat) [Eck95], [Lin96]. Im folgenden steht der Sicherheitsdienst Zugriffskontrolle (Autorisierung) im Mittelpunkt der Betrachtungen. Er wird durch Sicherheitsmechanismen zur Rechteprüfung und Rechtevergabe realisiert.

### 5.1.3 Sicherheitsstandards der IT

Seit den Anfängen der IT wurde der Sicherheit, vor allem in staatlichen und militärischen Systemen, ein hoher Stellenwert eingeräumt. Dies spiegelt sich in zahlreichen Bestrebungen, die Terminologie und Konzepte der IT-Sicherheit zu standardisieren und weiterzuentwickeln [Fumy95]. Seit Anfang der 80er Jahre entstand so eine große Zahl Kriterienwerke, wovon die TCSEC<sup>1</sup> des us-amerikanischen NCSC, die ITSK<sup>2</sup> des deutschen BSI, die europäisch harmonisierten ITSEC<sup>3</sup> und die amerikanisch-europäisch harmonisierten CC<sup>4</sup> die wichtigsten darstellen.

Diese Standards werden als Meilensteine der IT-Sicherheit angesehen, deren angemessene Berücksichtigung die Flexibilität, umfassende Funktion und Interoperabilität sicherer IT-Systeme und Produkte garantiert. Der größte Verdienst dieser Standards jedoch ist, daß die Transparenz und Akzeptanz sowohl auf Betreiber, als auch auf Benutzerseite gefördert wird. Dieser Punkt entscheidet - so trivial er auch sein mag- letztlich über den Einsatz sicherer IT-Systeme [Ker95], [Ran95].

#### Übersicht wichtiger Sicherheitsstandards

Die 1983 veröffentlichten TCSEC (*Orange Book*) des us-amerikanischen NCSC sind einer der ersten Standards zur IT-Sicherheit für IT-Systeme im militärischen oder staatlichen Einsatz. Bereits kurz nach Veröffentlichung unterlag er jedoch heftiger Kritik: Die starre Kopplung von Funktionalität und Qualität in vier Sicherheitsstufen ist zu unflexibel für praktische Erfordernisse. Dabei steht die Vertraulichkeit im Mittelpunkt, andere Sicherheitsziele werden nur am Rande behandelt. Problematisch ist auch der zeit- und kostenintensive Evaluierungsprozeß, der nur durch staatliche Stellen durchgeführt werden darf. Diese und weitere Kritikpunkte führten zu einer mangelnden Akzeptanz seitens der Wirtschaft, aber auch seitens der Behörden und sogar des Militärs [Ker95], [Ran95], [Fumy95]. Die TCSEC wurden 1992, trotz zahlreicher Ergänzungen, durch den deutlich verbesserten nationalen Standard *Federal Criteria* abgelöst.

Im Jahr 1989 veröffentlichte das heutige BSI den deutlich weiterentwickelten nationalen Sicherheitsstandard IT-SK (Grünbuch). Darin werden Funktionalität und Qualität auf 2 Dimensionen verteilt, und unter mehreren Aspekten betrachtet. Sicherheits- und Funktionsprofile können spezifisch festgelegt werden, was die Flexibilität beträchtlich erhöht. Eine Evaluierung kann von kommerzieller Seite marktgerecht durchgeführt werden [Ker95], [Fumy95]. Begrüßenswert ist

<sup>1</sup>*Trusted Computer Security Evaluation Criteria* - hrsg. vom National Computer Security Center (USA)

<sup>2</sup>*IT-Sicherheitskriterien* - hrsg. durch das Bundesamt für Sicherheit in der IT (Deutschland)

<sup>3</sup>*IT-Security Evaluation Criteria* - EG-weit verabschiedet von Deutschland, Frankreich, Großbritannien u.a.

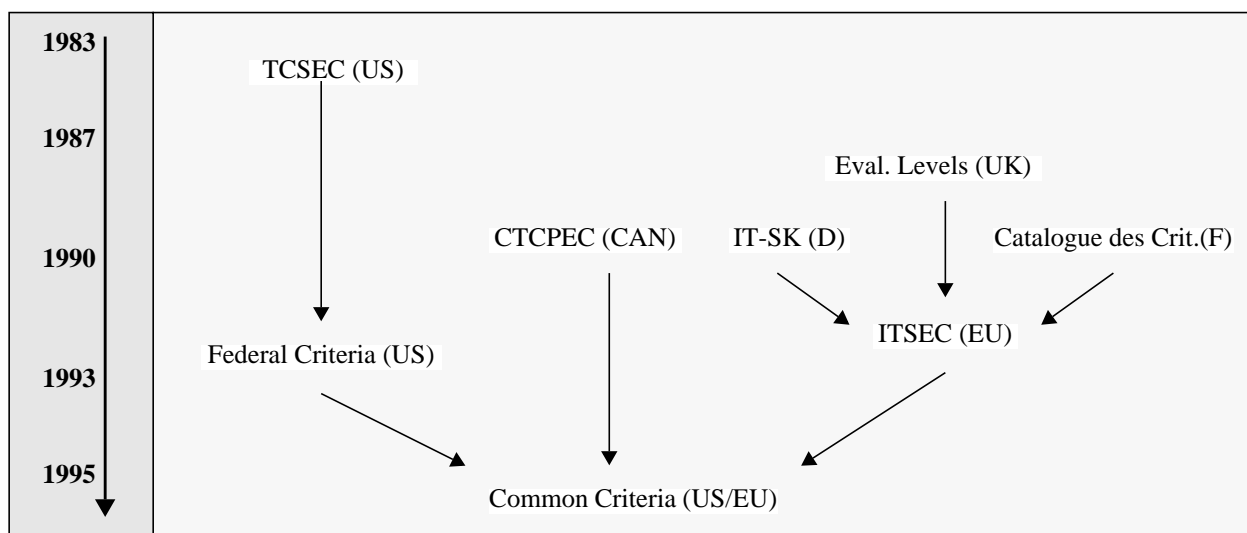
<sup>4</sup>*Common Criteria* - verabschiedet von USA, Kanada und zahlreichen europäischen Staaten



zudem die ausgewogene Berücksichtigung aller Sicherheitsziele, insbesondere der Kommunikationssicherung. Diese wurde in Anlehnung an das Security-Addendum OSI-7498/2 festgelegt [Pom91].

Bereits in den niedrigen Funktionsklassen F1 bis F5, vergleichbar den Klassen der TCSEC, werden alle Grundfunktionen mit zunehmender Stärke berücksichtigt. Ab F4 / F5 werden Rollen vorgesehen, um einerseits Zugriffsrechte effizient zu verwalten, und andererseits die Trennung von Kompetenzen zu ermöglichen. In den höheren Klassen F6 - F10 kommen Konzepte für spezielle IT-Systeme hinzu, wie z.B. typisierte Objekte und - Zugriffsrechte, oder abstrakte Zugriffsmechanismen (Views, Filter) - auch im Kontext offener Kommunikationssysteme [Ker95].

Die 1991 verabschiedeten ITSEC sind ein aus europäischen Standards hervorgegangener harmonisierter Sicherheitsstandard. Dabei wurden die Grundfunktionen der IT-SK weitgehend übernommen und z.T. erweitert. In der EU war er zeitweilig defacto Standard, nach dem insgesamt eine 3-stellige Zahl IT-Produkte evaluiert wurde. Mit ihm können Sicherheitsvorgaben noch flexibler (produktspezifisch) festgelegt, und sogar weitere Standards eingebunden werden [Ker95], [Fum95]. Die Vorgaben zur Funktionalität orientieren sich an der Systematik der IT-SK, auch die Bewertung der Qualität wird sehr ähnlich vorgenommen. Obwohl Kriterien und Prüfverfahren zur Zertifizierung 1993 EG-weit durchgesetzt wurden, gab es z.T. Probleme bei der Anerkennung von Zertifikaten, was auf nicht harmonisierte Rechtssprechungen zurückzuführen ist [Ker95].



**Abbildung 5.1:** Nationale und internationale Sicherheitskriterien

Die Ende 1997 verabschiedeten CC stellen eine erzwungene Harmonisierung europäischer, amerikanischer und kanadischer Standards dar, motiviert durch eine gegenseitige Nichtanerkennung. In den CC werden sehr unterschiedliche Strukturierungen und Methodiken vermengt, was zu sehr komplexen Kriterien führt, die allenfalls noch von Spezialisten verstanden und umgesetzt werden

können [Ker95], [Ran95]. Dennoch gibt es auch hier zahlreiche positive Neuerungen, so lassen sich z.B. (sinnige) Sicherheitsprofile aus einzelnen Bausteinen zusammensetzen, außerdem werden auch sicherheitsbezogene Anforderungen an die Infrastruktur gestellt [Ker95].

### **Kritik der bisherigen Sicherheitsstandards**

Die stetigen Entwicklungen (vgl. Abb. 5.1) auf dem Gebiet der Sicherheitsstandards führten zu einem umfassenderen und tieferen Verständnis der IT-Sicherheit. Dennoch enthalten selbst neuere Standards noch zahlreiche Kritikpunkte die der Verbesserung bedürfen. Kritisiert werden vor allem der Entstehungsprozeß, die Aussagekraft der Zertifikate, der Aufwand zur Evaluierung und die Unausgewogenheit bei der Sicherheitsfunktionalität sowie den berücksichtigten Gruppen [Ran95].

Es ist absehbar, daß in künftigen Kriterienwerken vor allem letzterem Kritikpunkt begegnet werden muß. Bisher standen meist nur die Interessen von Besitzern und Betreibern der IT-Systeme im Vordergrund, was einer Einseitigen Sicherheit entspricht. Mit einem zunehmenden Sicherheitsbewußtsein und der Forderung nach informationeller Selbstbestimmung müssen zukünftig auch die Sicherheitsinteressen anderer Parteien beachtet werden [Ker95], [Ran95]. Dies ist eine unabdingbare Voraussetzung für die ausreichende Akzeptanz automatisierter Geschäftsprozesse.

## **5.1.4 Sicherheit in interorganisationellen Workflows**

Interorganisationelle Workflows sind automatisierte Geschäftsprozesse, bei denen Akteuren verschiedener Organisationen kooperativ auf ein gemeinsames Ziel hin arbeiten. Dabei wird häufig über Kommunikationssysteme auch auf Daten und Ressourcen anderer Organisationen zugegriffen, wobei dem Sicherheitsaspekt eine besonders hoher Stellenwert zukommt [Pern97b]. Für die kommenden Jahre wird erwartet, daß interorganisationelle Workflows zu grundlegend veränderten Interaktionen mit Banken, Wirtschaftsunternehmen und Behörden führen werden. Dadurch wird zwar die Lebensqualität aller verbessert, gleichzeitig aber entsteht eine zunehmende Abhängigkeit von der Ordnungsmäßigkeit und Sicherheit der IT-Infrastrukturen [Pern97b]. Zu diesen zählen, neben den IT-Systemen, auch WfMS<sup>1</sup> und Kommunikationssysteme. Um eine breite Akzeptanz solcher Workflows zu erreichen, müssen gewisse grundlegende Anforderungen erfüllt werden. Dazu gehören eine mehrseitige Sicherheit, der Einbezug organisatorischer und globaler Kausalitäten, sowie einige spezielle Aspekte, die die Umsetzung der Sicherheit betreffen.

---

<sup>1</sup>Workflow-Management-System - System zur automatisierten Steuerung von Geschäftsprozessen

### **Mehrseitige Sicherheit**

Unter Mehrseitiger Sicherheit versteht man die Berücksichtigung der Sicherheitsinteressen mehrere (aller) Parteien, die an einem Prozeß beteiligt sind. In einem interorganisationellem Workflow wären dazu folgende Gruppen bzw. Rollen einzubeziehen (Vgl. [Ker95], [SSON96]):

- Besitzer / Betreiber von IT-Systemen - diese sind zwar oft identisch, doch werden zunehmend auch spezielle Dienste externer Anbieter in Anspruch genommen. So z.B. Einrichtung, Betrieb und Wartung einer Website als *Point of Service* oder *Point of Sale*.
- Besitzer / Betreiber von Kommunikationssystemen - diese fallen i.d.R. zusammen.
- Benutzer der IT-Systeme - sind entweder selbst Besitzer/Betreiber, oder lassen sich von diesen die Einhaltung ihre Sicherheitsinteressen vertraglich zusichern.
- Workflows - ihre Ausführung ist nicht selten mit gewissen übergeordneten Sicherheitsanforderungen verbunden, impliziert durch globale Ziele der beteiligten Parteien. Beispiele dafür wären die Trennung von Kompetenzen oder die Beschränkung von Privilegien.

Bei der Ausführung eines Workflows kann durchaus die Situation eintreten, daß sich einzelne Sicherheitsinteressen der Beteiligten widersprechen oder nicht eingehalten werden können. Da eine Ausführung jedoch im Interesse aller liegt, kann dabei ggf. ein Kompromiß eingegangen werden. Dabei dürfen aber keinesfalls globale Sicherheitsanforderungen des Workflows verletzt, oder den Sicherheitsinteressen Einzelner grob zuwider gehandelt werden [Pat98].

### **Organisatorische und Globale Kausalitäten**

Die **Kausalitäten einer Organisation** (Unternehmenskausalitäten) die in Geschäftsprozessen beachtet werden müssen manifestieren sich in organisationsspezifischen Sicherheitspolitiken. Diese müssen bei einer Umsetzung der Sicherheit grundsätzlich berücksichtigt werden, wobei die Autonomie des Einzelnen - auch bei Konflikten mit den Sicherheitsinteressen Andererer - unbedingt erhalten bleiben muß. Dies bedeutet einerseits, daß eine Aktivität, bei der den Kausalitäten einer Organisation zuwider gehandelt würde nicht ausgeführt werden darf. Andererseits muß jede derart betroffene Organisation die Freiheit besitzen, ihre Kausalitäten kurzzeitig auszusetzen, um die Aktivität dennoch frei zu geben [Pat98].

Aus Sicht eines interorganisationellen Workflows können auch gewisse globale Kausalitäten existieren, die sich z.B. aus allgemeingültigen (üblichen) organisatorischen Kausalitäten ableiten lassen. Bei der Umsetzung der Sicherheit müssen diese ebenfalls berücksichtigt werden. Was die Autonomie eines Workflows bei deren Verletzung angeht, gilt auch hier das bereits oben gesagte. Dabei fällt die Entscheidung über eine Aussetzung der Kausalitäten einer vertrauenswürdigen und kompetenten Domäne zu, z.B. dem Koordinator der Workflows, dem WfMS [Pat98].

### Umsetzungsaspekte der Sicherheit

Bei der Ausführung interorganisationeller Workflows müssen lokale Informationen und Ressourcen Außenstehenden zugänglich gemacht werden. Dabei sind gewisse grundlegenden Prinzipien der IT-Sicherheit zu beachten, wie z.B. das der *least privileges* [Same95]. Dieses besagt, daß immer nur die unbedingt erforderlichen (minimalen) Privilegien vergeben werden dürfen. Übertragen auf die vierte Dimension bedeutet dies, im Kontext der Zugriffskontrolle in Workflows, daß Zugriffsrechte an die Ausführung von Aktivitäten gekoppelt werden müssen. Praktisch kann dies z.B. erreicht werden, indem alle zu einer Aktivität benötigten Zugriffsrechte zu deren Beginn vergeben, und bei deren Ende wieder entzogen werden.

Bei der Mehrseitigen Sicherheit wurde bereits angesprochen, daß sich Sicherheitsinteressen der Beteiligten eines interorganisationellen Workflows u.U. widersprechen oder unerfüllbar sind. In diesem Fall muß es möglich sein, den Widerspruch durch einen Kompromiß zu bereinigen, da sonst mit Verklemmungen zu rechnen wäre. Aus ähnlichen Gründen wie bei den organisatorischen und globalen Kausalitäten sollte auch diese Kompetenz dem WfMS zufallen, da es eine globale Sicht besitzt und angemessen reagieren kann. Als Grundlage solcher Kompromisse sind definitive Absprachen mit den Beteiligten erforderlich, die in Umsetzungsverträgen mit dem WfMS festgehalten werden müssen. In diesen Verträgen kann auch eine Garantie zur Einhaltung der organisatorischen Kausalitäten festgeschrieben werden [Kind97], [Pat98].

Ein Kompromiß zur Ausführung einer Aktivität trotz verletzter Sicherheitsinteressen kann auf unterschiedlicher Basis realisiert werden. Zum einen kann ein WfMS dazu, entsprechende Umsetzungsverträge vorausgesetzt, einen gewissen Ermessenspielraum nutzen um sich darüber hinwegzusetzen [Pat98]. Zum anderen können zusätzliche Sicherheitsdienste und -mechanismen einbezogen werden, um die Interaktionen im Rahmen der Aktivität unter sichereren Rahmenbedingungen durchzuführen. So könnten z.B. bei zu geringer Vertraulichkeit Verschlüsselungsdienste partiell angewandt (Anonymisierung), oder bei mangelnder Integrität die Prüfung von Integritätsregeln durchgeführt werden. Die grundlegende Idee dazu und eine zugehörige Schichtenarchitektur ist in [Pern97a] zu finden, worauf im nächsten Kapitel noch einmal eingegangen wird.

## 5.2 Lösungsansatz in PoliFlow

Das Stuttgarter Workflow- und Telekooperations System (SWATS) ist ein prototypisches WfMS zur Abwicklung und Unterstützung weitverteilter Geschäftsprozesse, das in PoliFlow entwickelt wurde [Poli98a]. Die Basisfunktionen des Systems sind bereits weitgehend realisiert, wodurch die Ankopplung von Sicherheitsfunktionen und die erreichbare Sicherheit zu einem gewissen Maße

vorgegeben sind. Bevor wir jedoch darauf eingehen, soll zunächst ein Einblick in den Stand der Sicherheit im Workflow-Management gegeben, und ein Grundansatz zur Umsetzung der Sicherheit entwickelt werden. Danach wird die Basisarchitektur des SWATS, sowie die dazu entwickelte Sicherheitsarchitektur erörtert, um schließlich auf die Konzepte ihrer Realisierung einzugehen.

## 5.2.1 Sicherheit in Workflow-Management-Systemen

Sicherheit in interorganisationellen Workflows beruhte bisher fast ausschließlich auf den Mitteln und Maßnahmen, die in den beteiligten IT-Systemen zur lokalen Sicherheit eingesetzt wurden [Pern97a]. Bei der Ausführung solcher Workflows stehen jedoch eher übergeordnete Sicherheitsinteressen im Vordergrund, weswegen die bisherige Vorgehensweise der Problematik nicht gerecht wird. Eine angemessene Sicherheit in interorganisationellen Workflows erfordert eine globale Perspektive, wobei von den lokalen Sicherheitspolitiken, aber auch von den vorhandenen Sicherheitsdiensten und -mechanismen weitgehend abstrahiert werden sollte [Kind97].

### Sicherheit in IT-Systemen

Bei der Umsetzung lokaler Sicherheit spielen die Dienste der eingesetzten Betriebssysteme eine fundamentale Rolle. Ein kurzer Einblick in deren Sicherheitsdienste soll zeigen, das gerade hier noch zahlreiche Schwachstellen, ja sogar Lücken vorhanden sind. Da derzeit eine große Zahl Betriebssysteme existiert, wurde hierzu exemplarisch Unix und dessen Derivate ausgesucht.

Das *multiuser / multitasking* Betriebssystem Unix entstand in den frühen Anfängen der IT im universitären Umfeld. Es wird seit den 80er Jahren zunehmend auch in Behörden und Unternehmen eingesetzt, wodurch der Sicherheitsaspekt ständig an Bedeutung gewinnt. Unter Fachleuten gilt es als sehr stabil und relativ sicher. Dennoch besitzt es zahlreiche Schwachstellen, auch auf konzeptioneller Ebene, die es ohne Erweiterungen äußerst verwundbar machen [Eck95]:

- Identifikation - wird nur durch Abfrage einer (statischen) Benutzerkennung realisiert. Eine solche kann auch von Außenstehenden ohne großen Aufwand ermittelt werden, und wird meist nach den Eigennamen der Benutzer festgelegt.
- Authentifikation - wird durch ein (einweg-verschlüsseltes) statisches Passwort realisiert. Die Schwachstelle hierbei ist eine Passwort-Datei, in der die verschlüsselten Passwörter aller Benutzer gesammelt werden. Gelingt der Zugriff auf diese Datei, können mit einem der zahlreichen frei verfügbaren Crackprogramme Passwörter (indirekt) entschlüsselt werden.

- Zugriffskontrolle - wird mit Zugriffskontrolllisten durchgeführt, die vor Zugriffen zur Prüfung herangezogen werden. Dies geschieht jedoch nur beim Öffnen einer Datei. Diese kann danach beliebig lange offen gehalten werden, auch wenn die Rechte eigentlich wieder entzogen wurden. Problematisch sind auch die uneingeschränkten Zugriffsrechte der *Superuser* und das *Ownership* Konzept, die den Benutzern zu viele Freiheiten lassen.
- Wiederaufbereitung - hierfür sind in Unix keinerlei Dienste oder Funktionen vorgesehen. Damit sind sogenannten verdeckte Kanäle Tür und Tor geöffnet.
- Kommunikationssicherheit - stellt hier ebenfalls eine Schwachstelle dar. Kommunikation erfolgt meist nach dem TCP/IP Protokoll, wobei die nicht verschlüsselten Datenpakete mühelos abgehört und (unbeobachtet) ausgewertet werden können. Auch das *Trusted Host* Konzept ist nicht unproblematisch, da diese Eigenschaft auch vorgespiegelt werden kann.

Diese Liste könnte noch weiter ausgeführt werden, was aber den vorgegebene Rahmen sprengen würde. Doch auch so ist bereits eines gezeigt, nämlich daß die Sicherheitsdienste und -mechanismen eines IT-Systems u.U. noch nicht einmal die lokale Sicherheit garantieren können. Als Konsequenz dieser Erkenntnis sollte, für interorganisationelle Workflows, das WfMS dahingehend erweitert werden, daß es deren sichere Ausführung unterstützen kann. Dies betrifft vor allem den globalen Aspekt bei Zugriffen auf Informationen und Ressourcen durch fremde Organisationen.

### **Sicherheit in derzeitigen WfMS**

Im Bereich des WfM wurde der IT-Sicherheit bisher noch weniger Bedeutung zugemessen als dies bei Betriebssystemen der Fall ist. Dies ist zu einem gewissen Maße verständlich, da es sich um eine noch relativ junge Technologie handelt. Dennoch ist es bedenklich, daß selbst bei kommerziellen WfMS dem Sicherheitsaspekt offenbar sehr wenig Beachtung geschenkt wird [SSON96]:

*Work Party* ist ein WfMS der SNI AG, das bereits auch zur Steuerung interorganisationeller Workflows eingesetzt wurde. Zur Gewährleistung der Vertraulichkeit werden hier Verschlüsselungsverfahren eingesetzt, was jedoch bei Überschreiten von Landesgrenzen zu rechtlichen Problemen führt. Weitere Informationen zur Sicherheit wurden bisher leider noch nicht publik.

*ProMInanD* ist ein WfMS der IABG, das Rollenkonzepte verwendet um die Sicherheitsproblematik anzugehen. Hier werden einzelnen Aktivitäten eines Workflows Rollen zugewiesen, um damit den Kreis der Ausführenden einzuschränken. Zudem können in einer Organisationsdatenbank auch einzelne Rollenkombinationen festgelegt bzw. ausgeschlossen werden.

Ein prototypisches System ist das *BPAFrame* der TU Dresden, das massiv auf die CORBA Technologie aufsetzt. Bisher ist aber außer der Sicherung der Datenbank mit den Workflow-Beschreibungen und administratorischen Informationen keinerlei Sicherheitsfunktionalität enthalten. Für eine neue Version ist zwar vorgesehen, zukünftige CORBA-Security Services einzubeziehen, was aber gewisse Abhängigkeiten schafft, und mit hoher Wahrscheinlichkeit auch Lücken offen läßt.

*CodAlf* ist ein weiteres prototypisches System der TU Dresden, das auf dem Trader-Konzept aufsetzt. Ähnlich wie im letzten Beispiel existieren jedoch auch hier Abhängigkeiten zu vorhandenen Sicherheitsdiensten des OSF/DCE. Als neuer Ansatz werden hier teilweise Sicherheitsfunktionen durch die Trader realisiert. Zudem existiert die Möglichkeit, den Ausführungsstatus eines Workflows einzusehen und zu beeinflussen, entsprechend den Kompetenzen der eingenommenen Rolle.

## 5.2.2 Grundansatz zur Umsetzung der Sicherheit

Sicherheit in verteilten Umgebungen war bisher oft eine *Ende-zu-Ende* Angelegenheit, die durch Applikationen gewährleistet wurde [Pern97a]. Dies scheint im Kontext einzelner Applikationen und Benutzer gerechtfertigt, da diese meist unabhängig sind und unterschiedliche Sicherheitsprofile besitzen. Im Kontext interorganisationeller Workflows jedoch gerät das Sicherheitsinteresse einzelner Benutzer und Applikationen eher in den Hintergrund, wogegen eine globale Sicherheit an Bedeutung gewinnt. Diese repräsentiert ein übergeordnetes Sicherheitsinteresse, impliziert durch die globalen Ziele der Workflows, die letztlich von allen beteiligten Parteien angestrebt werden.

Eine solche Art der Sicherheit könnte z.B. vom integrierenden WfMS gewährleistet werden, sofern es geeignet erweitert wird. Zumindest aber sollte das WfMS an einer Umsetzung der Sicherheit beteiligt werden, da hierzu sehr detaillierte Informationen über den möglichen Ablauf und den Ausführungsstand der Workflows erforderlich sind [Pat98].

### Verteilungsaspekte einer Sicherheitsarchitektur

Nachdem nun fest steht, daß das WfMS an der Umsetzung der Sicherheit beteiligt sein muß, ist noch die Frage zu klären, welche Komponenten in einer verteilten Umgebung welche Aufgaben übernehmen. Diese Aufgaben sind die Speicherung, Verwaltung und Prüfung globaler Sicherheitsspezifikationen, sowie deren Umsetzung durch die lokalen Sicherheitsdienste und -mechanismen. Um diese Frage zu beantworten betrachten wir die Alternativen der horizontalen Verteilung, die Ansätze die diesen zu Grunde liegen, und die Vor- und Nachteile die sich daraus ergeben. Zwar steht dabei der Sicherheitsaspekt im Vordergrund, doch spielen in Workflows auch Effizienz, Flexibilität und Offenheit eine bedeutende Rolle (Vgl. [SSON96]), weshalb auch sie in die

Bewertung einbezogen werden. Im folgenden soll der Begriff Domäne anstelle von Beteiligter oder Partei verwendet werden, worunter ein autonomer Verwaltungsbereich mit eigener Sicherheitspolitik zu verstehen ist [Kind98].

### **Strategie 1** - dezentrale Sicherheitsfunktionen

Ansatz: Durch eine massive Verteilung sicherheitsbezogener Informationen und Funktionen sollen Angriffe unattraktiv und deren Auswirkungen lokal eingeschränkt werden.

- Sicherheit - ist hier sehr gering. Informationen über Workflows und Sicherheitsspezifikationen werden hier unnötig exponiert. Zudem hängt der erreichbare Grad an Sicherheit von den Gegebenheiten und der Verlässlichkeit der Domains ab. Auch scheinbar sichere Domains könnten so von unsicheren aus angegriffen werden (Vgl. "schwächstes Glied einer Kette"), zudem könnten Abhängigkeiten bei Ausfall auch zu Verklemmungen führen.
- Effizienz - ist hier sehr gut, da wenig Kommunikation anfällt. Damit verbunden ist jedoch eine hohe funktionelle und datenbezogene Redundanz, die bei häufigen Änderungen oder leistungsschwachen Domains dennoch zu einer schlechten Gesamteffizienz führen kann.
- Flexibilität - ist eher als schlecht einzustufen. Änderungen der Workflows oder Sicherheitsspezifikationen müssen allen in Frage kommenden Domains propagiert werden, wobei sowohl lokale als auch globale Probleme entstehen können (Versionsproblematik).
- Offenheit - ist als mittel einzuschätzen. Zwar können prinzipiell beliebige neue Domains einbezogen werden, wozu aber zahlreiche Absprachen zwischen den Domains erforderlich sind, um alle potentiellen interorganisationellen Interaktionen abzusichern.

### **Strategie 2** - zentrale Sicherheitsfunktionen

Ansatz: Durch eine Zentralisierung der sicherheitskritischen Informationen sollen diese einerseits effektiv geschützt und andererseits auch effizient umgesetzt werden.

- Sicherheit - ist hier sehr gut. Dennoch ist man zur Umsetzung letztlich auf die lokalen Gegebenheiten angewiesen, und das WfMS stellt dabei einen *single point of failure* dar.
- Effizienz - ist als mittel einzustufen. Zwar kann eine Gesamteffizienz durch eine angemessenen Auslegung verbessert werden, doch ist hier der Kommunikationsaufwand deutlich höher als bei der 1. Strategie. Bei sehr hoher Last besteht zudem die Gefahr eines *bottlenecks*.
- Flexibilität - ist hier sehr gut. Änderungen der Workflows oder Sicherheitsspezifikationen sind nach außen hin nicht sichtbar, also transparent. Dennoch können Änderungen prinzipiell Auswirkungen auf einzelnen Absprachen zwischen WfMS und Domains haben.
- Offenheit - ist hier sehr gut. Neue Domains können jederzeit (dynamisch) einbezogen werden, wozu nur eine Absprache zwischen der Domäne und dem WfMS erforderlich ist.



**Strategie 3** - zentrale Speicherung, Verwaltung und Prüfung - dezentrale Umsetzung.

Ansatz: Durch eine überlegte Kombination obiger Strategien sollen ihre Vorteile vereint werden.

- Sicherheit - wie bei der 2. Strategie. Um die Gefahren der 1. Strategie zu relativieren, sind vom WfMS ggf. noch zusätzliche Sicherheitsfunktionen lokal einzubringen.
- Effizienz - ist hier gut. Informationen können effizient verwaltet und geprüft werden, gleichzeitig hält sich die Kommunikation zur Umsetzung in Grenzen. Prüfungen die von lokalen Gegebenheiten abhängen müssen aber u.U. dennoch lokal durchgeführt werden.
- Flexibilität - wie bei der 2. Strategie.
- Offenheit - wie bei der 2. Strategie, wobei aber u.U. bestehende Absprachen anzupassen sind.

Zusammenfassend kann man sagen, daß die Vorteile der beiden Strategien vereint werden können. Dabei sind aber auch einige Nachteile in Kauf zu nehmen, vor allem die Tatsache, daß das WfMS einen *single point of failure*, u.U. auch ein *bottleneck* darstellt. Dies ist aber ohnehin der Fall, da nur das WfMS auf Informationen zum Ausführungsstand der Workflows direkt zugreifen kann.

**Globale Zugriffskontrolle in Workflows**

Wie bereits angesprochen steht hier die Zugriffskontrolle in interorganisationellen Workflows im Mittelpunkt der Untersuchungen. Aus diesem Grund wurde nur zu diesem Aspekt ein Grundansatz gemäß Strategie 3 weiter ausgearbeitet und realisiert. Dabei wurde die Rolle eines WfMS als zentraler Informationsserver für Autorisierungen nach einem globalen Autorisierungsschema untersucht, und die Anforderungen bestimmt, die an einen solchen Server zu stellen wären. Diese werden vor allem durch die Forderung nach angemessener Sicherheit und Flexibilität der Workflows, wie z.B. im *electronic commerce* erforderlich, impliziert. Es sind [Same95]:

- Zugriffe der Benutzer sollten entsprechend vorgegebener Informationskategorien, gemäß dem Prinzip der *least privileges* geregelt werden.
- Zugriffe sollten sich an Sicherheitseinstufungen der Subjekte (z.B. Benutzer, Applikationen) und Objekte (z.B. Dateien, Relationen) orientieren, um sowohl die Sensibilität von Daten als auch die Vertrauenswürdigkeit von Personen zu berücksichtigen (Angemessenheit).
- Spezielle Einschränkungen bezüglich Identität, Gruppe, Authentifikationsgrad, Netzwerk Lokation und der Zugriffszeit sollten spezifizierbar und überprüfbar sein (Kausalitäten).
- Die Unterteilung einzelner oder mehrerer Organisationen in unterschiedliche Domains sollte berücksichtigt werden (Mehrseitige Sicherheit).
- Die Delegation von Privilegien an andere Subjekte sollte prinzipiell möglich sein.

Vor jedem Zugriff muß eine Zugriffsfunktionsfunktion die Zugriffskontrolle (Rechteprüfung und Rechtevergabe) gemäß den Angaben zu Subjekten und Objekten, durchführen, und dabei den Kontext (Zeit des Zugriffs, Subjektidentität und Lokationen, etc.) berücksichtigen [Same95].

### 5.2.3 Die SWATS Basis-Architektur

Das SWATS enthält als Kern die kommerzielle Workflow-Engine *Montana* (Hewlett-Packard), an die weitere Komponenten zur Unterstützung der synchronen Telekooperation bei der Ausführung von Workflows gekoppelt sind. Das Gesamtsystem besitzt eine durch CORBA integrierte Client / Server Architektur, dessen Basisfunktionen bereits implementiert sind [Poli98b]. Zum besseren Verständnis der entwickelten Sicherheitsarchitektur soll dieses System kurz skizziert werden.

#### Funktionelle Gliederung

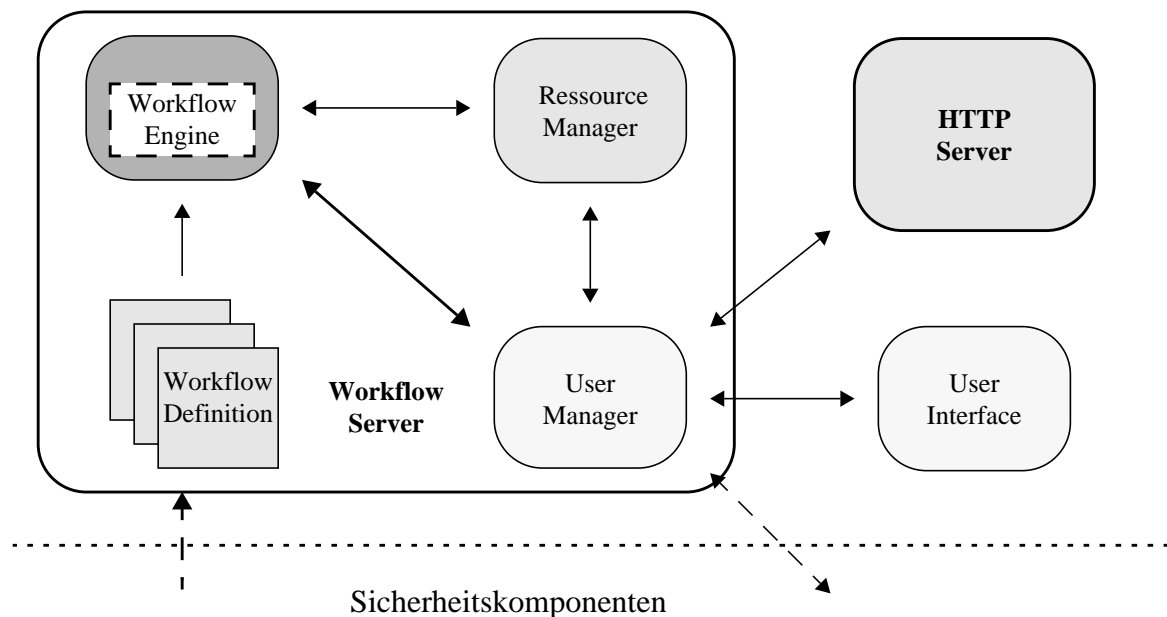
Die Komponentenarchitektur von SWATS (vgl. Abb. 5.2) läßt sich grob in zwei funktionelle Subsysteme unterteilen, die Client-Seite und die Server-Seite. Letztere besteht aus einem Workflow-Server, der eine mittels CORBA implementierte Systemschnittstelle anbietet, und einem HTTP-Server, der den Workflow-Clients Aktivitäten durch ein visuelles Frontend vermittelt. Der Workflow-Server ist in drei weitere Komponenten untergliedert, die - durch CORBA-ORB Dienste verbunden - gemeinsam die Basisfunktionen des Systems erbringen [Poli98b]:

- Workflow-Engine - sie bestimmt mit Hilfe der aktuellen Workflow-Zustände die nächsten auszuführenden Aktivitäten, und protokolliert dabei die internen Abläufe und Entscheidungen. Workflow-Definitionen werden gemäß einem speziellen Workflow-Modell in der Workflow-Sprache *Montana PDL* erstellt.
- Ressource Manager - er bestimmt auf Anfrage der Workflow-Engine hin die möglichen Akteure zu einzelnen Aktivitäten (Rollenauflösung). Dazu greift er auf spezielle Informationen über den Aufbau betreffender Organisationen zu.
- User Manager - er stellt an einer externen Schnittstelle Basisdienste für die Benutzer, die Workflow-Clients, bereit. Dazu verwaltet er Arbeitslisten mit den Aktivitäten der Benutzer, worauf diese mit Hilfe eines User Interface (klientenseitig) zugreifen können, um über angebotene oder zugeteilte Aktivitäten nach eigenem Ermessen zu entscheiden.

Der HTTP-Server schließlich hält aufbereiteten HTML- und Java Code bereit, um die Benutzer am User Interface bei der Durchführung ihrer Aufgaben im Workflow zu unterstützen [Poli98b].

#### Ankopplung der Sicherheitskomponenten

Voraussetzung für die angemessene Umsetzung der Sicherheitsanforderungen in Workflows ist deren Spezifikation vorab, soweit möglich, gemeinsam mit den Workflow-Definitionen [Pern97b]. Diese Forderung motiviert eine erweiterte Workflow-Definitionen, deren Umsetzung



**Abbildung 5.2:** Komponentenarchitektur von SWATS

die Ankopplung entsprechender Komponenten am Workflow-Server erforderlich macht. Die hierzu am besten geeignete Komponente ist der User Manager: Zum einen laufen hier alle Informationen der Aktivitäten, der Rollenauflösung und des HTTP-Servers zusammen, zum anderen kann hier die Zuteilung und Abwicklung der Aktivitäten weitgehend transparent beeinflusst werden.

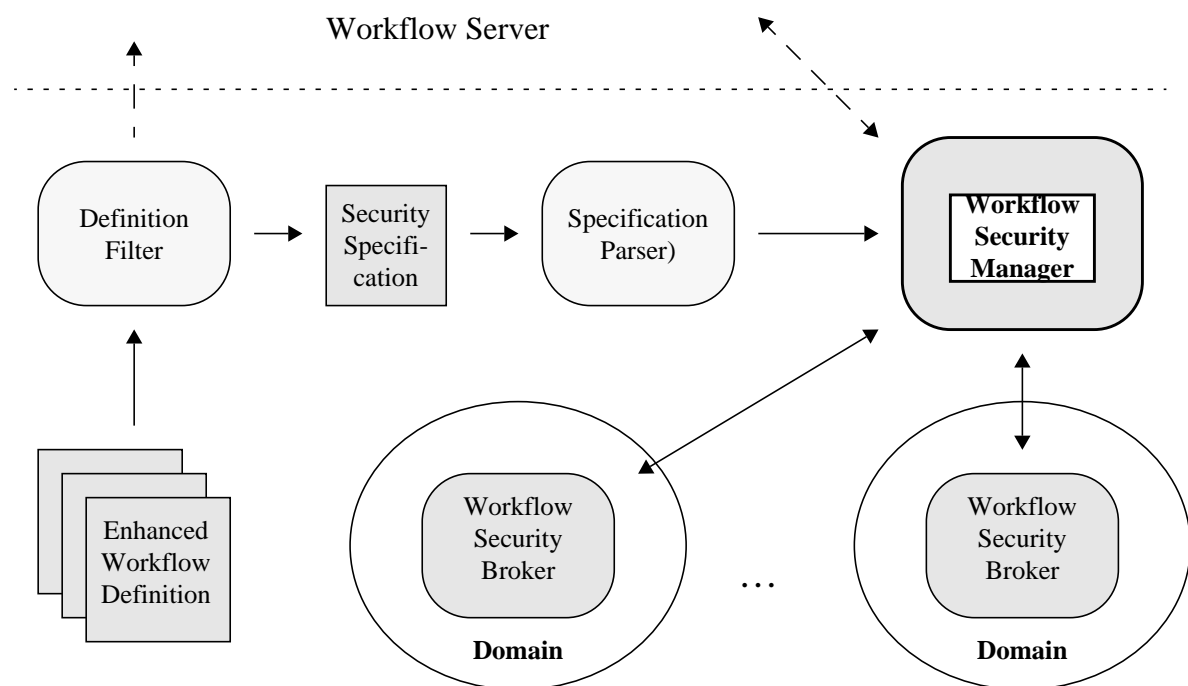
## 5.2.4 Die SWATS Sicherheitsarchitektur

Die SWATS Sicherheitsarchitektur wurde nach der 3. Strategie ausgelegt (vgl. Kapitel 5.2.2), und besitzt somit prinzipiell auch deren Eigenschaften hinsichtlich Sicherheit, Effizienz, Flexibilität und Offenheit. Dabei wurden auch Probleme berücksichtigt, die sich aus einer heterogenen und verteilten Realisierung der Zugriffskontrolle ergeben (Vgl. [Ker95]). Im derzeitigen Stand der Entwicklung werden die Integrität und Vertraulichkeit bei der Zugriffskontrolle berücksichtigt. Die Architektur ist aber derart ausgelegt, daß die Berücksichtigung weiterer Sicherheitsziele, bei Einbezug anderer Sicherheitsdienste, mit nur geringem Aufwand möglich ist.

Durch die Realisierung basierend auf einem hierzu entwickelten speziellen Sicherheitsmodell für interorganisationelle Workflows wird es, nicht zuletzt, auch den Anforderungen eines zentralen Informationsservers für globale Autorisierungen gerecht.

### Funktionelle Gliederung

Bei der Entwicklung der Sicherheitsarchitektur wurde von einer erweiterten Workflow-Definitionen ausgegangen, die neben den eigentlichen Workflow-Definitionen auch zugehörige Sicherheitsspezifikationen enthalten. Diese sollen zur weiteren Verarbeitung voneinander getrennt werden. Um dabei syntaktische Fehler auszuschließen müssen die Spezifikationen vorher grammatikalisch geprüft werden. Im Betrieb soll die Einbindung an den Workflow-Server über eine Schnittstelle zum User-Manager erfolgen, und die Umsetzung der Sicherheit in den Domänen bedarf zusätzlicher (domänenspezifischer) Komponenten. Somit ergibt sich die Architektur wie sie in Abb. 5.3 dargestellt ist. Die zusätzlichen Komponenten gegenüber der Basisarchitektur sind folgende:



**Abbildung 5.3:** Sicherheitsarchitektur von SWATS.

- Definition-Filter - er liest die Erweiterte Workflow-Definition ein, und generiert daraus die eigentliche Workflow-Definition und die Sicherheitsspezifikationen.
- Specification-Parser - er unterzieht die Spezifikationen einer grammatikalischen Analyse, und leitet sie bei syntaktischer Korrektheit weiter.
- Workflow-Security-Manager (WSM) - diese zentrale Komponente übernimmt korrekte Spezifikationen, um sie temporär zwischenspeichern. Bei der Ausführung von Workflows werden diese herangezogen, um Prüfungen zur Sicherheit durchzuführen und ggf. eine Umsetzung auf Domänenseite anzustoßen.

- Workflow-Security-Broker (WSB) - diese domänenspezifischen Komponenten befinden sich außerhalb des WfMS in den beteiligten Domains, und sind verantwortlich für die Umsetzung der Sicherheit vor Ort. Daneben können auch in ihnen spezielle Sicherheitsprüfungen lokal durchgeführt werden.

Neben obigen Anforderungen existieren noch weitere, sehr konkrete Anforderungen an die funktionellen Eigenschaften des WSM [Pat98].

### **Zusammenspiel der Komponenten**

Der Definition-Filter generiert vor Ausführung eines Workflows aus der Enhanced-Workflow-Definition die Workflow-Definition und die zugehörige Security-Specification. Erstere wird an die Workflow-Engine, letztere dagegen an den Specification-Parser weitergeleitet, und nach erfolgreicher Analyse an den WSM übertragen. Während der Ausführung des Workflows führt der User-Manager vor jeder Aktivität eine Anfrage am WSM durch, um festzustellen ob die Aktivität ausgeführt werden darf oder nicht. Der WSM führt mit Hilfe der zwischengespeicherten Spezifikation interne, und mit den Workflow-Security-Brokern auch externe Prüfungen durch. Dabei werden ggf. konkrete Sicherheitsdienste in den betroffenen Domänen involviert. Fallen alle Prüfungen positiv aus und können die benötigten Sicherheitsdienste einbezogen werden, wird die Aktivität vom WSM freigegeben, ansonsten jedoch abgelehnt. Zum Ende jeder Aktivität wird der WSM vom User-Manager darüber in Kenntnis gesetzt, um ggf. in den Domänen temporär vergebene Privilegien wieder zu entziehen. Nach Ablehnung einer Anfrage kann der User-Manager eine erneute Rollenauflösung anstoßen, um die Anfrage am WSM mit neuen Akteuren zu wiederholen.

## **5.2.5 Realisierung der Sicherheitsarchitektur**

In vielen Arbeiten, die sich mit der Integration von Sicherheit in Workflows befassen, wird diese sehr eingeengt gesehen, und allein durch Adaption herkömmlicher Zugriffskontrollverfahren zu einer globalen Zugriffskontrolle realisiert. Obwohl die Zugriffskontrolle in interorganisationellen Workflows von fundamentaler Bedeutung ist, wird diese Vorgehensweise der Sicherheitsproblematik dabei nicht gerecht. Sicherheit muß gerade dort modular und erweiterbar sein, um außer den derzeitigen auch zukünftigen Bedrohungen begegnen zu können [Pern97a], [Pern97b].

Auf architektonischer Ebene kann diesen Forderungen durch eine adäquate Sicherheitsarchitektur genügt werden. Die angemessene Umsetzung der Sicherheit erfordert aber zusätzlich eine besondere Sichtweise, die ihren Ausdruck z.B. in einem speziellen Modell finden kann. Darin können alle Teilaspekte der Sicherheit konkret berücksichtigt werden, die bei Interaktionen im Rahmen interorganisationeller Workflows eine Rolle spielen.

### **Sicherheitsmodelle der IT**

Der Entwurf sicherer IT-Systeme erfordert eine Vorgehensweise, die man als *Security Engineering* bezeichnet. Dabei wird, nach einer vorausgegangenen Bedrohungsanalyse und Spezifikation der Sicherheitsanforderungen, ein präzises Modell des Systems unter Sicherheitsaspekten erstellt. Ein solches Modell erlaubt es von den Randbedingungen und Mechanismen der Realisierung zu abstrahieren, um sich auf die eigentliche Problematik zu konzentrieren. Dabei kann u.U. sogar der Nachweis von Sicherheitseigenschaften bereits auf abstrakter Ebene erfolgen [Eck95].

Die bisherigen Sicherheitsmodelle der IT unterscheiden sich stark in ihrer Fähigkeit, relevante Sicherheitsaspekte vollständig zu erfassen. Eine Auswahl der Modelle muß sich deshalb daran orientieren, welche (problemspezifischen) Anforderungen existieren, und welches Modell geeignet ist, diese lückenlos zu beschreiben [Eck95]. Sollte jedoch kein geeignetes Modell existieren, können bestehende Modelle u.U. als Ausgangsbasis zur Entwicklung eines solchen dienen.

### **Anforderungen an ein spezielles Sicherheitsmodell**

Ausgangsbasis der Realisierung ist ein spezielles Sicherheitsmodell für interorganisationelle Workflows, dessen Angemessenheit durch eine Reihe von Anforderungen sichergestellt werden sollte. Dazu gehört neben einer ausreichenden Ausdruckskraft auch die Berücksichtigung der relevanten Sicherheitsaspekte, gemäß den aktuellen Sicherheitsstandards, wobei auch die Kritik bisheriger Standards Eingang fand. Da vorerst nur eine globale Zugriffskontrolle angestrebt war, wurden auch die Anforderungen an einen zentralen Informationsserver berücksichtigt.

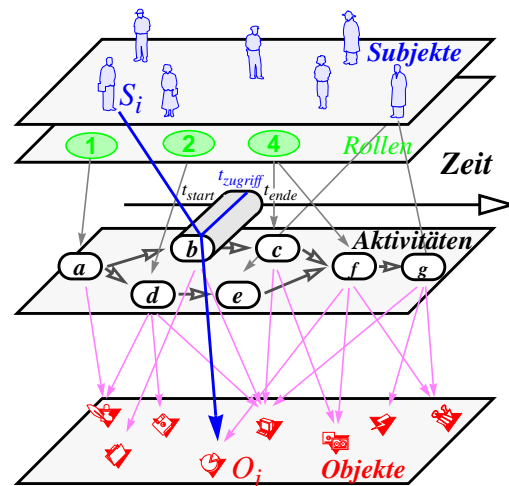
Um eine ausreichende Ausdruckskraft sicherzustellen, wurden einige allgemeine und spezielle Sicherheitsmodelle der IT untersucht, um daraus ein spezielles Modell zu konstruieren [Pat98]:

- **Subjekt-Objekt Modell** - dieses allgemeine Modell enthält grundlegende Komponenten, die bei einer abstrakten Analyse der IT-Sicherheit eine Rolle spielen. Dies sind neben den Initiatoren von Aktionen, den Subjekten (Benutzer, Prozesse, Rollen), auch darin involvierte Objekte (Daten, Prozesse, Rechner). Bei den Aktionen sind häufig auch problemspezifische Anweisungen (Vorgaben, Prozeduren) zu berücksichtigen [Ker95].
- **Domänen & Capabilities** - dieses Modell kann als Verallgemeinerung zahlreicher Zugriffskontrollpolitiken und der Implementierung von Zugriffsrechten angesehen werden [Ker95]. Es kann somit zum Nachweis der Abbildbarkeit allgemeiner (globaler) Zugriffskontrollverfahren auf konkrete Implementierungen in den Domains herangezogen werden [Pat98].
- **Framework für Sicherheit in Workflows** - hier wird versucht, die Sicherheitsinteressen von Objekten mit den Eigenschaften von Subjekten in Einklang zu bringen. Dazu werden allen Instanzen Sicherheitsgrade (je Sicherheitsziel) zugewiesen, und vor Aktionen miteinander verglichen. Bei einer ungenügenden Sicherheit der Subjekte wird der Einbezug zusätzlicher Sicherheitsdienste in entsprechender Stärke vorgeschlagen, um Aktionen unter veränderten Rahmenbedingungen dennoch durchführen zu können.

- Aktivitätsbasierte Zugriffskontrolle - dieses Modell (vgl. Abb. 5.4) berücksichtigt den zeitlichen Aspekt der Zugriffskontrolle in Workflows. Dazu werden Vergabe und Entzug der Zugriffsrechte im Kontext von Aktivitäten an deren Ausführung gekoppelt [Kind97].

Um die in Workflows relevanten Sicherheitsaspekte gemäß den aktuellen Sicherheitsstandards zu bestimmen, wurden letztere näher untersucht, und folgendes festgelegt:

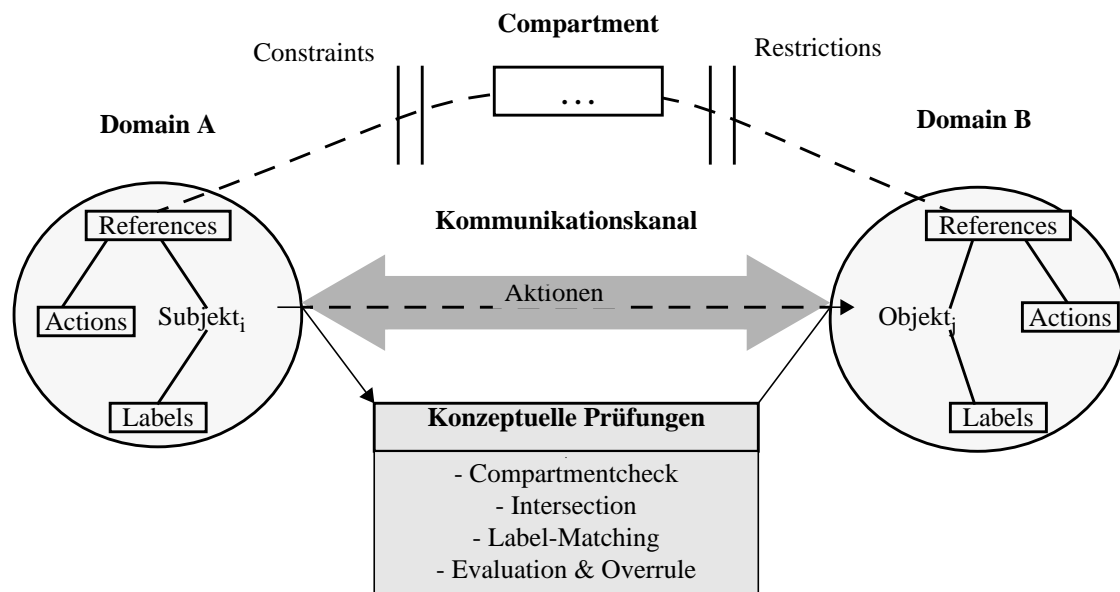
- Interaktionen in interorganisationellen Workflows finden in sehr komplexen Umgebungen statt, die sich aber i.d.R. in IT-Systeme und Kommunikationssysteme untergliedern lassen.
- Sicherheit ist stets im Kontext sicherheitskritischer Aktionen zu betrachten, nämlich den abstrakten Aktionen (Typen) Zugang, Zugriff, Speicherzugriff und Kommunikation.
- Zur Zugriffskontrolle existieren in IT-Systemen i.d.R. zumindest die konkreten Aktionen (Zugriffsrechte) *read*, *write*, *append*, *change*, *create*, *delete* und *execute*.
- Die relevanten Sicherheitsziele entsprechen denen in Kapitel 5.1.2, wobei die Grundziele vor allem in IT-Systemen und die sonstigen Ziele in Kommunikationssystemen von Bedeutung sind.
- Sicherheitsdienste ergeben sich ebenfalls wie in Kapitel 5.1.2, wobei die eine variable Stärke vorzusehen ist. Unterschiedliche Sicherheitsmechanismen sind nur bei der Zugriffskontrolle, hinsichtlich der Abbildbarkeit der globalen Zugriffskontrolle zu untersuchen.
- Elementare Sicherheitsprinzipien wie *least privileges*, *separation of duties*, *limited privileges* und *prerequisite privileges* (Vgl. [Sand96]) sollten festlegbar sein.



**Abbildung 5.4:** Aktivitätsbasierte Zugriffskontrolle

### SIW - Sicherheitsmodell für interorganisationelle Workflows

Das Sicherheitsmodell SIW vereint Komponenten und Konzepte obiger Modelle, wobei diese aber weiter detailliert und ergänzt werden. In einer daraus abgeleiteten Spracherweiterung *Workflow Security Language* sind zwar nur Spezifikationen zur globalen Sicherheit möglich, doch erlaubt eine weitere Spracherweiterung *Domain Security Language* auch Einschränkungen zu speziellen organisatorischen Kausalitäten[Pat98].



**Abbildung 5.5:** Sicherheitsmodell für interorganisationelle Workflows

Im Modell **SIW** (vgl. Abb. 5.5) werden Subjekte und Objekte einzelnen Domains zugewiesen, wobei Aktionen über Kommunikationskanäle stattfinden. Ihre Sicherheitsanforderungen bzw. -eigenschaften werden in Form von Sicherheitsstufen(Label) festgelegt, geforderte bzw. angebotene funktionelle Eigenschaften dagegen in Aktionen. Logisch verbunden werden Instanzen durch Referenzen, die zur effizienteren Verwaltung und besseren Übersichtlichkeit in Zuständigkeitsbereiche(Compartments) zusammengefaßt werden. Kausalitäten werden in Form logischer Ausdrücke als globale *Constraints* (z.B. Sicherheitsprinzipien) oder organisatorische *Restrictions* (z.B. Einschränkungen) festgelegt, was genau genommen zu bedingten Referenzen führt.

Vor der Freigabe jeder Aktion finden grundsätzlich folgende Sicherheitsprüfungen statt [Pat98]:

- *Compartment Check* - Aktionen können nur dann (prinzipiell) stattfinden, wenn Subjekt und Objekt mindestens ein *Compartment* gemein haben.
- *Intersection* - durch bilden der Schnittmenge der vom Subjekt geforderten und vom Objekt angebotenen Aktionen werden die tatsächlich ausführbaren Aktionen bestimmt.
- *Label Matching* - dabei wird geprüft, ob die Sicherheitsgrade eines Subjekts mindestens so hoch sind, wie die des involvierten Objekts. Ist dies nicht der Fall, wird wie im Framework für Sicherheit in Workflows vorgeschlagen verfahren (Vgl.[Pern97b])
- *Evaluation* - dabei werden *Constraints* und *Restrictions* ausgewertet und konjunktiv verknüpft, was bei mindestens einem negativem Ergebnis zur Ablehnung der Aktion führt. Wird dabei das *Overrule* Konzept angewandt um eine Kausalität auszusetzen, wird die entsprechende *Constraint* bzw. *Restriction* bei der Evaluation auf *true* gesetzt.



## 5.3 Zusammenfassung

Das Ziel dieses Arbeitsbereichs war Sicherheitsaspekte bei der Ausführung von Workflows zu berücksichtigen. Dabei sollte es möglich sein, Spezifikationen zur Zugriffskontrolle bei der Ausführung von Workflows umzusetzen.

Zu diesem Zweck wurden zuerst nationale und internationale Sicherheitsstandards untersucht, um daraus die für Workflows relevanten Sicherheitsaspekte abzuleiten. Im weiteren wurden Anforderungen an die Sicherheit in Workflows, speziell an die Zugriffskontrolle in interorganisationelle Workflows bestimmt, um die Anforderungen an eine angemessene Modellierung festzulegen. Mit Hilfe eines allgemeinen Sicherheitsmodells zur Beschreibung der Komponenten eines IT Systems wurde dann ein Sicherheitsmodell entwickelt, das diese Anforderungen erfüllt. Durch eine schrittweise Verfeinerung und Hinzunahme weiterer Komponenten wurde letztlich ein Modell entwickelt, das die Sicherheit in Workflows einerseits organisationsübergreifend beschreibt, andererseits aber auch die Belange einzelner Sicherheitsdomains berücksichtigt.

Auf der Basis des “Sicherheitsmodells für interorganisationelle Workflows” und des “Aktivitätenbasierten Sicherheitsmodells” wurde der SWATS Prototyp so erweitert, daß von der Modellierung über die Vorgangsausführung bis zur Aktivitätsbearbeitung die Sicherheitsanforderungen integral und durchgängig unterstützt werden.

## 5.4 Literatur

- [Beut90] Beutelspacher, A.; Gundlach, M. (1990). *Algorithmen, Mechanismen und Dienste*. Informationstechnik it, 1/90 : 24-32. R. Oldenburg Verlag, 1990.
- [Cer97] Cerny, D. (1997). *Information Warfare - Eine neue Bedrohung für Staat und Wirtschaft ?* Tagungsband zum 5. Deutschen IT-Sicherheitskongreß des Bundesamts für Sicherheit in der IT, 1997
- [Eck95] Eckert, C. (1995). *Vorlesungsskript Sichere Rechensysteme WS 95/96*. Institut für Informatik, TU München, 1995.
- [Fumy95] Fumy, W.; Horster, P.; Kraaibeek, P. (1995). *Standards und Patente zur Sicherheit in der IT*. Oldenburg Verlag GmbH, München, 1995.
- [Glei89] Gleißner, W (1989) *Manipulationen in Rechnern und Netzen: Risiken, Bedrohungen und Gegenmaßnahmen*. Addison Wesley, Bonn, 1989.
- [Ker95] Kersten, H. (1995). *Sicherheit in der Informationstechnik - Einführung in Probleme, Konzepte und Lösungen*. Oldenburg Verlag GmbH, München, 1995.

- [Kind97] Kindler, T. (1997). *Activity based security in intranet and internet workflows*. Proceedings of the seventh Annual Workshop on Information Technologies and Systems, 1997.
- [Lin96] Lin, P.; Lin, L. (1994). Security in Enterprise Networking : A Quick Tour. IEEE Communications Magazine, 1/96 : 56-61.
- [Pat98] Pataric, Z. (1998). *Erweiterung eines Workflow Managementsystems um Sicherheitsaspekte bei der Modellierung und Ausführung von Workflows*. Diplomarbeit am Institut für Parallele und Verteilte Höchstleistungsrechner, Uni Stuttgart, 1998.
- [Pern97a] Pernul, G.; Hermann, G. (1997). *Zur Bedeutung von Sicherheit in introrganisationalen Workflows*. Wirtschaftsinformatik, 39 : 217-224, 1997.
- [Pern97b] Pernul, G.; Hermann, G. (1997). *A General Framework for Security and Integrity in Interorganizational Workflows*. Proceedings of the Tenth International Bled Electronic Commerce Conference, 1997.
- [Poli98a] *Poliflow- Überblick und Motivation (Zuverlässige und sichere Vorgangsbearbeitung für weitverteilte Anwendungsumgebungen unter Berücksichtigung gemischter Arbeitsformen)*. <http://www.informatik.uni-stuttgart.de/ipvr/as/projekte/poliflow/poliflow.html>, 1998.
- [Poli98b] *SWATS - Stuttgarter Workflow- und Telekooperationsystem*. <http://www.informatik.uni-stuttgart.de/ipvr/as/projekte/poliflow/poliflow.html>, 1998.
- [Pom91] Pommerening, K. (1991). *Datenschutz und Datensicherheit*. BI-Wissenschafts-Verlag, Mannheim, 1991.
- [Ran95] Rannenber, K. (1995). *Evaluationskriterien zur IT Sicherheit - Entwicklungen und Perspektiven in der Normung und außerhalb*. Verlässliche IT Systeme - Proceedings der GI-Fachtagung, 1995.
- [Same95] Sameshira, Y.; Kirstein, P. (1997). *Authorization with security attributes and privilege delegation, Access control beyond the ACL*. Computer Communications, 20 : 376-384. Elsevier Science B.V., 1995.
- [Sand96] Sandhu, R; Coyne, E. (1996). *Role Based Access Control Models*. Computer, 2/96: 38-47, 1996.
- [SSON96] Pfitzmann, A; Schill, A, (1996). *SSONET - Sicherheit und Schutz in offenen Datenetzen*. <http://mephisto.inf.tu-dresden.de/RESEARCH/ssonet/reports.html>, 1998.

# 6 Gemischte Arbeitsformen (MWM)

## 6.1 Motivation

Der hauptsächliche Verwendungszweck von Computeranwendungen im kommerziellen Bereich besteht in der Unterstützung einzelner Tätigkeiten am Arbeitsplatz. Für die Vielzahl verschiedener Bürotätigkeiten werden eine ganze Reihe von Applikationen auf dem Software-Markt angeboten: Textverarbeitungssysteme für Schreibarbeiten, CAD-Systeme zur Durchführung von Konstruktionsarbeiten, Tabellenkalkulationen usw. Im allgemeinen sind diese Software-Systeme nur auf die Interaktion eines Benutzers mit dem System ausgerichtet, Mechanismen, die Interaktionen und Kommunikation zwischen Benutzern unterstützen, fehlen.

Ein großer Teil der im Geschäftsleben anfallenden Arbeiten erfolgt jedoch eher im Kontext einer Gruppe und nicht in einem individuellen Kontext. Zu solchen Gruppenaktivitäten gehören beispielsweise der Austausch oder das Weiterleiten von Dokumenten, Diskussionen oder auch das gemeinsame Erstellen und Bearbeiten von Dokumenten.

Auf dem Software-Markt werden heutzutage diverse Werkzeuge angeboten, die gewisse Formen elementarer kooperativer Gruppenarbeit unterstützen. Dazu gehören Videokonferenzsysteme, Nachrichtensysteme wie e-mail zum Versenden von Nachrichten oder Workflow-Systeme zur automatischen Weiterleitung von Dokumenten. Mit diesen Werkzeugen, lassen sich in der Regel aber nur einzelne Teile langdauernder komplexer Arbeitsprozesse unterstützen.

In räumlich weit verteilten Organisationseinheiten, wie sie sowohl im Behörden- und Dienstleistungssektor als auch im industriellen Bereich zu finden sind, besteht ein wachsender Bedarf an geeigneten Systemen, die Geschäftsvorgänge in ihrer Gesamtheit und nicht nur einzelne separate Teile davon unterstützen. Durch den zunehmenden Einsatz von Rechnern am Arbeitsplatz und die rasante Entwicklung und Verbreitung der Internet- und Intranet-Technologie rückt die Notwendigkeit der Unterstützung kooperativer Tätigkeiten immer stärker in den Vordergrund.

Durch Einsatz modernster Technologie soll eine reibungslosere und effizientere Abwicklung von Geschäftsvorgängen erreicht werden. Als besonders problematisch erweist sich hierbei, daß während der Abwicklung eines Geschäftsprozesses die unterschiedlichsten Formen kooperativer Arbeit auftreten können, und daß Vorgänge oftmals dynamischen Anpassungen in ihrem Ablauf unterliegen. Ein universelles Software-System zur ganzheitlichen Unterstützung betrieblicher Aufgaben muß demnach nicht nur anpassungsfähige Vorgänge, sondern darüberhinaus auch

gemischte Arbeitsformen unterstützen. Dabei muß das ganze Spektrum von asynchronen bis synchronen sowie von strukturierten bis hin zu unstrukturierten Aktivitäten und Vorgängen unterstützt werden.

Die Untersuchungen im Anwendungsumfeld der Oberfinanzdirektion (OFD) Berlin haben gezeigt, daß die dort anfallenden Geschäftsprozesse sehr häufig Aktivitäten kooperativer Art enthalten. Abstimmungen zwischen einzelnen Bereichen (z.B. Architekten, Elektro und Sanitär im Rahmen des HU-Bau-Vorgangs) sind ebenso wichtiger Bestandteil, wie auch planerische Aktivitäten in Form von Konferenzen, bei denen der weitere Verlauf eines Vorgangs dynamisch festgelegt wird. Der nächste Abschnitt gibt zunächst einen kurzen Überblick über den Forschungsbereich, der sich mit rechnergestützter Kooperation beschäftigt.

## 6.2 CSCW, Groupware & Workflow

Der Forschungsbereich Computer Supported Cooperative Work (CSCW) beschäftigt sich mit der Zusammenarbeit von Personen innerhalb von Gruppen und mit den Technologien, mit denen Gruppenarbeit unterstützt werden kann (siehe auch [Elli91]). Im Zusammenhang mit CSCW spielen insbesondere die beiden Begriffe Groupware und Workflow eine zentrale Rolle.

### 6.2.1 Groupware

Gruppenarbeit ist dadurch charakterisiert, daß mehrere Personen gemeinsam als Team an der Lösung einer Aufgabe arbeiten. Effektive Teamarbeit setzt einen gleichen Wissensstand innerhalb der Gruppe voraus, das heißt, die benötigten Informationen müssen allen Beteiligten in ihrer aktuellsten Fassung zur Verfügung stehen. Befinden sich alle Mitglieder einer Arbeitsgruppe zur selben Zeit am selben Ort, so können (Sach-) Informationen ausgetauscht und die Aufgaben gemeinsam bearbeitet werden. Aber gerade die zeitgleiche Anwesenheit aller Mitglieder größerer Gruppen am selben Ort gestaltet sich insbesondere bei räumlich weit verteilten Organisationseinheiten oftmals schwierig. Durch den Einsatz von Groupware-Werkzeugen sollen die Mitglieder von Arbeitsgruppen oder Projektteams zum Beispiel so unterstützt werden, daß deren gleichzeitige physische Anwesenheit am selben Ort zur Erreichung ihres Ziels nicht mehr unbedingt erforderlich ist.

Für den Begriff Groupware existieren in der Literatur eine ganze Reihe von Definitionen (siehe z.B. [Bochen94], [Elli91]). In [Elli91] wird Groupware definiert als

*“computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment”.*

Eine Taxonomie für Groupware-Systeme, die man häufig in der Literatur findet (siehe dazu [Elli91], [Lewe91], [Robi94], [Nava92], [Berg93]), unterscheidet zwischen zeitlichen und räumlichen Aspekten. Eine solche Raum-Zeit-Klassifikation von Groupware-Werkzeugen ist in der nachfolgenden Tabelle dargestellt.:

Anwesenheit der Teilnehmer	zur gleichen Zeit (synchron)	zu unterschiedlichen Zeiten (asynchron)
am gleichen Ort (zentral)	face-to-face interaction, Group Decision Support Systems (GDSS)	Bulletin Board, Time Scheduling
an unterschiedlichen Orten (verteilt)	Shared Screen, Computer Conferencing, Real-Time Group Editors, Audio-, Video-Conferencing	Procedure Coordination, Message System

**Tabelle 6.1:** Klassifikation von Groupware-Systemen

## 6.2.2 Workflow-Systeme

Ein anderes Teilgebiet des umfassenden Forschungsbereichs CSCW stellt das Workflow-Management dar [Hase93]. Workflow-Systeme dienen zur Unterstützung der Abwicklung von Geschäftsprozessen. Geschäftsprozesse bestehen häufig aus mehreren Teilvorgängen, die teilweise parallel ausgeführt werden können, aufgrund von möglichen Abhängigkeiten zum Teil aber auch sequentiell ausgeführt werden müssen. Werden die einzelnen Teilvorgänge von verschiedenen Personen (-gruppen) bearbeitet, so muß das Workflow-System dafür sorgen, daß die jeweils betroffenen Personen auch die benötigten Dokumente zur Erfüllung ihrer Aufgabe rechtzeitig und vollständig zur Verfügung gestellt bekommen. Im Gegensatz zu Groupware-Werkzeugen sind Workflow-Systeme mehr dahingehend ausgerichtet, das Weiterleiten von Dokumenten einfacher und reibungsloser zu gestalten, insbesondere beim wiederholten Abarbeiten von Routinevorgängen. Zu den Aufgaben eines Workflow-Systems gehören (siehe dazu auch [Hase93], [Karl93]):

- Steuern, Kontrollieren und Überwachen von Vorgängen
- Zuordnung von Aktionen zu konkreten Akteuren
- sicheres und termingerechtes Weiterleiten/Versenden von Dokumenten an die zuständigen Mitarbeiter oder deren Vertreter
- Entlastung des Anwenders von Routineaufgaben (z.B. Suchen von Dokumenten) und damit Erhöhung des wertschöpfenden Zeitanteils

- Initiierung von Vorgängen (z.B. indem ein Bestellauftrag einen davon unabhängigen Vorgang zur Nachbestellung auslöst)
- Treffen von Entscheidungen aufgrund vorgegebener Bedingungen (z.B. Weiterleiten von Dokumenten an eine Vertretung, Alarmierung falls sich irgendwo Verzögerungen ergeben und ein Dokument nicht termingerecht fertig wird)
- Sammlung statistischer Informationen zur Erkennung von Einsparungspotentialen (hohe Liegezeiten,...)

Die Hauptziele von Workflow-Management-Systemen sind:

- Reduktion und Kontrolle von Durchlaufzeiten
- Verbesserung der Auskunftsbereitschaft
- Ruhe- und Liegezeiten von Vorgängen verkürzen

### **6.2.3 Abgrenzung zwischen Groupware und Workflow-Systemen**

Groupware und Workflow-Systeme unterstützen beide die kooperative Zusammenarbeit mehrerer Personen im Kontext einer Gruppe. Dennoch unterscheiden sie sich in der Art und Weise, mit der diese Unterstützung erfolgt (siehe dazu auch [Lewe91], [Boch94], [Hase93], [Karc93]). Der wohl wesentlichste Unterschied zwischen Groupware und Workflow-Systemen liegt in der Verteilung der aktiven und passiven Rollen zwischen System und Benutzer.

Groupware-Werkzeuge werden eingesetzt, um eine Gruppe als Gruppe zu unterstützen, wobei die Unterstützung des Einzelnen bei seiner Interaktion mit der Gruppe im Vordergrund steht. Groupware-Applikationen sind eher als passiv einzustufen. Sie kümmern sich weniger darum, wer spezielle Informationen zu sehen bekommt und was damit gemacht wird. Beim Workgroup-Computing stellt das System nur eine passive Infrastruktur bereit, auf der sein Benutzer dann aktiv alle Aktionen selbst bestimmt und verursacht. Damit lassen sich Groupware-Werkzeuge mehr als Instrumente zur Unterstützung von wenig strukturierten Vorgängen einsetzen, wie z.B. Entscheidungsfindungen oder das gemeinsame Erstellen von Dokumenten (Protokolle, Berichte, Publikationen, usw.).

Die Einsatzkonzepte von Groupware-Werkzeugen beziehen sich auf die Unterstützung eher informeller Aufgaben im Rahmen kleiner Gruppen, die durch Selbstorganisation gekennzeichnet sind. Die Steuerung und die Kontrolle des Arbeitsfortschritts liegen dabei auf der Seite des Benutzers.

Beim Workflow-Computing kommt dagegen eher dem Workflow-System die aktive Rolle zu und dem Benutzer die passive. Das System steuert Abläufe, löst Aktionen aus und reicht Dokumente selbsttätig an die betreffenden Personen weiter. Workflow-Management-Systeme dienen vor allem zur Unterstützung der aufgabenbezogenen Koordination und Kommunikation zwischen Aufgabenträgern im Rahmen organisatorisch geregelter Abläufe.

Workflow-Systeme finden insbesondere in der Unterstützung gut strukturierter Büroprozesse Anwendung. Dazu gehören beispielsweise (siehe [Karc93]) Beschaffungsanforderungen, Baugenehmigungsverfahren oder Auftragsabwicklungen.

Workflow-Systeme spielen auch bezüglich der Bestimmung, wer gewisse Daten erhält, eine aktive Rolle und entscheiden auch, was die jeweiligen Personen damit machen. Der Benutzer ist hier eher passiv und reagiert auf die Wünsche und Vorschläge des Workflow-Systems. Die Steuerung und Kontrolle des Arbeitsfortschritts obliegt beim Workflow-Computing eher dem System.

Die wesentlichen Unterschiede zwischen Groupware und Workflow-Systemen sind nochmals in der folgenden Tabelle zusammengefaßt.

	Groupware	Workflow-System
Rollenverteilung	System: passiv Benutzer: aktiv	System: aktiv Benutzer: passiv
Steuerung und Kontrolle des Arbeitsfortschritts	liegt beim Benutzer	liegt beim System
Einsatzgebiet	Problemlöseaktivitäten, Entscheidungsfindung	Sachbearbeitung, Routinetätigkeiten

**Tabelle 6.2:** Unterschiede: Groupware <-> Workflow-System

In Abhängigkeit von der Definition der Begriffe Groupware und Workflow werden Workflow-Systeme manchmal auch als eine spezielle Klasse von Groupware-Werkzeugen betrachtet. In der Taxonomie von Groupware-Systemen lassen sich Workflow-Systeme dann in der Kategorie asynchron / verteilt einordnen.

## 6.2.4 Werkzeuge zur Kooperationsunterstützung

Zur Unterstützung kooperativer Arbeit gibt es verschiedene Klassen von Werkzeugen. Unter *cooperation-aware applications* versteht man Anwendungen, die in Ihrer Konzeption bereits so ausgelegt sind, daß sie gleichzeitig mehreren Benutzern Zugang bieten. Dagegen sind *coopera-*

*tion-unaware applications* Anwendungen, die nicht für Mehrbenutzerbetrieb ausgelegt sind. Solche Werkzeuge können unter gewissen Rahmenbedingungen jedoch mit Hilfe von *Sharing-Tools* mehrbenutzertauglich gemacht werden.

Werkzeuge zur Kooperationsunterstützung sind beispielsweise e-mail, Shared Editors, Group Decision Support Systems, Audio-/Video-Conferencing Tools, Sharing Tools, Workflow-Systeme, usw.

## 6.3 Problembereiche und Anforderungen

Im Rahmen der Arbeitspakete MWM.2 und MWM.3 wurden die Anforderungen an eine Integration von Workflow-Systemen und Telekooperationswerkzeugen zur Unterstützung gemischter Arbeitsformen untersucht. Anforderungen, die speziell die allgemeine Funktionalität von Workflow-Systemen bzw. von Telekooperationswerkzeugen betreffen, werden hier nicht weiter betrachtet. Im folgenden werden einige der wichtigsten Anforderungen näher betrachtet, sowohl Anforderungen aus Benutzersicht als auch aus technischer Sicht:

### Vermeidung von Medienbrüchen

Für Konferenzen, bei denen mehrere Personen an einem Ort zusammenkommen, existieren eine ganze Reihe von Möglichkeiten, um Informationen zwischen den Teilnehmern auszutauschen. Beispiele hierfür sind Dokumente in Papierform, Videogeräte, Tafeln, Flip-Charts, usw. Tritt an die Stelle einer solchen Konferenz, die durch die physische Präsenz der Teilnehmer gekennzeichnet ist, eine Video-Konferenz, bei der die Teilnehmer auch im Sinne einer Telepräsenz partizipieren können, so sind oben genannten Hilfsmittel zum Informationsaustausch nur schlecht geeignet. Damit das auszutauschende Informationsmaterial allen Konferenzteilnehmern gleichermaßen zugänglich sein sollte, sollte das gesamte Informationsmaterial in elektronischer Form zur Verfügung stehen.

### Terminplanung

Synchrone Gruppenaktivitäten als Bestandteile von Geschäftsprozessen erfordern einen zusätzlichen Koordinierungsschritt zur Abstimmung eines gemeinsamen Termins zwischen den beteiligten Personen. Dabei sind folgende Punkte zu berücksichtigen:

- individuelle Terminpläne der Teilnehmer
- verfügbare Ressourcen
- zeitliche Vorgaben, bis zu der die Aktivität abgeschlossen sein muß



Für synchrone Gruppenaktivitäten festgesetzte Termine müssen vom System verwaltet und überwacht werden. Erfolgt die Terminplanung bereits mit Hilfe des Systems, so müssen die beteiligten Personen von dem vereinbarten Termin in Kenntnis gesetzt werden. Der Beginn einer Aktivität mit fest geplantem Start und Dauer sollte dann zum vorgegebenen Zeitpunkt vom System initiiert werden.

### **Effizienter Datenzugriff**

Zur Erhöhung des wertschöpfenden Zeitanteils ist es notwendig, dem Bearbeiter einen effizienten Zugriff auf die von ihm zu bearbeitenden Dokumente zu ermöglichen. Liegen alle Daten in elektronischer Form vor, so kann das Volumen, der für eine Aktivität benötigten Daten, sehr groß werden, z.B. wenn die Vorgangsunterlagen digitalisierte Videos enthalten. Speziell bei synchronen Gruppenaktivitäten wie Videokonferenzen in weit verteilten Anwendungsumgebungen, bei denen mehrere Personen auf dieselben Daten derselben Datenquelle zugreifen, kann eine unzureichende Leistungsfähigkeit der Netzwerkverbindungen zu störenden Übertragungsverzögerungen führen (z.B. beim gemeinsamen Betrachten einer Videoanimation). Um einen effizienten Datenzugriff durch den oder die Bearbeiter einer Aktivität zu gewährleisten, muß das System in der Lage sein, entsprechende Maßnahmen wie Migration, Replikation oder Duplizierung von Daten, in einer für den Benutzer transparenten Weise, vor dem Beginn einer Aktivität zu ergreifen. Dazu muß eine Möglichkeit vorhanden sein, mit dem das System rechtzeitig darüber in Kenntnis gesetzt werden kann, wann welche Daten von welchen Benutzern benötigt werden.

### **Reservierung von Ressourcen**

Für synchrone Gruppenaktivitäten mit festgelegtem Beginn und Dauer muß vom System sichergestellt werden, daß die benötigten Ressourcen, wie z.B. Speicherplatz, Bandbreite, Videokonferenzräume usw., zum vorgesehenen Zeitpunkt auch tatsächlich zur Verfügung stehen. Kann eine solche Aktivität wegen nicht verfügbaren Ressourcen nicht durchgeführt werden, so kann dies negative Auswirkungen auf die weitere Vorgangsbearbeitung zur Folge haben. Es muß nicht nur ein Ersatztermin anberaumt werden, sondern der potentielle Beginn der unmittelbar nachfolgenden Aktivitäten wird auch verzögert. Der Arbeitsfluß der beteiligten Personen wird unterbrochen und eventuell getätigte Vorbereitungsmaßnahmen einzelner Konferenzteilnehmer fallen unter Umständen erneut an.

### **Anpassungsfähigkeit von Workflows**

Nicht alle Vorgänge im Geschäftsleben lassen sich bis ins Detail vorausplanen oder exakt gemäß ihrem geplanten Ablauf durchführen. In solchen Fällen kann das weitere Vorgehen zu bestimmten Zeitpunkten während der Abwicklung des Vorgangs von einer einzigen Person bestimmt oder aber auch zwischen mehreren Personen im Rahmen einer Konferenz abgestimmt werden. Solchen Aktivitäten, in denen dynamisch über den weiteren Ablauf des Vorgangs entschieden wird, machen es erforderlich, daß Vorgangsinstanzen von den dazu berechtigten Benutzern dynamisch

zur Laufzeit modifiziert werden können (vgl. Arbeitsbereich AWS). Dabei kann es vorkommen, daß Aktivitäten hinzugefügt, übersprungen, weggelassen oder zusammengefaßt werden, daß Abhängigkeiten zwischen Aktivitäten verändert werden, oder auch daß der Dokumentenfluß modifiziert wird.

### **Verwendung von Standards**

Liegen Vorgangsdaten in elektronischer Form vor, so werden nicht nur unterschiedliche Werkzeuge für verschiedene Dokumenttypen benötigt, sondern mitunter kommen auch verschiedene Werkzeuge zur Bearbeitung derselben Klasse von Dokumenten zum Einsatz. Die Wahl der an einem bestimmten Arbeitsplatz verwendeten Werkzeuge hängt von den vom Bearbeiter zu erledigenden Aufgaben, den Dokumententypen und von den Präferenzen des Bearbeiters für bestimmte Werkzeuge ab. Die Komponenten eines Systems zur Unterstützung gemischter Arbeitsformen müssen demnach so offen sein, daß eine unkomplizierte Integration neuer Komponenten bzw. Werkzeuge möglich ist. Idealerweise sollten in einem solchen System Standards für Schnittstellen und Datenformate zugrunde gelegt werden, um eine problemlose Zusammenarbeit der einzelnen Komponenten, wie z.B. für den Austausch von Daten, zu gewährleisten. Dies ist insbesondere dann ein wichtiges Kriterium, wenn Produkte verschiedener Hersteller auf heterogenen Plattformen zum Einsatz kommen, wie es häufig in weit verteilten Anwendungsumgebungen der Fall ist.

### **Nahtlose Integration von Applikations- und Telekooperationskomponenten**

Eine Workflow-Umgebung zur Unterstützung gemischter Arbeitsformen soll eine unterstützende Funktion haben, so daß sie sich die Aktoren vorwiegend auf die Bearbeitung der eigentlichen Aufgabe konzentrieren können und nicht durch technische oder organisatorische Schwierigkeiten aufgehalten werden. Dazu gehört auch, daß die Benutzeroberfläche des Systems den Bearbeiter mit möglichst wenig technischen Details konfrontiert, sondern in einfacher und übersichtlicher Weise dem Benutzer 'ihre Dienste' anbietet. Für die Bearbeitung von Dokumenten müssen entsprechende Werkzeuge angeboten werden. Dabei ist zu beachten, daß an verschiedenen Arbeitsplätzen durchaus für die Bearbeitung desselben Dokuments unterschiedliche Werkzeuge zum Einsatz kommen und nicht an allen Arbeitsplätzen dieselben Dokumenttypen bearbeitet werden. Ist beispielsweise zum Erstellen eines Konstruktionsplans ein komplexes CAD-System notwendig, so kann zum Betrachten der fertigen Zeichnung ein einfacher Viewer bereits vollkommen ausreichen. Weiterhin müssen dem Benutzer auch Telekooperationswerkzeuge zum Aufbau von Audio- und Videoverbindungen leicht zugänglich gemacht werden. Als Anwendungsbeispiel kann hier die Rückfrage an einen Bearbeiter einer vorangegangenen Aktivität angeführt werden, auf die in Abschnitt 6.4.2 noch näher eingegangen wird.

Im Rahmen von PoliFlow wurde im Bereich gemischte Arbeitsform der Schwerpunkt auf die letztgenannte Anforderung gelegt. Dabei wurden Konzepte für eine integrierte Lösung zur Unterstützung kooperativer Arbeit entwickelt und prototypisch realisiert. Die Ergebnisse werden im folgenden weiter ausgeführt.

## **6.4 Konzepte und Lösungsansätze zur Unterstützung von Kooperation**

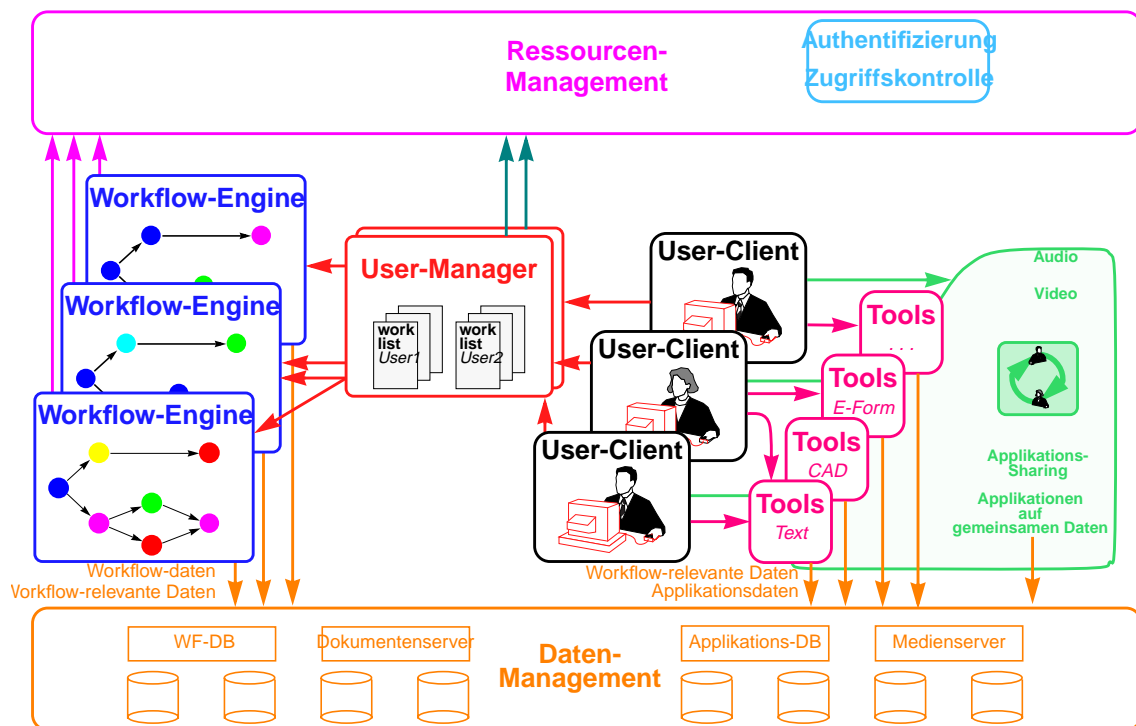
Das Spektrum kooperativer Arbeitsformen ist sehr vielfältig und umfaßt unter anderem spontane Rückfragen mit und ohne Dokumenten-Sharing, aufgabenbezogene Kommunikation, aber auch geplante Audio/Video-Konferenzen mit zusätzlichen Kommunikationsmitteln (Whiteboard, Shared Editing, usw.). Man unterscheidet, ob Kooperation geplant oder ad hoc erfolgt, im Kontext eines konkreten Vorgangs oder losgelöst von einer speziellen Vorgangsinstantz.

Die Integration von Workflow- und Telekooperationswerkzeugen kann insbesondere dann rasch komplex werden, wenn die kooperativen Tätigkeiten sehr eng mit dem gesamten Vorgang verflochten sind. Im Rahmen von PoliFlow wurde zunächst eine Referenzarchitektur eines Systems zur Unterstützung gemischter Arbeitsformen entwickelt. Darauf aufbauend wurde ein Prototyp entwickelt, in dem bestimmte Kernaspekte realisiert wurden.

### **6.4.1 Referenzarchitektur**

Die Referenzarchitektur (vgl. Abb. 6.1) gliedert sich im wesentlichen in folgende logischen Blöcke:

- **Daten-Management**  
Diese Komponente ist für die Verwaltung der Daten des Workflow-Management-Systems zuständig. Das Daten-Management wird zur Verwaltung der Workflow-Kontrolldaten, der Anwendungsdaten sowie der workflow-relevanten Anwendungsdaten genutzt.
- **Workflow Engine**  
Sie ist für die eigentliche Ablaufsteuerung des Geschäftsprozesses zuständig. Sie prüft, wann welche Voraussetzungen erfüllt sind und Aktivitäten mitsamt zugehöriger Dokumente zur Bearbeitung freigegeben werden können.
- **User Manager**  
Er verwaltet die Aufgabenlisten der Benutzer und ermöglicht transparenten Zugriff auf ein verteiltes Workflow-System.



**Abbildung 6.1:** Referenzarchitektur

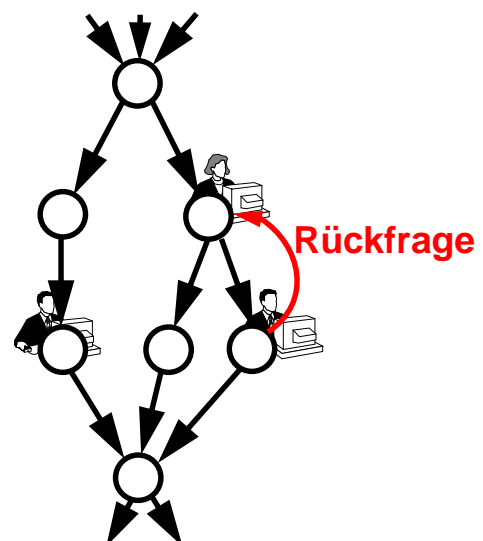
- **Ressourcen Management / Organization**  
Diese Komponenten dienen der Bereitstellung und Verwaltung der Informationen über die Organisationsstruktur eines Unternehmens. Hier werden Personen und Ressourcen verwaltet, die zur Ausführung von Aktivitäten zur Verfügung stehen, auch werden ihre Rollen und Kompetenzen innerhalb der Unternehmensstruktur abgebildet.
- **User Client**  
Er bildet die Schnittstelle für den Benutzer, die als Zugang zum Workflow-System dient. Hier kann sich ein Benutzer die Aktivitäten in seinem Eingangskorb und auf seinem Schreibtisch anzeigen lassen.
- **Toolbox**  
In der Toolbox werden verschiedener Werkzeuge zur Bearbeitung der Aktivitäten gesammelt.
- **Telecooperation Service**  
Der Telecooperation Service stellt spezielle Dienste zur Verfügung, die zur Durchführung synchroner Telekooperation benötigt werden. Videokonferenzen erfordern beispielsweise die Realisierung von Audio- und Videoübertragungen. Beim gemeinsamen Bearbeiten von Dokumenten müssen identische Darstellungen der Daten auf unterschiedlichen Bildschirmen realisiert und gemeinsame Interaktion ermöglicht sowie synchronisiert werden.

In einem ersten Schritt wurde eine Rückfrage-Funktionalität in eine Workflow-Umgebung integriert, mit der im Kontext einer Vorgangsbearbeitung auf einfache Weise eine Audio/Video-Konferenzschaltung zu einem vorangegangenen Bearbeiter aufgebaut werden konnte.

## 6.4.2 Integration einer Rückfrage-Funktionalität

Eine synchrone, unstrukturierte Aktivität, der im Büroalltag eine zentrale Rolle zukommt, ist die telefonische Rückfrage. Die Notwendigkeit von Rückfragen treten häufig im Kontext von Vorgangsbearbeitungen auf, wenn nämlich der aktuelle Bearbeiter in seinen Unterlagen Unklarheiten oder Unstimmigkeiten in den von einem seiner Vorgänger bearbeiteten Teile entdeckt. In solchen Fällen können Rückfragen entweder persönlich oder, da meist einfacher und bequemer, per Telefon erledigt werden.

Im Rahmen der Arbeitspakete MWM.4 und MWM.5 wurde ein Konzept entwickelt, um eine spontan initiierte Rückfrage im Kontext einer Vorgangsbearbeitung in ein Workflow-System zu integrieren. Die prinzipielle Idee ist in der Abbildung schematisch dargestellt. Die von einer Workflow-Engine verwalteten Aufgaben werden dem jeweiligen Benutzer über eine Benutzeroberfläche angezeigt. Die Benutzeroberfläche bietet weiterhin einen Mechanismus an (z.B. in Form eines Buttons), um im Kontext der aktuell von ihm bearbeiteten Aufgabe eine Videoverbindung aufzubauen. Aktiviert der Bearbeiter diesen Mechanismus (z.B.



durch Anklicken des Buttons), so erhält er eine Übersicht über die Aktivitäten des Vorgangs und ihrer Bearbeiter. Diese Darstellung kann in Form einer Liste oder auch wie in Abb. 6.5 dargestellt graphisch erfolgen. Der Aufbau einer Audio-/Videoverbindung zu einem vorherigen Bearbeiter kann nun durch einfaches Anwählen der betreffenden Person erfolgen.

### Instanziierung der Referenzarchitektur

Ein Ziel bei der Instanziierung der Referenzarchitektur (vgl. Abb. 6.2) war es, einzelne Komponenten weitestgehend durch Produkte abzudecken. Neben dem Einsatz von Produkten waren aber auch verschiedene Eigenentwicklungen notwendig, insbesondere im Bereich der Benutzeroberfläche und der Integration von Workflow-Systemen und Telekooperationswerkzeugen.

Für das Workflow-Management, Daten-Management und Organisations-Management wurde das HP Produkt WorkManager/Workflow eingesetzt. Außerdem hat er einen Teil des Login-Managers der Referenzarchitektur übernommen. Als Videokonferenzwerkzeug wurde Communique! von

InSoft verwendet. Das Dokumenten-Sharing im Rahmen einer Videokonferenz erfolgt mit Hilfe von HP Shared X, das über einen eigenentwickelten Plugin mit InSoft Communique! integriert wurde. Die Kopplung der Workflow-Komponente mit dem Videokonferenz-Werkzeug, zur Unterstützung kontextbezogener Rückfragen wurde vollständig am IPVR entwickelt.

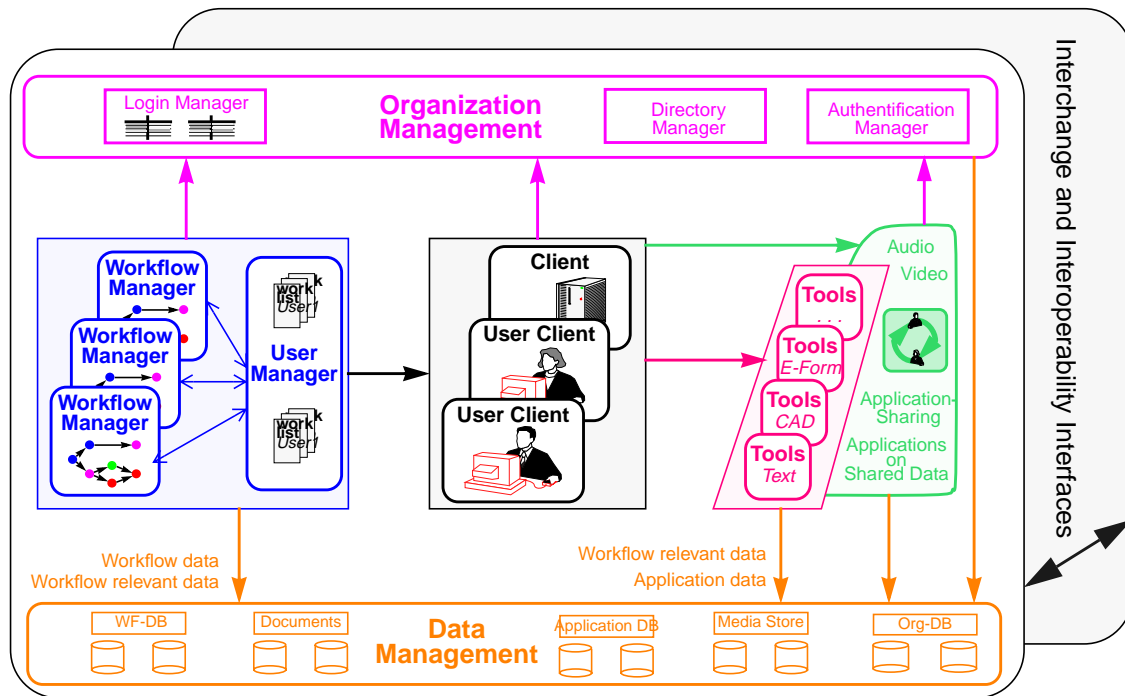


Abbildung 6.2: Instanziierung der Referenzarchitektur

### Prototyp I

Werden verschiedene Anwendungsprogramme gleichzeitig benötigt, so reicht der Platz auf dem Bildschirm häufig nicht mehr aus, um alle Fenster der Anwendungen benutzerfreundlich anzuordnen. Speziell im Rahmen einer Telekonferenz, während eine oder mehrere Anwendungen zwischen den Konferenzteilnehmern gemeinsam bearbeitet werden (application sharing), kommen noch entsprechende Fenster zur Sitzungssteuerung und die Anordnung der Fenster auf dem Bildschirm kann leicht sehr unübersichtlich werden. Aus diesem Grund stand für das Design der Benutzerschnittstelle zum Workflow-System eine platzsparende, übersichtliche Gestaltung, die eine intuitive Benutzung ermöglicht, mit an oberster Stelle der Anforderungsliste.

In Abb. 6.3 ist die Benutzeroberfläche eines Bearbeiters während der Bearbeitung einer Aktivität gezeigt. Neben der eigentlichen Benutzeroberfläche sind verschiedene Applikationen mit zur Bearbeitung geöffneten Dokumenten zu sehen.

Der Prototyp I bildet die Grundlage für den Prototyp II, für den verschiedene Komponenten wie der User Manager oder das User Interface überarbeitet oder weiterentwickelt wurden (vgl. auch Arbeitsbereich WR).

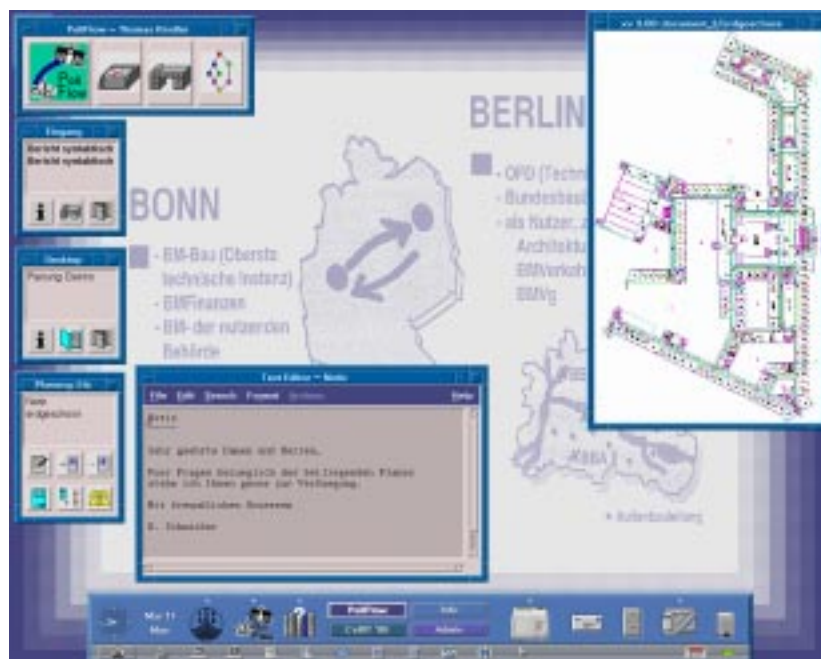
Das Hauptfenster besteht im wesentlichen aus drei, funktional unterlegten Icons (siehe Abb. 6.3 oben links):

### 1. Eingangskorb

Zur Bearbeitung anliegende Aktivitäten werden den betreffenden Personen durch einen Dokumentenstapel im Eingangskorb symbolisch angezeigt. Die Anzahl der anliegenden Arbeiten wird durch die Höhe des Stapels repräsentiert. Kommen weitere Aktivitäten hinzu, so wächst der Stapel, nimmt die Zahl ab, so wird der Stapel kleiner. Wird der Eingangskorb angeklickt, so wird ein Fenster geöffnet, in dem die einzelnen offenen Aufgaben angezeigt werden. Zum Eingangskorb eines bestimmten Benutzers gehören dabei alle anstehenden Aufgaben, die in seinen Zuständigkeitsbereich fallen und potentiell von ihm erledigt werden können. Dazu gehören also nicht nur persönlich an einen Benutzer gerichtete Aktivitäten, sondern auch an bestimmte Gruppen gerichtete Aktivitäten, zu denen der Benutzer gehört. Aktivitäten, die an eine Gruppe (oder Rolle) adressiert sind, erscheinen bei allen Mitgliedern dieser Gruppe (bzw. allen Rolleninhabern) im Eingangskorb. Nimmt eine dieser Personen die Aktivität zur Bearbeitung an, so verschwindet sie aus allen Eingangskörben und erscheint beim akzeptierenden Benutzer auf dem Schreibtisch.

### 2. Schreibtisch

Der Schreibtisch repräsentiert alle Aktivitäten, deren Bearbeitung im Verantwortungsbereich des Benutzers liegen. Die Anzahl der zu erledigenden Aufgaben wird, analog zum Eingangskorb, durch entsprechende Dokumentenstapel auf dem Schreibtisch symbolisiert.



**Abbildung 6.3:** Workflow Arbeitsplatz

### 3. Prozeßliste

Durch Anklicken wird ein Fenster geöffnet, indem eine Liste von Prozessen angezeigt wird, die vom Benutzer betrachtet oder gestartet werden können.

Wird eines dieser Icons angeklickt, so wird das zugehörige Fenster geöffnet. Diese Fenster enthalten eine Liste mit anstehenden Aktivitäten, angenommenen Aktivitäten bzw. Prozessen. Neben der Standard-Funktion zum Schließen des Fensters werden folgende Funktionen angeboten:

- Eingangskorb-Fenster
  - Info: Dient zum Anzeigen von Informationen zu den Aktivitäten im Eingangskorb.
  - Annehmen: nimmt die markierten Aufgaben an und
  - Schließen: schließt das Eingangskorbfenster
- Desktop-Fenster
  - Info: Dient zum Anzeigen von Informationen zu den Aktivitäten auf dem Schreibtisch
  - Öffnen: öffnet für alle markierten Aktivitäten ein Aktivitätsfenster (s.u.)
  - Schließen: schließt das Desktopfenster
- Prozeßlisten-Fenster
  - Ansehen: graphische Darstellung der markierten Prozesse
  - Starten: von den markierten Prozessen wird jeweils eine Instanz gestartet
  - Schließen: schließt das Prozeßlistenfenster

Wird im Desktopfenster eine Aktivität zur Bearbeitung geöffnet, so erscheint ein Fenster, in dem die zugehörigen Dokumente angezeigt werden. Dieses Aktivitätsfenster stellt folgende Funktionen zur Verfügung:

- Bearbeiten: die selektierten Dokumente werden mit den zugehörigen Anwendungen zur Bearbeitung geöffnet.
- Dokument löschen: löschen eines Dokuments aus einer Vorgangsmappe
- Dokument hinzufügen: hinzufügen eines Dokuments in die Vorgangsmappe
- Beenden: die Bearbeitung der Aktivität wird beendet und die Vorgangsmappe geschlossen. Sie bleibt jedoch weiterhin auf dem Schreibtisch des Bearbeiters und kann vom Desktopfenster aus zu einem späteren Zeitpunkt zur Weiterbearbeitung wieder geöffnet werden.
- Beenden und Weiterleiten: die Bearbeitung der Aktivität wird beendet, die Vorgangsmappe geschlossen und an den nächsten Bearbeiter weitergeleitet.
- Rückfrage: öffnet das Rückfragefenster, das unten beschrieben wird

Während der Bearbeitung einer Aktivität genügt es in der Regel, wenn der Benutzer die benötigten Dokumente über das Aktivitätsfenster im Zugriff hat. Veränderungen des Status von Eingangskorb und Schreibtisch sind zudem aus den graphischen Animationen im Hauptfenster



ersichtlich. Der große Vorteil der entwickelten Benutzeroberfläche liegt darin, daß der Benutzer bereits mit diesen beiden Fenstern die wichtigsten Informationen im Zugriff hat, die nur sehr wenig Bildschirmfläche in Anspruch nimmt.

### Kopplungskomponente zur Unterstützung von Rückfragen

Wird im Aktivitätsfenster das Telefon-Icon angeklickt, so kann im Kontext der Bearbeitung der jeweiligen Vorgangsaktivität eine Rückfrage an einen Bearbeiter einer vorangegangenen Aktivität getätigt werden. Es wird ein Fenster geöffnet, in dem die Instanz des zugehörigen Vorgangs graphisch dargestellt wird (siehe Abb. 6.4).



**Abbildung 6.4:** Rückfrage im Kontext eines Vorgangs

Neben den Aktivitäten, die durch die Knoten des Graphen repräsentiert werden, werden zusätzliche Informationen angezeigt, wie der Name des Bearbeiters, die Bezeichnung der Aktivität usw. Die Auswahl der Gesprächspartner erfolgt durch Anklicken der Bilder mit der Maus, ebenso der Start der Konferenz. Die Rückfrage-Komponente zeichnet sich insbesondere durch folgende Features aus:

- Einblenden der Bilder von Bearbeitern

Um auch soziale Kontakte zwischen verschiedenen Benutzer zu fördern, wurde die Darstellung von Vorgangsinstanzen so erweitert, daß nicht nur die Namen der jeweiligen Bearbeiter angezeigt werden, sondern auch deren Bild in das Symbol der entsprechenden Aktivität eingeblendet wird.

- **Dynamische Anzeige der momentanen Erreichbarkeit eines Benutzer per Videokonferenz**  
Eine weitere, sehr vorteilhafte Verbesserung ist die dynamische Anzeige der Verfügbarkeit von potentiellen Gesprächspartnern. Das Bild eines Bearbeiters wird nur dann eingeblendet, wenn dieser im Augenblick im System angemeldet und per Videokonferenz erreichbar ist.
- **Dynamisches Hinzunehmen weiterer Teilnehmer**  
Auch während einer Rückfrage-Konferenz kann der Personenkreis der Konferenzteilnehmer dynamisch erweitert werden. Die Hinzunahme einer Person, die an dem betreffenden Vorgang als Bearbeiter beteiligt waren, erfolgt durch entsprechendes Anwählen der Person in der Rückfrage-Komponente mit der Maus.

Als Werkzeug für Videokonferenzen wurde Communicue! von InSoft eingesetzt. Die Kopplungskomponente zur graphischen Darstellung der Vorgangsinstantz und zur Auswahl der Konferenzteilnehmer wurde am IPVR entworfen, implementiert und mit Communicue! integriert.

Zum gemeinsamen Betrachten und Bearbeiten von Dokumenten wurde ein Sharing-Plugin entwickelt, das im nächsten Abschnitt beschrieben wird.

### **Dokumenten-Sharing im Rahmen einer Videokonferenz**

Das von InSoft verfügbare Sharing-Plugin erfüllt einige wichtige Anforderungen aus dem Poli-Flow-Umfeld nicht, u.a. das Sharing von Applikationen nach deren Start bzw. die Kontrolle des Eingaberechtes in die Applikationen. Daher war es notwendig ein Sharing-Werkzeug an Communicue anzubinden, das diese Anforderungen erfüllt. Als Sharing-Werkzeug wurde HP SharedX verwendet, da es neben den oben genannten Anforderungen auch noch eine Schnittstelle zur externen Steuerung zur Verfügung stellt. Diese Schnittstelle ist für eine nahtlose Integration als Plugin in Communicue unerlässlich.

Das vom IPVR entwickelte Sharing-Plugin für Communicue bietet folgende Funktionalität:

- Automatische Übernahme der relevanten Konferenzinformationen aus Communicue, wie z.B. der Teilnehmerdaten (Rechner, Displayadresse usw.)
- Die Möglichkeit zur einfachen Initiierung des Sharings von Anwendungsfenstern bzw. ganzer Anwendungen mit mehreren Fenstern an alle Konferenzteilnehmer durch einfaches Anklicken der Fenster.
- Graphisch unterstützte Eingabekontrolle (floor control) für alle Anwendungsfenster auf Teilnehmerbasis (grant floor control / revoke floor control).
- Automatische Aktualisierung des Sharing-Werkzeuges, wenn sich der Konferenzzustand von Communicue ändert, z.B. wenn ein Teilnehmer die Konferenz verläßt wird das Sharing für alle Fenster, die zu ihm in Bezug stehen, beendet. Zu den Fenstern die zu dem Teilnehmer in Bezug stehen gehören sowohl die, die von dem Teilnehmer am Sharing beteiligt wurden, wie auch die Fenster die dem Teilnehmer durch das Sharing von anderen Konferenzteilnehmern zur Ansicht bzw. Bearbeitung zur Verfügung gestellt wurden.

Neben der reinen Funktionalität wurde die Benutzeroberfläche für das Sharing-Plugin so entworfen, daß es dem Look-and-Feel von Communicate entspricht, somit wurde neben der technischen Integration auch eine nahtlose Integration für die Benutzerinteraktion realisiert.

Durch die Eigenentwicklung des Sharing-Plugins stand PoliFlow nun ein Telekonferenzsystem zur Verfügung, das den wichtigsten Anforderungen entsprach, die aufgrund der Evaluierung in der ersten Phase des Projektes zu erfüllen sind.

### **Workflow-Adapter**

Um das System möglichst flexibel und offen zu gestalten, basiert der gesamte Systementwurf auf einem Adapter-Konzept. Die Aufgabe von Adaptern ist die Abbildung von standardisierten Schnittstellen auf produktspezifische Schnittstellen und umgekehrt. Innerhalb eines (wenn möglich kleinen) Systemkerns erfolgt die Kommunikation über standardisierte Schnittstellen. Produkte können nun mit Hilfe eines geeigneten Adapters in das System integriert werden. Diese Lösungsansatz bietet zwei Vorteile, die ihn besonders wertvoll machen:

- Heutige am Markt erhältliche, aber nicht standardisierte Produkte können mit Hilfe von Adaptern in das System integriert werden. Ein System, dessen Offenheit auf einem solchen Adapterkonzept beruht, soll hier als Pseudo-Plug-and-Play-System bezeichnet werden.
- Im Zuge zukünftiger Standardisierungen von Produkten können Adapter wegfallen und Produkte direkt integriert werden. Auf diese Weise erhält man ein echtes Plug-and-Play-System.

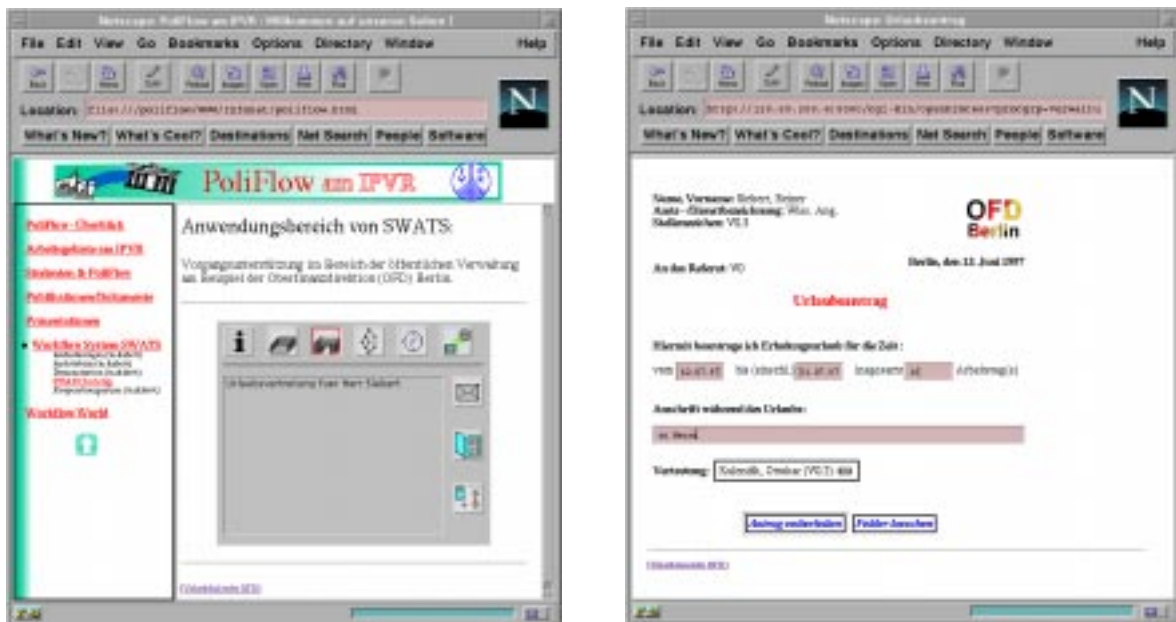
Damit bietet dieser Lösungsansatz sowohl das Potential, integrierte Lösungen auf der Basis heutiger Produkte aufzubauen, als auch eine Basis für zukünftige Systeme mit weitgehend standardisierte Komponenten.

### **Prototyp II**

Im Prototyp II wurden die Konzepte des Prototyps I an den WWW-basierten SWATS-Prototyp angepaßt und integriert. Im Prototyp wurde der Communicator von Netscape eingesetzt, der in der Version 4.x das Programm Conference enthielt, das die wichtigsten Werkzeuge integriert hat, um gemischte Arbeitsformen (point-to-point) zu unterstützen: So kann mit dem full-duplex Speakerphone-Modul eine Audiokonferenz über das Inter-/Intranet geführt werden. Das Shared Whiteboard ermöglicht es den Benutzern Grafiken gemeinsam anzuschauen und zu editieren. Collaborative Browsing erlaubt die Synchronisation der Web-Browser der Benutzer. Mit dem Text Chat Tool kann der Benutzer eingegebene Nachrichten oder ganze Text-Dateien senden und empfangen. Durch die Anwahl eines speziellen Links kann man so mit einem gewünschten Gesprächspartner eine Telekonferenz eröffnen, die allerdings bislang keine Videoübertragung realisiert.

## 6.5 WWW-basierte Vorgangsbearbeitung

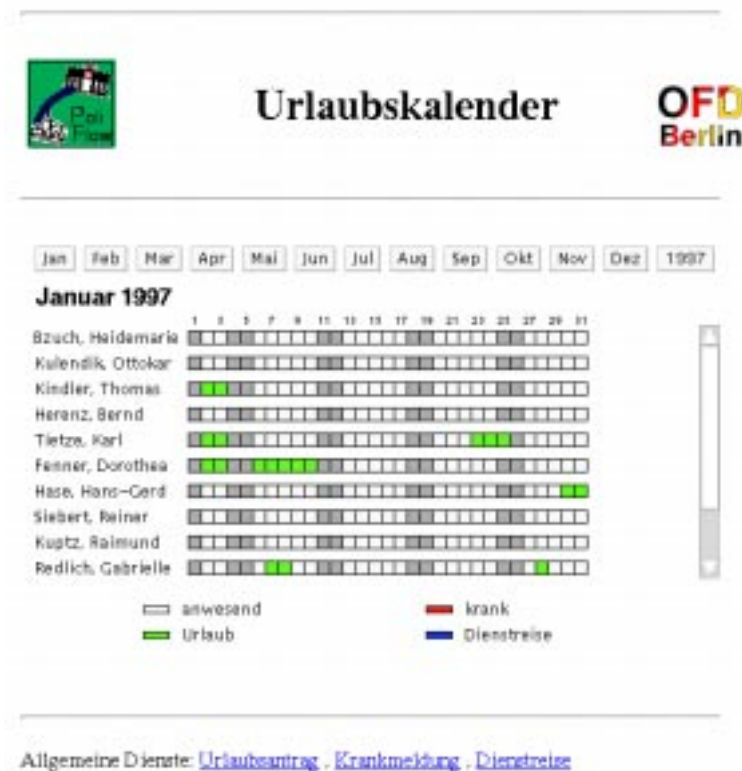
Durch die rasche Verbreitung der Internet-Technologie wurden auch die Rahmenanforderungen während der Projektlaufzeit entsprechend angepaßt. Um Plattformunabhängigkeit und Interoperabilität zu gewährleisten wurden bei der Entwicklung des zweiten Prototypen Internet-Technologie, wie Java, HTML, HTTP, CGI, sehr stark berücksichtigt. In MWM wurde insbesondere eine Menge von Basiskomponenten entwickelt, die Mechanismen anbieten, so daß die Bearbeitung von Aktivitäten eines Vorgangs über das Internet abgewickelt werden können. Für den Benutzer ergibt sich dadurch der entscheidende Vorteil, daß die Bearbeitung direkt über das Netz erfolgen kann und außer einem Java-fähigen WWW-Browser keine weitere Software lokal installiert werden muß (vgl. Abb. 6.5).



**Abbildung 6.5:** Workflow-Benutzungsschnittstelle und Urlaubsantragsformular

Als erstes Werkzeug zur Kooperation wurde ein Web-basierter Urlaubskalender und als Beispielvorgang der dazugehörige Urlaubsantragsvorgang entwickelt. Mitarbeiter eines Unternehmens können per WWW ihren Urlaubsantrag stellen, der zur weiteren Genehmigung an Stellvertreter und anschließend zum Vorgesetzten weitergeleitet werden. Der Antragsteller wird über die Entscheidung per e-mail benachrichtigt. Wurde der Urlaub genehmigt, so wird er automatisch vom System in einen Urlaubskalender eingetragen (vgl. Abb. 6.6), der eine Übersicht über die Urlaubstermine der Mitarbeiter gibt.

Der Web-basierte Urlaubskalender wurde zu einer vollwertigen Anwendung, dem WWW-basier-



**Abbildung 6.6:** Urlaubskalender

ten Gruppenkalender, weiterentwickelt, um die Terminplanung von Arbeitsgruppen zu unterstützen und die Einbindung in Verwaltungsprozesse zu gestatten. Wie das Workflow-User-Interface enthält er eine in Java implementierte Benutzungsschnittstelle, ist damit in die gewohnte Arbeitsumgebung eingebunden und erlaubt den verteilten Zugriff von heterogenen Plattformen aus. Der Kalender ermöglicht Mitgliedern von Arbeitsgruppen wie Abteilungen, Projekten etc. die Planung, Übersicht und Kontrolle von Terminen, indem er gruppenspezifische Einträge und Sichten anbietet. Mit Hilfe automatischer Aktionen in Aktivitäten kann etwa nach der Genehmigung eines Antrags der Termin ohne weitere Einwirkung des Anwenders in den Kalender eingetragen werden. Dadurch wird die Termininformation über die Anwender aktuell und konsistent verfügbar.

Durch den Kalender wird das Integrationspotential der WWW-basierten Architektur von SWATS (siehe Abschnitt zu WR) für Behörden mit einer Anwendung aufgezeigt, die Gruppenarbeit unterstützt. Mit der Web-Technologie HTML gelingt die Verknüpfung zwischen dem Workflow-System und dem Informationssystem Gruppenkalender: Beim Start oder bei der Bearbeitung von Dienstreise- oder Urlaubsanträgen kann aus den Formularen auf den Kalender zugegriffen werden; umgekehrt kann man aus den Web-Seiten, in die der Kalender eingebettet wird, über Verweise Vorgänge starten. Diese Verwendung gemeinsamer Datenbestände im Beispiel der Prozeß-

realisierung Urlaubsantrag macht deutlich, wie durch die offene Architektur von SWATS Redundanz, das heißt in diesem Fall die mehrfache Verwaltung freier Termine, vermieden und Erweiterungen des Systems um neue Anwendungen erleichtert werden.

Ein typisches Problem, das auch im Anwendungsfeld häufig auftritt, ist, daß mehrere Personen gleichzeitig auf gemeinsame Datenbestände zugreifen müssen. Bei der Erstellung der HU-Bau werden beispielsweise die Pläne des Architekten von Elektro- und Sanitärinstallationsbetrieben benötigt. Dabei ist es notwendig, daß allen Beteiligten die Pläne in der jeweils aktuellsten Fassung vorliegen. Ansonsten kommt es leicht zu Mißverständnissen, die zu Zeitverzögerungen in der Fertigstellung oder zu zusätzlichen Kosten führen können.

Um dies zu vermeiden, muß der gleichzeitige Zugriff auf Datenbestände gewährleistet werden. Darum wurden in einem nächsten Schritt erste Konzepte für ein kooperatives synchrones Werkzeug entwickelt. Als Beispielanwendung wurde ein synchroner Texteditor ausgewählt.

Mit diesem Editor können Texte gemeinsam von verschiedenen Arbeitsplätzen aus editiert werden. Da auch hier die Implementierung vollständig in der Programmiersprache Java erfolgte, kann die Bearbeitung sowohl von einem PC-Arbeitsplatz aus erfolgen, aber auch von einer UNIX-Workstation. Der kooperative Texteditor zeigt einem Benutzer alle anderen Benutzer an, die gerade dasselbe Dokument editieren. Wird ein Bild eines dieser Benutzer angeklickt, so erscheint ein Popup-Menü, in dem verschiedene Kommunikationsmechanismen (asynchron und synchron) mit der betreffenden Person angeboten werden und ausgewählt werden können.

Diese Konzepte können weiter ausgebaut werden, so daß verschiedene Synchronisationsmechanismen zum Einsatz kommen, unterschiedlichen Rollen der beteiligten Benutzer unterschiedliche Zugriffsrechte zugewiesen werden. Außerdem lassen sich die Konzepte auch auf andere Bereiche übertragen, wie z.B. CAD, wo kooperatives Arbeiten häufig eine sehr viel wichtigere Rolle als beim gemeinsamen Erstellen von Textdokumenten spielt.

## 6.6 Literatur

- [Berg93] Bergmann, N.W., und Mudge, J.C (1993). A Survey of Multi-media Computing and Communications Methods for Remote Collaboration. MCAT93, Wollongong, Australia, 14.-15. Juli 1993.
- [Boch94] Bochenski, B. (1994). Implementing Production-Quality Client/Server Systems. John Wiley & Sons, Inc., Kapitel 19, S. 319-325.
- [Elli91] Ellis, C., Gibbs, S., und Rein, G. (1991). Groupware - Some Issues and Experiences. *Communications of the ACM*, 34(1):39-58.
- [Hase93] Hasenkamp, U. und Syring, M. (1993). Workflow Management : Beispiele Zeitschriftenproduktion. *Office Management*, (6):32-36.

- [Karc93] Karcher, H. (1993). Workflow und Groupware als Gegenpole im Office-Kontinuum. *Computerwoche* (39): S31-35.
- [Karl93] Karl, R. (1993). Workflow Management : Prozessorientierte Vorgangsbearbeitung. *Office Management*, (3):45-47.
- [Lewe91] Lewe, H. und Krcmar, H. (1991). Groupware-Das aktuelle Schlagwort. *Informatik-Spektrum*, 14:345-348.
- [Nava92] Navarro, L., Prinz, W., und Rodden, T. (1992). Towards Open CSCW Systems. In *IEEE-1992*, Seiten 4-10.
- [Robi94] Robinson, J.A. (1994). Communications services architecture for CSCW. *Computer Communications*, (17): 339-347, Mai 1994.

## 7 Konsistenzsicherung und Zuverlässigkeit in WFMS (CAM)

Das Projekt PoliFlow befaßt sich mit der zuverlässigen und sicheren Vorgangsbearbeitung weitverteilter Anwendungsumgebungen unter Berücksichtigung gemischter Arbeitsformen. Durch den Regierungsumzug von Bonn nach Berlin ergibt sich einerseits ein erhöhter Aufwand durch die Verlegung eines großen Teils der Regierung von Bonn nach Berlin. Andererseits wird es durch den doppelten und verteilten Regierungssitz in Berlin und Bonn notwendig, die räumlich weitverteilte Verwaltung durch ein innovatives System zur Vorgangsbearbeitung so zu unterstützen, daß die Arbeit effizient und kostengünstig durchführbar ist. Ein System zur effizienten automatischen Steuerung und Koordination von Vorgängen in einem Unternehmen bzw. einer Verwaltung ist ein Workflow-Management-System (siehe [Jab95], [Rei94]). Die Integration von Telekooperationstechniken sowie die Unterstützung weiterer Formen kooperativer Prozesse ist zusätzlich erforderlich, um die Mitarbeitenden in der verteilten Umgebung optimal zu unterstützen.

Der Einsatz von Workflow-Management-Systemen, insbesondere zur Ausführung und Steuerung der unternehmens- bzw. verwaltungskritischen Prozesse, stellt jedoch höchste Anforderungen an die Korrektheit der Abläufe bzw. die Zuverlässigkeit und Verfügbarkeit der Systeme. Der Verlust oder die Inkonsistenz von Daten, von Informationen sowie von Abläufen ist nicht tolerierbar und selbst wenn ein konsistenter Zustand wiederhergestellt werden kann, muß dies schnell geschehen, um mit der Bearbeitung von Vorgängen rasch fortfahren zu können. Eine gleichzeitige Bearbeitung der Vorgänge auf die herkömmliche Art und Weise, beispielsweise durch Führung von Akten, und durch Rechnerunterstützung ist nicht praktikabel. Der Aufwand ist zu hoch, und die Fehlermöglichkeiten werden vergrößert. Die Anforderungen an die Sicherheit, die Verlässlichkeit, die Integrität, die Verbindlichkeit, die Vertraulichkeit von Daten sowie die Rückverfolgbarkeit von Vorgängen können ohne die Sicherstellung der Zuverlässigkeit der Systeme nicht erfüllt werden.

Workflow-Management-Systeme müssen in der Lage sein, verlässliche Zusicherungen über ihr Verhalten, gerade auch in Fehler- und Ausnahmefällen sowie in Konkurrenzsituationen, zu geben. Das bedeutet, die Konsistenzsicherung muß so weit wie möglich vom System übernommen werden. Heutige Workflow-Management-Systeme werden diesen Anforderungen noch nicht gerecht. Die Sorge der Konsistenzsicherung der Abläufe bleibt noch weitestgehend den Anwendungen selbst überlassen. Die Verfahren zur Konsistenzsicherung langandauernder Vorgänge sind auch heute noch aktiver Gegenstand der Forschung. Arbeiten dazu finden sich unter anderem in [Elm91] und [JK97].



Die Arbeitspakete CAM (Consistency Assuring Mechanisms in Workflow Systems) haben Verfahren zur Konsistenzsicherung für Workflows zum Inhalt. Der vorliegende Teilbericht gibt eine Übersicht über die Ergebnisse der Arbeitspakete CAM.1 bis CAM.4 von 1994 bis September 1996. Diese Arbeitspakete befaßten sich mit der Anforderungsanalyse an die Verfahren zur Konsistenzsicherung sowie der Auswahl und der Identifikation eines Basismodells. Das zu dieser Zeit gewählte und betrachtete Anwendungsszenario umfaßte Prozesse in der Verwaltung, welche sich mit der Planung, Erfassung und Allokation der notwendigen Gebäude in Berlin befassen.

Im vorliegenden Teilbericht werden nach einem kurzen Überblick über die Prinzipien der Konsistenzsicherung die Ergebnisse der Arbeitspakete CAM skizziert. Das definierte Basismodell, notwendige Weiterentwicklungen sowie Aspekte der Kompensation werden erläutert. Das realisierte Werkzeug zur Modellierung von Kompensation in Workflow-Management-Systeme wird skizziert. Der Bericht endet mit einer Diskussion und Bewertung.

## **7.1 Prinzipien der Konsistenzsicherung**

Eine hohe Zuverlässigkeit und Verfügbarkeit eines heterogenen und weitverteilten Systems zur Vorgangsbearbeitung ist eine wesentliche Voraussetzung für dessen Nutzen und Akzeptanz. Ein zuverlässiges System hat eine geringe Wahrscheinlichkeit für ein Verhalten, das nicht dem spezifizierten und gewünschten Verhalten des Systems entspricht [GR93]. Die durchschnittliche Zeit bis zu einer Störung, beispielsweise einem Betriebsausfall, ist hoch. Zur Bestimmung der Verfügbarkeit eines Systems wird zusätzlich die durchschnittliche Zeit, die für eine Reparatur (Recovery) benötigt wird, berücksichtigt. Die Erhöhung der Zuverlässigkeit eines Systems wird dementsprechend entweder durch die Vermeidung bzw. Verbesserung der latenten Fehler erreicht oder durch die Korrektur der Wirkungen der effektiven Fehler auf das Verhalten des Systems. Wird gleichzeitig die notwendige Reparaturzeit minimiert, so wird auch die Verfügbarkeit des Systems erhöht.

Die Anzahl der latenten Fehler kann zwar schon im Vorfeld durch eine sorgfältige Konstruktion des Systems vermindert werden. Dies wird jedoch erschwert durch die immer größer und komplexer werdenden Systeme und Anwendungen. Eine wesentliche Rolle spielt hierbei die steigende Komplexität der notwendigen Software. Software Fehlerursachen sind heutzutage die Hauptursachen der effektiven Fehler. Andere Fehler können maskiert werden durch eine Kombination von Redundanz, geographischer Verteilung sowie Software zum Automatisieren der Dienste. Die Software Fehler bleiben ein ungelöstes Problem [GR93]. Da das Vorkommen der latenten Fehler

in einem System sowie die Störungen außerhalb des Systems und somit auch das Auftreten effektiver Fehler nicht zu verhindern ist, ist es notwendig, die Wirkungen der effektiven Fehler auf das Systemverhalten zu korrigieren und so eine hohe Zuverlässigkeit des Systems zu gewährleisten.

Verfahren zur Sicherstellung der Konsistenz der Anwendungen werden benötigt. Die grundsätzliche Problemstellung ist die parallele Verarbeitung von Vorgängen (Prozessen) unter Verwendung gemeinsamer Betriebsmittel. Hier lassen sich zwei Problembereiche unterscheiden. Einerseits ist es notwendig Mechanismen zur Synchronisation der Prozesse bereitzustellen, um gegebenenfalls einen logischen Einbenutzerbetrieb zu ermöglichen. Je nach den zu unterstützenden Kooperationsformen sind hierbei jedoch unterschiedliche Isolationsbedürfnisse der Anwendungen zu beachten. Andererseits sind Mechanismen zur Fehler- und Ausnahmebehandlung notwendig. Weder Daten noch Abläufe dürfen durch Systemausfälle, Kommunikationsfehler, Fehler während der Bearbeitung von Vorgängen oder beliebige andere Ausnahmen verloren gehen oder inkonsistent werden.

Besonders die Fehlerbehandlung ist keine triviale Aufgabe und je komplexer die Anwendungen und Systeme sind, umso schwieriger sind die notwendigen Maßnahmen zu durchschauen, zu implementieren und zu warten, ganz besonders dann, wenn sie in den Anwendungsprogrammen verstreut vorliegen. Zur Sicherung der Zuverlässigkeit der Anwendungen ist es notwendig, die Fehlerbehandlung so weit wie möglich aus den Anwendungsprogrammen herauszulösen und durch das System bereitzustellen. Dies bietet eine Reihe von Vorteilen. Die Anwendungsprogrammiererinnen werden entlastet, die Programme werden einfacher, und somit wird die Wahrscheinlichkeit von Fehlern verringert. Auf die Fehlerbehandlung spezialisierte Personen können diese Mechanismen entwerfen, implementieren und warten. Dadurch können auch Konsistenzgarantien zugesichert werden. Die Mechanismen werden wiederverwendet, getestet und verbessert. Das Wissen ist dementsprechend global zugreifbar und nutzbar. Das Bereitstellen von Basismechanismen für die anwendungsunabhängigen Anteile zur Fehlerbehandlung verringert somit wiederum die möglichen Fehlerursachen und erhöht die Zuverlässigkeit des Systems. Besonders effektiv ist dies, wenn die Menge der möglichen Fehler in wenige Klassen eingeteilt werden kann und dementsprechend Basismechanismen für diese Klassen ausreichen, um die vielfältigen (erwarteten) Fehler zu behandeln.

Ein sehr erfolgreicher Ansatz zu diesem Zweck ist das klassische Transaktionskonzept ursprünglich aus dem Bereich der Datenbankverwaltungssysteme. Die Transaktionsverarbeitung hat wegweisende Konzepte für verteilte und fehlertolerante Systeme bereitgestellt. Der Einsatz von ACID-Transaktionen verhindert Mehrbenutzeranomalien bei einer konkurrierenden Ausführung, schränkt außerdem die Auswirkungen der effektiven Fehler so weit wie möglich ein und erleichtert die Fehlererkennung. Er gewährleistet zudem im Fehlerfall die (automatische) Wiederherstellung eines möglichst aktuellen und korrekten, d.h. als konsistent definierten, Zustands des Systems vor Eintreten des Fehlers, so daß die durch den Fehler unterbrochenen Dienste erneut

angeboten werden können. Die durch einen Fehler unterbrochenen ACID-Transaktionen werden zurückgesetzt. Das heißt, ihre inkonsistenten Zwischenergebnisse wurden (anderen konkurrierenden Transaktionen) nicht sichtbar, und sie werden wieder vollständig aus dem System entfernt. Die grundsätzliche Idee ist es, die Anwendungsprogramme in wohldefinierte Teile, die semantisch korrekte Übergänge zwischen konsistenten Zuständen darstellen, aufzuspalten.

Zur Sicherung der Zuverlässigkeit der Anwendungen in der vorgesehenen Anwendungsumgebung ist das klassische Transaktionsmodell jedoch nicht anwendbar bzw. nicht ausreichend.

## 7.2 Eigenschaften der Anwendung

Durch eine Analyse der Anwendungsumgebung bzw. der Vorgänge, wurden viele Merkmale der Anwendungen festgestellt (siehe Zwischenbericht 1), welche die notwendigen Verfahren zur Konsistenzsicherung beeinflussen. Wesentliche Eigenschaften von solchen langandauernden Vorgängen werden im folgenden kurz skizziert (siehe auch [RSS97]):

- Langandauernde Vorgänge sollten oder können nicht im transaktionalen Sinn zurückgesetzt und wiederholt werden, da dies beispielsweise zu teuer ist oder daraufhin eine kritische Deadline überschritten würde. Sie verlangen eine vorwärts-orientierte Fehlerbehandlung.
- Ein langandauernder Vorgang muß Systemausfälle, Reorganisationen des Systems oder andere Unterbrechungen, d.h. sogar einen Ausfall aller teilnehmenden Klienten, überleben.
- Langandauernde Vorgänge beteiligen viele Klienten und diese auch gleichzeitig.
- Langandauernde Vorgänge sind nicht unbedingt von vorneherein vollständig spezifizierbar.

Eine vorwärts-orientierte Fehlerbehandlung von Vorgängen dieser Art ist unbedingt notwendig, da sonst der Verlust der Arbeit von Stunden, Tagen bis hin zu Jahren in Kauf genommen werden müßte. Gleichzeitig sollte die Möglichkeit zur vollständigen oder teilweisen Stornierung eines Vorgangs gewährleistet werden. Bezüglich der Synchronisation ist eine vollständige Isolation dieser langandauernden Prozesse untereinander zu restriktiv (siehe Zwischenberichte 2 und 3). Zudem müssen die kooperativen Anteile der Vorgänge geeignet unterstützt werden. Ein erweitertes Modell zur Trennung der anwendungsunabhängigen Anteile der Fehlerbehandlung und Synchronisation von den Anwendungsprogrammen ist notwendig. Einen Vorgang auf eine ACID-Transaktion abzubilden ist nicht möglich. Insgesamt ist ein modularer Aufbau der Vorgänge und eine damit auch einhergehende feinere Recovery-Granularität notwendig. Der Kontrollfluß innerhalb der Vorgänge muß explizit sein und geschützt werden. Aus Effizienzgründen sollte sowohl der Grad der Intertransaktionsparallelität, als auch der Grad die Intratransaktionsparallelität maximiert werden.

## 7.3 Erweiterte Transaktionsmodelle

In den letzten Jahren wurde eine Reihe von erweiterten Transaktionskonzepten für die unterschiedlichsten Anwendungsfelder vorgestellt. Es war zu untersuchen inwieweit diese Modelle und ihre Mechanismen zur Fehlerbehandlung in dem vorgesehenen Anwendungsumfeld einsetzbar bzw. ihr Einsatz sinnvoll ist. Wesentliche Konzepte (Sagas [GMS87], Flex-Transaktion [LEB90] [RELL90] [BEK93] [ZNBB94] [JNRS93] [Elm91], ConTracts in [Elm91] [Schw94] [RS95], Split-Transaktionsmodell in [Elm91], S-Transaktionen in [Elm91], Polytransaktionen [Elm91], DOM [Elm91] [KGH93] [GHS95] [GHM+93], Mehrschichtentransaktionen in [Elm91], Toolkit-Ansatz in [Elm91] [Unl94], NT/PV-Modell [KS94], Trigger und Transaktionen" [DHL90], kooperative Transaktionshierarchie in [Elm91]) wurden analysiert und auf ihre Eignung für das vorgesehene Anwendungsumfeld hin betrachtet (siehe Zwischenbericht 2).

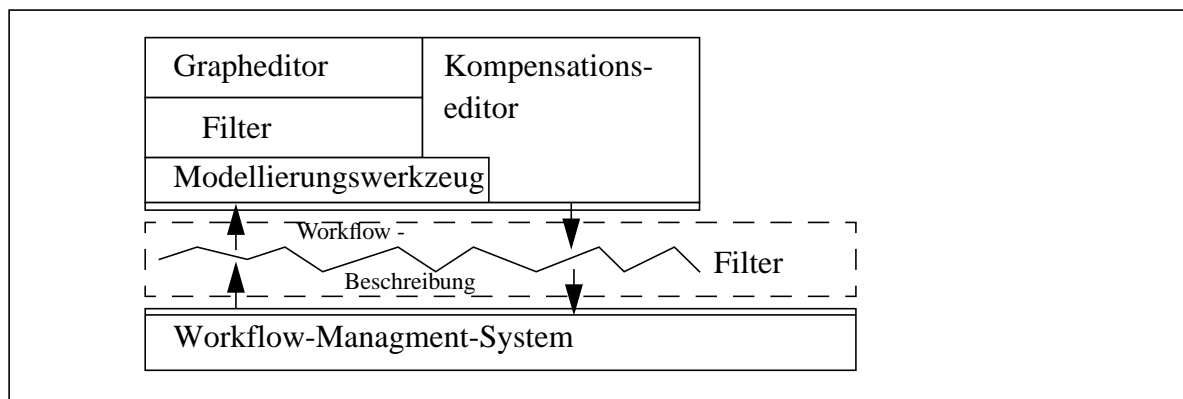
## 7.4 Definition eines Basismodells

Im Rahmen des Arbeitspaketes CAM.4 wurde ein Basismodell für die zuverlässige Ausführung von Workflows definiert. Durch die Analysen innerhalb der Arbeitspakete im Bereich „Konsistenzsichende Verfahren“ hat sich gezeigt, daß das ConTract-Modell ein besonders flexibler Ansatz ist und sich unter anderem durch das zweistufige Programmiermodell sehr gut zum Einsatz in Workflow-Management-Systemen eignet. Die speziellen Anforderungen der Anwendungen im Bereich der Vorgangsbearbeitung in der öffentlichen Verwaltung erfordern jedoch noch einige Erweiterungen dieses Modells, welche das ConTract-Modell aufgrund seiner Struktur ermöglicht. Zusätzlich werden beispielsweise auch Konzepte bzw. Ansätze der Flex-Transaktionen zur Behandlung von nicht-kompensierbaren Aktivitäten miteinbezogen, da sie als geeignet in diesem Zusammenhang identifiziert wurden [BEK93]. Sie werden unter dem Abschnitt Fixpunkte näher beschrieben. Und schließlich werden Split-Transaktionen (in [Elm91]) für die kooperativen Anteile der Vorgänge in Erwägung gezogen.

Das ConTracts zum Einsatz in Workflow-Management-Systemen sehr gut geeignet ist, und in dem PoliFlow-Prototypen nach 1996 auch genutzt hätte werden können, was jedoch Erweiterungen innerhalb der jeweiligen Workflow-Engine notwendig macht, zeigt sich auch durch die Entwicklung von ConTracts nach 1996. Das ConTract-Modell wurde 1997 erweitert (siehe [RSS97]) und seine Anwendbarkeit zur zuverlässigen Ausführung von Workflows wurde durch eine neue prototypische Implementierung des Modells, APRICOTS (A PRototype Implementation of a COnTrat-System) [Schn98], als transaktionales Workflow-Management-System auf der Basis von CORBA gezeigt.

Die Durchführung der Kompensation von erfolgreich ausgeführten Transaktionen oder anderen realen Aktionen hat sich als ein notwendiger Teil der Fehlerbehandlung für langandauernde Prozesse herausgestellt (siehe auch [WR91], [Ley95]).

Als erste zu realisierende Komponente im Bereich Konsistenzsicherung für PoliFlow wurde daher 1996 ein Werkzeug zur Modellierung von Kompensation für Workflow-Management-Systeme [MD97] prototypisch entwickelt, welches einerseits die Workflow-Modellierung von Montana zugrundelegte. Andererseits sollte es jedoch auch für ein eventuell anderes in PoliFlow einzusetzendes Workflow-Management-System geeignet sein. Zur Anpassung an "beliebige" Workflow-Management-Systeme muß ein Filter implementiert werden (siehe Abb. 1) und das Workflow-Management-System muß schließlich die Kompensation ausführen können, d.h. die erweiterte Workflow-Beschreibung interpretieren können.



**Abbildung 7.1:** Werkzeug zur Modellierung von Kompensation

Im Modellierungswerkzeug wurde sowohl die Kompensation für erfolgreich bearbeitete Aktivitäten, als auch die Kompensation als Recovery-Maßnahme für durch Fehler unterbrochene nicht-transaktionale Aktivitäten miteinbezogen. Im Gegensatz zum damaligen ConTract-Modell wurde die Kompensation flexibler modelliert und konnte somit den Anforderungen der analysierten Vorgänge besser gerecht werden. Gleichzeitig wurden Konzepte der Flex-Transaktionen miteinbezogen (siehe [MD97]).

Ein erster Korrektheitsbegriff für Vorgänge wurde definiert (siehe [MD97]). Ein Vorgang ist konsistenzerhaltend, wenn er vollständig, dauerhaft und semi-isoliert ist. Ein Vorgang ist vollständig, wenn er unter allen Umständen entweder selbst zu Ende geführt werden kann und in einem zulässigen Zustand endet oder in einen anderen vollständigen Vorgang überführt werden kann. Jedes System muß einen leeren Vorgang enthalten, der dadurch gekennzeichnet ist, daß die Menge seiner Aktivitäten leer ist. Der leere Vorgang sei immer vollständig. Die Vollständigkeit besagt somit, daß unzulässige Zustände immer nach endlicher Zeit verlassen werden. Ein Vorgang ist

dauerhaft, wenn seine Ergebnisse nicht mehr verloren gehen, sobald er vollständig abgelaufen ist und ein Vorgang ist semi-isoliert, wenn seine unzulässigen Zustände für die Bearbeiter anderer Vorgänge unsichtbar sind.

Im folgenden werden nach einem Überblick über das ConTract-Modell die notwendigen Erweiterungen kurz erläutert. Daraufhin wird auf die Kompensation eingegangen, welche auch heute noch aktiver Gegenstand der Forschung ist. Schließlich werden die im Rahmen der durchgeführten Anforderungsanalyse als notwendig erkannten Bereiche zur Integration und Ausführung von Kompensationen in einem Workflow-Management-System skizziert.

### 7.4.1 Das ConTract-Modell

Das ConTract-Modell erweitert die Kontrollmöglichkeiten klassischer Transaktionen und verbindet sie so mit Betriebssystemdiensten, daß der Anwendungsebene auf hohem Abstraktionsniveau eine Programmierumgebung zur Erstellung und fehlertoleranten Ausführung paralleler und verteilter Abläufe zur Verfügung steht [RW90]. Das Ziel ist es ACID-Transaktionen, wann immer es möglich ist zu nutzen. Dies gilt sowohl innerhalb des Systems, als auch explizit, d.h. Aktivitäten oder Gruppen von Aktivitäten als ACID-Transaktionen auszuführen, sofern ihre Eigenschaften dies erlauben.

Der Ansatz des ConTract-Modells ist es, die Anwendungs- und Kontrollaspekte zu entkoppeln und getrennt zu beschreiben. Der Ablauf eines Vorgangs wird durch ein Skript beschrieben. Das Skript ist ein rein administratives Programm, das die Ausführung der operativen Programme steuert und koordiniert, selbst jedoch keine Operationen auf der Datenwelt des Unternehmens ausführt. Im Skript schlagen sich auch viele dynamische Integritätsbedingungen nieder (zeitliche Abhängigkeiten, Organisationsvorschriften, usw.), die auf diese Weise nachvollziehbar und bei einer korrekten Abwicklung des Skripts ohne weitere Überprüfung automatisch eingehalten werden [RW90]. Die Programmierung der elementaren Berechnungsschritte bzw. die Beschreibungen der Aktivitäten innerhalb der Anwendungen geschieht innerhalb der Steps. Die Steps führen die Operationen auf der Datenwelt des Unternehmens aus. Die Steps werden für die Codierung der Verarbeitungsschritte mit den algorithmischen Teilen des Problems genutzt und stellen die eigentliche Anwendungsprogrammierung dar. Die Anwendungsprogrammierer können dadurch auf dieser Ebene von Aspekten bezüglich Verteilung, Asynchronität, Zugriffskonflikten, Fehlerzuständen usw. weitestgehend abstrahieren.

Die Zuverlässigkeitszusicherung des ConTract-Modells beruht auf der folgenden zentralen Konsenzeigenschaft: Ein ConTract terminiert unter (System-)Garantie in endlicher Zeit und in einem korrekten Endzustand. Es gelten folgende Grundannahmen. Alle Fehlersituationen in Subsystemen müssen zur Ebene der Anwendungsprogrammierung hin verborgen werden. Fehler, die

in dem System selbst nicht behandelt werden können, führen notwendigerweise zum automatischen Stoppen der Anwendung bzw. auf expliziten Wunsch der Anwendung eventuell zur geregelten Stornierung (Kompensation) des ConTracts, d.h. die Anwendungsprogrammierung ist auch hier nur indirekt betroffen. Aspekte der Nebenläufigkeit, Asynchronität, Betriebsmittelzuordnung usw. müssen zur Ebene der Anwendungsprogrammierung hin verborgen werden. Ein verteiltes System ist nur dann programmierbar (beherrschbar), wenn es zum Anwendungsprogrammierer, d.h. in unserem Fall zum Workflow-Programmierer, hin wie ein zentrales System aussieht.

Das ConTract-Modell sieht zur Behandlung von Fehlern ein Forward-Recovery auf der Skript-Ebene vor. Die durch den Fehler abgebrochenen transaktionalen Steps werden automatisch zurückgesetzt und wiederholt. Lediglich in den Fällen, in denen auch nach einer definierten Anzahl von Wiederholungen keine erfolgreiche Durchführung eines unterbrochenen Dienstes möglich und auch keine Alternative ausgeführt werden kann, ist per Default ein manuelles Eingreifen eines Systemadministrators erforderlich.

Auf ausdrücklichen Wunsch des Anwenders bzw. explizit durch die Anwendung, kann der gesamte Vorgang storniert werden. Dies geschieht durch die Durchführung von Kompensationsaktivitäten für die erfolgreich ausgeführten und somit dauerhaften Aktivitäten. Dies ist lediglich unter Nutzung semantischer Anwendungsinformation durch die Ausführung von Kompensationsaktivität möglich. Um jederzeit die Stornierung eines ConTracts ermöglichen zu können, muß zu jedem Zeitpunkt der Ausführung eines ConTracts die Existenz, die Ausführbarkeit und die Eindeutigkeit einer Kompensation für den aktuellen Teilvorgang, d.h. für alle bisher erfolgreich durchgeführten Aktivitäten, garantiert werden. Die Existenz einer Kompensation zu jedem Zeitpunkt kann beispielsweise garantiert werden, wenn für jeden Step ein Kompensationsstep definiert sein muß.

Die einzelnen transaktionalen Steps werden von dem darunter liegenden Basissystem wie traditionelle Transaktionen synchronisiert. Die Sperren werden am Ende einer Transaktion wieder freigegeben. Somit können ohne zusätzliche Maßnahmen inkonsistente nicht serialisierbare Abläufe entstehen. Deshalb werden zusätzlich Invarianten eingesetzt, die die parallel ablaufenden ConTracts über Prädikate synchronisieren. Die Synchronisationsanforderungen der Anwendungen werden unter Verwendung der Exit- und Entry-Invarianten explizit modelliert. Prädikatsperren bzw. Escrow-orientierte Verfahren ermöglichen einen hohen Parallelitätsgrad und schließen gleichzeitig Konsistenzverletzungen aus (siehe [Schw94]). Das Korrektheitskriterium ist die invariantenbasierte Serialisierbarkeit [RSS97].

## 7.4.2 Erweiterungen des ConTract-Modells

Die als notwendig erkannten Weiterentwicklungen, Erweiterungen oder Anpassungen des ConTract-Modells für die Anwendungsumgebung beziehen sich im wesentlichen auf die folgenden Bereiche:

- Integration und Behandlung von nicht-transaktionalen (bspw. nicht-konformen oder autonomen) Aktivitäten: Gerade in den untersuchten Vorgängen der öffentlichen Verwaltung enthalten Aktivitäten häufig reale Aktionen (Zwischenbericht 2). Diese Aktivitäten können nicht als klassische Transaktionen modelliert werden, da beispielweise ihre Atomarität nicht garantiert werden kann. Interaktionen mit der Umwelt sind in der Regel nicht transaktional zurücksetzbar. Beim Versenden eines Briefes verliert das System die Kontrolle über das bearbeitete Objekt. Es ist also notwendig, auch nicht-transaktionale bzw. nicht-konforme Aktivitäten zu unterstützen und auch hier weitestgehende Konsistenzgarantien zu geben.
- Integration und Behandlung von kooperativen Aktivitäten: Zur Realisierung der kooperativen Aktivitäten könnte das Split-Transaktionsmodell (in [Elm91]) verwendet werden.
- Erweiterte Fehlerbehandlungsstrategien: Einerseits werden durch die Integration nicht-transaktionaler Aktivitäten zusätzliche Fehlerbehandlungsstrategien notwendig. Der Default-Fall ist ein manuelles Forward- oder Backward-Recovery der unterbrochenen nicht-transaktionalen Aktivität durch einen Systemadministrator. Sinnvolle Strategien zur automatischen Forward- oder Backward-Recovery sind gegebenenfalls zu modellieren und zu unterstützen.
- Eine flexiblere Kompensation: Für Workflows erweist sich die Methode die Existenz eines Kompensationssteps für jeden Step eines ConTracts zu fordern in vielen Fällen als ungeeignet und zu unflexibel. Es kann sinnvoll sein, gesamte Gruppen von Aktivitäten, d.h. Kompensationsbereiche bzw. Kompensationssphären, durch eine oder wenige Kompensationsaktionen rückgängig zu machen. Beispielsweise muß nicht ein Dokument Stück für Stück zurückgenommen werden, wenn dies durch ein Löschen oder ein als kompensiert Markieren des gesamten Dokumentes deutlich einfacher und schneller durchzuführen ist. Bei überlappenden Kompensationsbereichen müssen Regeln definiert sein, welche die Auswahl der tatsächlich auszuführenden Kompensationen ermöglichen.
- Partielle Kompensation: Bei der Durchführung von partiellen Kompensationen sind noch Fragen offen. Die Möglichkeit zur partiellen Kompensation von Vorgängen ist für die vorgesehene Anwendungsumgebung jedoch sehr nützlich bzw. notwendig. Es ist sinnvoll, in manchen Fällen lediglich einen Teil des Vorgangs d.h. einige Aktivitäten zu kompensieren und danach wieder mit der Bearbeitung des Vorgangs fortzufahren. Als ein Beispiel für eine partielle Kompensation kann das Zurückgehen auf einen vorherigen Planungszustand in der Planungs-



phase der HUBau genannt werden. Durch regelmäßige Absprachen mit dem Nutzer, können Fehlplanungen früh erkannt und durch eine partielle Kompensation des Workflows schnell und effizient korrigiert werden.

- Die Behandlung von externen Ereignissen (Events) muß untersucht werden. Als ein wichtiges externes Ereignis ist beispielsweise das Zeitereignis „Datum“ zu nennen. Dadurch können Termine modelliert werden, so daß Terminüberschreitungen während der Ausführung automatisch erkannt und darauf reagiert werden kann.

## 7.5 Aspekte der Kompensation

Im Bereich der Kompensation ist auch heute noch viel Forschungsarbeit zu leisten. So sind die Details der Modellierung und Ausführung von Kompensationsaktivitäten sowie deren Abhängigkeiten zum großen Teil noch nicht bestimmt bzw. verstanden. Im folgenden werden einige wichtige Aspekte bei der Definition und Durchführung von Kompensationen kurz erläutert.

Wie oben beschrieben werden kompensierende Aktivitäten bzw. kompensierende Transaktionen benötigt, um reale Aktionen semantisch rückgängig zu machen. Dies sind zum einen Transaktionen bzw. Subtransaktionen, welche schon ein Commit durchgeführt haben, als auch solche Aktivitäten die andere Transaktionen berühren (beeinflussen), so daß durch Kompensation kaskadierende Aborts (Abbrüche) der Transaktionen vermieden werden können [KLS90].

Die unterschiedlichen Arten der jeweils notwendigen Kompensationsvorgänge reichen dabei vom herkömmlichen UNDO einer Transaktion, bis hin zu anwendungsabhängigen oder spezialisierten kompensierenden Transaktionen. In jedem Fall handelt es sich um das Zurücknehmen von Effekten, welche anderen Transaktionen schon sichtbar waren, ohne die Wirkungen dieser Transaktionen damit zu beeinflussen und gleichzeitig die Konsistenz der Daten und der Abläufe wiederherzustellen. Eine Kompensation geschieht auf eine semantische bzw. logische Art und Weise, und es werden dementsprechend auch semantische Informationen aus der Anwendung dazu benötigt. Der durch die Durchführung von Kompensationen erzeugte konsistente Zustand muß nicht grundsätzlich identisch mit einem der Zustände sein, die ohne das Ausführen von kompensierenden Aktionen erreicht worden wären bzw. erreicht worden sind. Welche Zustände nach einer Kompensation als gültig akzeptiert werden, ist anwendungsabhängig.

Kompensierende Transaktionen besitzen außer den grundsätzlichen Eigenschaften von Transaktionen noch einige spezielle Charakteristiken [KLS90], die im folgenden kurz beschrieben werden. Die kompensierende Transaktionen sind atomar. Zwischenergebnisse sollten nebenläufigen Transaktionen auf keinen Fall sichtbar sein. Zudem sind sie auf eine besondere Art und Weise dauerhaft, denn, wurde eine kompensierende Transaktion gestartet, so soll sie unter allen Umstän-

den durchgeführt werden, da es nicht sinnvoll ist sie abubrechen oder zu kompensieren. Jede kompensierende Transaktion verhält sich gemäß der Konsistenzbedingungen. In einigen Fällen wird die Konsistenz wiederhergestellt, statt sie zu bewahren. Eine kompensierende Transaktion existiert nicht für sich alleine, sondern immer in Zusammenhang mit einer durch sie kompensierbaren Transaktion. Die kompensierende Transaktion wird grundsätzlich nach der zu kompensierenden Transaktion durchgeführt. Ihre Aktionen sind abgeleitet aus den Aktionen der zu kompensierenden Transaktion. In einigen Fällen ist es möglich diese Aktionen automatisch aus dem Programm der zu kompensierenden Transaktion, dem aktuellen Zustand der Datenbank sowie dem aktuellen Zustand des Log abzuleiten. Ansonsten sind die kompensierenden Transaktionen durch den Anwendungsprogrammierer vorzudefinieren. Ein Ziel ist dem Anwendungsprogrammierer in diesem Fall eine größtmögliche Systemunterstützung zu geben.

### **Bearbeitungsstand nach der Ausführung einer (partiellen) Kompensation**

Der Bearbeitungsstand nach der Ausführung einer Kompensation kann ein zuvor erreichter Bearbeitungsstand sein, er kann aber auch ein beliebiger anderer sein. Unter Umständen ist der durch eine Kompensation erreichte Bearbeitungsstand, im kompensationsfreien Prozeßmodell noch gar nicht enthalten. Die Modellierung oder die Ausführung von Kompensationen kann daher auch mit dem Einfügen von neuen Bearbeitungszuständen und mit dem Einfügen von neuen Aktivitäten, die aus diesen Bearbeitungsständen in existierende führen verbunden sein. Eine Kompensationsaktivität kann häufig nicht den exakten Zustand wiederherstellen, der vor der Durchführung der Aktivität bestanden hat. Hier ist es erforderlich die akzeptierbaren Abweichungen oder Zustände definieren zu können. Bei einer partiellen Kompensation ist gleichzeitig auch zu klären an welcher Stelle im Vorgang daraufhin die Bearbeitung fortgesetzt werden sollte.

### **Sicherstellung der Kompensierbarkeit**

Es ist notwendig einen Zustand der Daten während der weiteren Bearbeitung des Vorgangs sicherzustellen, der Voraussetzung für die Durchführung der Kompensation einer Aktivität ist, solange die Aktivität gegebenenfalls kompensiert werden muß. Wird in einer Aktivität beispielsweise der Kontostand eines Kontos erhöht, so darf das Konto solange nicht gelöscht werden, bis sichergestellt ist, das die Aktivität auf keinen Fall mehr zurückgenommen werden kann bzw. soll. Dies ist beispielsweise dann der Fall, wenn der gesamte Vorgang abgeschlossen wurde. Es ist vorgesehen zur Sicherstellung dieser Zustände Invarianten zu nutzen.

### **Randbedingungen von Kompensationen**

Für Kompensationsaktivitäten gelten in der Regel eine Reihe von Randbedingungen. Falls diese nicht erfüllt sind, darf eine Kompensation nicht ausgeführt werden. Beispielsweise können dies technische, administrative, verhaltensbezogene, organisatorische oder informationale Randbedin-

gungen sein. Ein Beispiel für eine organisatorische Randbedingung ist es, die Ausführung der Kompensation lediglich einer für die Ausführung der zu kompensierenden Aktivität übergeordneten Rolle zu erlauben (siehe [MD97]).

### **Fixpunkte**

Es ist nicht in allen Fällen möglich, für jede Aktivität eine Kompensation zu definieren, die die bisherigen Effekte wieder (semantisch) zurücknimmt (siehe auch [BEK93]). Außerdem muß berücksichtigt werden, daß es häufig nicht erwünscht oder sogar nicht erlaubt ist, spezielle schon abgeschlossene Aktivitäten innerhalb eines Workflows rückgängig zu machen. Beispiele sind Genehmigungen, nachdem sie unterschrieben und rechtskräftig geworden sind, oder Gebäude, die schon errichtet wurden. Deshalb muß es möglich sein, sogenannte Fixpunkte zu definieren, die die Kompensation in Form der semantischen Rücknahme der Effekte der Aktivitäten, die vor dem Fixpunkt ausgeführt wurden, verhindern bzw. verbieten. Als Beispiel kann hier die HUBau aufgeführt werden. Diese ist im Falle ihres Abbruchs (Stornierung) lediglich zu stoppen, d.h. alle aktiven Aktivitäten werden abgebrochen oder beendet. Die HUBau wird als gestoppt markiert, und die erstellten Unterlagen werden archiviert. Im Falle einer Wiederaufnahme der Baumaßnahme wird eine HUBau mit einer neuen Identifikationsnummer gestartet, d.h. es wird ein neuer Workflow initiiert.

## **7.6 Diskussion und Bewertung**

Zur Sicherstellung der Zuverlässigkeit eines Systems zur Vorgangsbearbeitung sind die transaktionalen Aspekte sowie weitere Aspekte bezüglich der Synchronisation und der Fehler- und Ausnahmebehandlung in die Modellierung der Vorgänge zu integrieren. Die Ausführungsmaschine des Workflow-Management-Systems ist so zu erweitern, daß die integrierten Aspekte interpretiert und unterstützt werden können. Und schließlich müssen auch alle anderen Komponenten des Workflow-Management-Systems entsprechend erweitert werden. Der Arbeitsbereich CAM befaßte sich mit der Anforderungsanalyse und der Definition eines Basismodells zur Sicherstellung der Konsistenz von langandauernden Vorgängen zur zuverlässigen Vorgangsbearbeitung in weitverteilten Anwendungsumgebungen. Aus heutiger Sicht ist das in CAM definierte und ausgewählte Basismodell zur Sicherstellung der Konsistenz von langandauernden Vorgängen grundsätzlich sehr gut geeignet. Dies zeigt besonders die Entwicklung von ConTracts [RSS97] und seine neueste prototypische Implementierung APRICOTS [Schn98] als transaktionales Workflow-Management-System. Um das Basismodell jedoch in der Anwendungsumgebung (PoliFlow) erfolgreich einzusetzen, ist seine Integration in das jeweils ausgewählte Workflow-Management-System bzw. sind Anpassungen des Workflow-Management-Systems erforderlich. Mechanismen

zur Konsistenzsicherung können nicht lediglich durch eine zusätzliche Komponente realisiert werden, sondern erfordern Änderungen und Erweiterungen innerhalb des Workflow-Management-Systems.

Grundsätzlich ist ohne die Integration von Mechanismen zur Konsistenzsicherung eine zuverlässige Vorgangsbearbeitung nicht garantiert. Der Einsatz eines Workflow-Management-Systems für die unternehmens- bzw. verwaltungskritischen Prozesse ist nur dann sinnvoll, wenn eine ausreichende Konsistenzsicherung gewährleistet ist.

## 7.7 Literatur

- [Anw88] Anwendung des Aktenplans, 1988.
- [BEK93] Omran Bukhres, Ahmed Elmagarmid und Eva Kuhn. Implementation of the Flex Transaction Model. In *Bulletin of the Technical Committee on Data Engineering*, volume 16, pages 28–32. IEEE Computer Society Press, June 1993.
- [CKO92] Bill Curtis, Marc I. Kellner und Jim Over. Process Modeling. *Communications of the ACM*, 35(9):75–90, September 1992.
- [DHL90] Umeshwar Dayal, Meichun Hsu, and Rivka Ladin. Organizing Long-Running Activities with Triggers and Transactions. In *ACM SIGMOD*, 1990.
- [Elm91] Ahmed K. Elmagarmid. *Database Transaction Models for Advanced Applications*. Morgan Kaufmann Publishers, San Mateo, California, 1991.
- [Ges75] Geschäftsordnung für die Oberfinanzdirektion, November 1975.
- [GHM<sup>+</sup>93] D. Georgakopoulos, M. F. Hornick, F. Manola, M. L. Brodie, S. Heiler, F. Nayeri, and B. Hurwitz. An Extended Transaction Environment for Workflows in Distributed Object Computing. In *Bulletin of the Technical Committee on Data Engineering*, volume 16, pages 24–27. IEEE Computer Society Press, June 1993.
- [GHS95] Dimitrios Georgakopoulos, Marek Hornick, and Amit Shet. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, April 1995.
- [GMS87] Hector Garcia-Molina and Kenneth Salem. Sagas. pages 249–258. Association of Computing Machinery, 1987.
- [GR93] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [HFK90] Abraham Silberschatz Henry F. Korth und Eliezer Levy. A formal approach to recovery by compensating transactions. In *Proceedings of the [16th]VLDB Conference*, pages 95–106, 1990.
- [Jab95] Stefan Jablonski. *Workflow-Management-Systeme, Modellierung und Architektur*. An International Thomson Publishing Company, 1995.

- [JG93] Andreas Reuter und Jim Gray. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [JK97] Sushil Jajodia und Larry Kerschberg. *Advanced Transaction Models and Architectures*. Kluwer Academic Publishers, 1997.
- [JNRS93] W. Woody Jin, Linda Ness, Marek Rusinkiewicz, and Amit Sheth. Concurrency Control and Recovery of Multidatabase Work Flows in Telecommunication Application. In *ACM SIGMOD*, pages 456–459, Washington D.C., 1993. Association of Computing Machinery.
- [KGH93] D. Georgakopoulos, M. F. Hornick, and Piotr Krychniak. An Environment for Specification and Management of Extended Transactions in DOMS. In *Proceedings: RIDE-IMS*, pages 253–257. IEEE Computer Society Press, 1993.
- [KLS90] Henry F. Korth, Eliezer Levy, and Abraham Silberschatz. A Formal Approach to Recovery by Compensating Transactions. In *Proceedings of the 16<sup>th</sup> VLDB Conference*, pages 95–106, Brisbane, Australia, 1990.
- [KS94] Henry F. Korth and Greg Speegle. Formal Aspects of Concurrency Control in Long-duration Transaction Systems Using the NT/PV Model. In *ACM Transactions on Database Systems*, volume 19, pages 492–535, September 1994.
- [LEB90] Y. Leu, A. Elmagarmid, and N. Boudriga. Specification and Execution of Transaction for Advanced Database Applications. Technical report, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, 1990. CSD-TR 1030
- [Ley95] Frank Leymann. Supporting Business Transactions Via Partial Backward Recovery In Workflow Management Systems. In Georg Lausen (Hrsg.), *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), GI-Fachtagung, Dresden, 22.-24. März 1995*, Informatik Aktuell, S 51-70. Springer, 1995
- [MD97] Marcus Danner. Entwicklung eines graphischen Werkzeugs zur Modellierung von Kompensation in Workflow-Management-Systemen. Diplomarbeit Nr. 1426, Universität Stuttgart, 1997.
- [Rei94] Dr. Berthold Reinwald. Workflow-Management, Tutorium 01. 12. 1994 in Mainz. Technical Report, Deutsche Informatik Akademie, Gesellschaft mit beschränkter Haftung, Wissenschaftszentrum, Ahrstraße 45, 53175 Bonn, 1994.
- [RELL90] Marek Rusinkiewicz, A.K. Elmagarmid, Y. Leu, and W. Litwin. Extending the Transaction Model to Capture More Meaning. *SIGMOD RECORD*, 19(1):3–7, March 1990.
- [RS95] Andreas Reuter und Friedemann Schwenkreis. Contracts - A Low-Level Mechanism for Building General-Purpose Workflow Management-Systems. In *Bulletin of the Technical Committee on Data Engineering*, volume 18, pages 4–10. IEEE Computer Society Press, March 1995.
- [RSS97] Andreas Reuter, Kerstin Schneider und Friedemann Schwenkreis. ConTracts Revisited. Chapter 5 in [JK 97].
- [RW90] Andreas Reuter und Helmuth Wächter. Grundkonzepte und Realisierungsstrategien des Contract-Modells. In *Informatik Forschung und Entwicklung*, volume 5, pages 202-212. Springer Verlag, 1990.

- [RW93] Andreas Reuter and Helmuth Wächter. ConTracts. Chapter 7 in [Elm91].
- [Schw94] Friedemann Schwenkreis. A Formal Approach to Synchronize Long-Lived Computations. In *Proceedings of the 5<sup>th</sup> Australasian Conference on Information Systems*, volume 1, pages 377–387, Department of Information Systems, Monash University, September 1994.
- [Schw95] Friedemann Schwenkreis. Extended Abstract of Work in Progress: Apricots - A Workflow Programming Environment, 1995.
- [Schn98] Kerstin Schneider. *APRICOTS - ein verteiltes, transaktionales Workflow-System auf der Basis von CORBA*, Extended Abstract, in: Datenbank Rundbrief der Gesellschaft für Informatik, Mai 98
- [Unl94] Rainer Unland. Topaz, A Tool Kit for the Assembly of Transaction Managers for Non-Standard Applications. Working paper no.34, University of Münster, Institute of Business Informatics, Greverer Str 91, D-48159 Münster, Germany, November 1994.
- [ZNBB94] Aidong Zhang, Marian Nodine, Bharat Bhargava, and Omran Bukhres. Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems. In *ACM SIGMOD*, pages 67–78. Association of Computing Machinery, 1994.

## 8 Technologie Evaluierung (TE)

In PoliFlow ist ein weit verteiltes Anwendungsszenario gegeben, aufgrund dessen wurden Technologien untersucht, die helfen die räumliche Trennung der Arbeitsplätze zu überbrücken und die Vorgangsbearbeitung effizienter zu gestalten.

Zu den Technologien, die das weit verteilte Anwendungsszenario möglichst optimal unterstützen, zählen Systemdienste zur Unterstützung von verteilten Systemen, Multimedia-Technologien zur Vermeidung von Medienbrüchen und Telekooperationstechniken zur Gestaltung effektiver Gruppenarbeit.

In PoliFlow gibt es zwei Arbeitspakete, die sich mit der Evaluierung dieser Technologien beschäftigen. In dem folgenden Abschnitt wird ein Überblick über Technologien gegeben die verteilte Systemdienste zur Verfügung stellen. Im Abschnitt 8.2 werden Multimedia Technologien behandelt, die die Kommunikation und Kooperation im weit verteilten Anwendungsfeld verbessern könnten.

### 8.1 Evaluierung von verteilten Systemdiensten

Die Notwendigkeit Systemdienste einzusetzen, die das Arbeiten in weit verteilten, heterogenen Systemen unterstützen ergibt sich aus der räumlichen Trennung der Arbeitsplätze im Anwendungsumfeld von PoliFlow. Zur Klärung welche Systemdienste erforderlich sind, um eine effiziente und integrierte Vorgangssteuerung in einem verteilten System zu realisieren, werden zuerst die wichtigsten Anforderungen identifiziert, die von der Vorgangssteuerung an eine verteilte Systemumgebung gestellt werden. Der nächste Abschnitt beschreibt die Anforderungen und anschließend werden die Ergebnisse der Evaluierung von OSF DCE, OMG OMA vorgestellt und bewertet.

#### 8.1.1 Anforderungen

Bei der Analyse der Anforderungen, wurde festgestellt, daß zwei verschiedene Klassen von Anforderungen existieren. Dies sind zum einen die Anforderungen, die entstehen wenn Benutzer an ein Vorgangssteuerungssystem angebunden werden und zum anderen Anforderungen, die von einer verteilt realisierten Vorgangssteuerung an die Systemplattform gestellt werden.

*Anforderungen für die Benutzeranbindung:*

## 1. Offene Schnittstellen / Offenes Systeme

Die Integrationsfähigkeit von funktionalen Komponenten und Werkzeugen ist ein sehr wichtiger Aspekt für die Realisierung von Vorgangssteuerungssystemen. Auf der Basis von offenen Schnittstellen in einem offenen System ist eine nahtlose Integration der verschiedenen Komponenten erst möglich.

## 2. Standards / Interoperabilität

Neben der Integrationsfähigkeit zählt auch die Interoperabilität unterschiedlicher Komponenten zu einer der Voraussetzungen, damit in einer verteilten Systemumgebung ein Vorgangssteuerungssystem realisiert werden kann. Zur Gewährleistung der Interoperabilität ist die Verwendung von Standards unabdingbar.

## 3. Verfügbarkeit von Kommunikationskanälen

Die Vorgangssteuerung enthält alle Zustände, Daten und sonstige Informationen über die Vorgänge. Wenn ein Benutzer seine Aufgaben, die ihm von der Vorgangssteuerung zugewiesen wurden, abarbeiten soll, muß der Benutzer diese Informationen über Kommunikationskanäle von der Vorgangsteuerung erhalten. Auch für die Rückmeldungen der Benutzer zu der Vorgangssteuerung sind Kommunikationskanäle erforderlich.

## 4. Authentifikation von Kommunikationspartnern

Da im Anwendungsszenario von weit verteilten Systemen ausgegangen wird, muß sichergestellt sein, daß Benutzer keine falsche Identität einnehmen können.

## 5. Schutzmechanismen für die Kommunikationskanäle

Das Anwendungsszenario besteht aus weit verteilten Systemen und daher muß sichergestellt sein, daß weder Kommunikationskanäle abhört noch manipuliert werden können.

## 6. Authorisierung für den Dienstzugriff/Werkzeugzugriff

Es muß sichergestellt sein, daß nur berechtigte Benutzer und Systemkomponenten Dienste und Werkzeuge nutzen können.

## 7. Ortstransparenter Zugriff auf Daten.

Daten, die von den Benutzern bearbeitet werden sollen, müssen entweder mit der Aktivität übergeben werden oder müssen über eine ortstransparente Referenz erreichbar sein. Die Migration/Replikation von Daten kann zur Steigerung der Performanz und Verfügbarkeit eingesetzt werden.



*Anforderungen einer verteilten Vorgangssteuerung:*

## 1. Zuverlässige Kommunikationskanäle

Die Kommunikationskanäle die zwischen den WF-Steuerungskomponenten notwendig sind haben besonders hohe Anforderungen bezüglich Sicherheit und Zuverlässigkeit. Es dürfen keine Informationen über Zustände, Daten oder Vorgänge verloren gehen (vgl. IPVR.CAM)

## 2. Stabiler Sekundärspeicher

Die WF-Steuerung hat hohe Anforderung an die Integrität der Zustände und Daten der Vorgänge. Daher ist eine stabile und zuverlässige Speicherung auf Sekundärspeicher notwendig.

## 3. Managementunterstützung

Eine einheitliche Überwachung und Kontrolle des verteilten WF-Steuerungssystems ist notwendig. Standardisierte Mechanismen der Systemplattform vereinfachen das Systemmanagements.

## 5. Offene Schnittstellen / Offene Systeme / Standards / Interoperabilität

Genormte Protokolle und Standards unterstützen die angestrebte Interoperabilität verschiedener Vorgangssteuerungssysteme im heterogenen Anwendungsumfeld.

Diese beiden Klassen der Anforderungen bilden die Grundlage für die Technologie Evaluierung. Es fließen allerdings neben diesen Anforderungen auch praktische Aspekte, wie die Verfügbarkeit der Systemplattformen, in die Bewertung mit ein.

## 8.1.2 Technologie Evaluierung

Es wurden das *Distributed Computing Environment* der OSF und die *Objekt Management Architecture* der OMG untersucht. Das OSF DCE ist seit einigen Jahren kommerziell verfügbar und wurde als Vertreter der klassischen Client/Server Architektur untersucht. Die Entwicklung rund um die OMG OMA ist neuer und zur Zeit noch auf dem Weg der Standardisierung und wird als Vertreter des objektorientierten Ansatzes untersucht.

### OSF DCE

Das *Open Software Foundation's Distributed Computing Environment* (OSF DCE) stellt eine Infrastruktur zur Verfügung mit dessen Hilfe Anwendungen effizienter erstellt werden können, die auf dem Client/Server Paradigma basiert sind. Die Basis von DCE sind POSIX-Threads, zur Realisierung multitaskingfähiger Applikationen, und die Kommunikation mittels Remote Procedure Call's (RPC's), zur Abstraktion der u.U. heterogenen Netzwerk- und Rechnerumgebung.

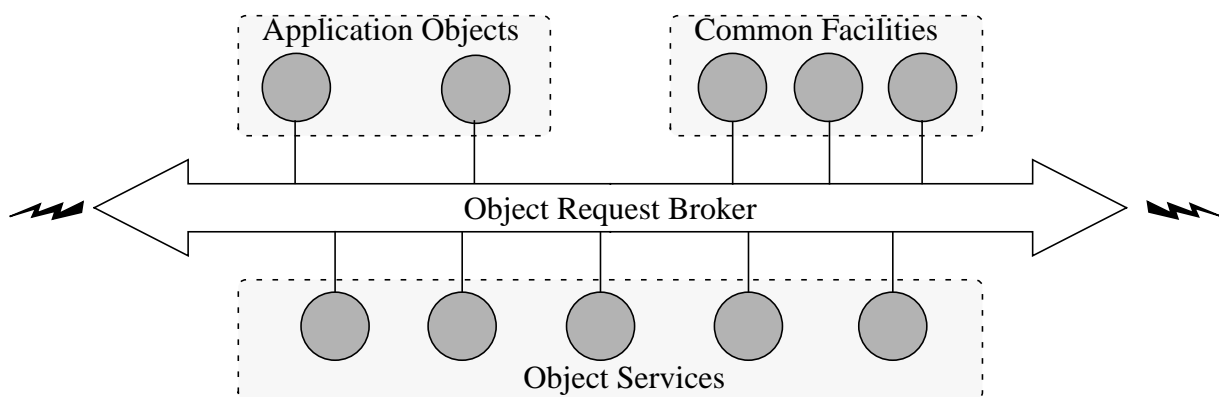
Als serverbasierte Dienste steht ein Verzeichnis-, Zeit- und Sicherheitsdienst zur Verfügung. Die Schnittstellen zu diesen Diensten sind standardisiert und bieten auf unterschiedlichen Plattformen einen einheitliche Entwicklungsumgebung an. Neben diesen Basisdiensten können noch weitere Dienste eingesetzt werden, zu ihnen gehören z.B. ein verteiltes Dateisystems (Distributed File System, DFS) oder Transaktionsmonitore (Encina).

Die Anforderungen der Benutzersicht können durch DCE abgedeckt werden, besonders die Kommunikation und Sicherheitsaspekte. Auch die Anforderungen eines verteilten Vorgangssteuerungssystems können durch DCE erfüllt werden, allerdings müssen dazu zusätzliche Dienste, wie transaktionale RPC's (TRPC), verwendet werden.

Eine Schwachstelle von DCE ist die unzureichende Abbildung der Schnittstellendefinitionen auf Programmiersprachen. DCE bietet im Standard nur die Abbildung auf die Programmiersprache C. Dies bereitet Probleme bei der Integration von neuen Anwendungen und Altsoftware. Bei neuen Anwendungen fehlt die Möglichkeit moderne Programmiersprachen wie Java einzubinden. Im Allgemeinen ist bei DCE die Einbindung objektorientierter Software nur unzureichend möglich.

### OMG OMA

Das Ziel der *Object Management Group* (OMG) ist die Entwicklung von standardisierten Schnittstellen für interoperable objektorientierte Software Komponenten. Dies soll durch die *Object Management Architecture* (OMA) realisiert werden. In dieser Architektur kommunizieren Anwendungsobjekte über eine Infrastrukturkomponente, die *Object Request Broker* (ORB) genannt wird, mit Objekten die Basisdienste oder allgemeine Dienste zur Verfügung stellen. Der ORB ist eine ortstransparente, verteilte Objektplattform und der *Common Object Request Broker Architecture* (CORBA) Standard spezifiziert die standardisierten Schnittstellen für den ORB.



**Abbildung 8.1:** Object Management Architecture (OMA)

**Tabelle 5.1:**

Zu den Object Services (OS) zählen der Name-, Lifecycle-, Event-, Persistence-, Interoperability und weitere Dienste, sie stellen den Anwendungsentwicklern eine standardisierte allgemeine Entwicklungsumgebung zur Verfügung.

Bei den allgemeinen Diensten (*Common Facilities*) handelt es sich, um spezialisiertere Dienste wie Backup, Mail oder Workflow, die benötigte Standardkomponenten für Anwendungsapplikationen realisieren. Bisher sind allerdings nur wenige CF-Dienste spezifiziert worden.

	OSF DCE	OMG OMA
Modell	Prozedural, Client/Server	Objektorientiert,
Kontroll- und Datenfluß	Remote Procedure Call (RPC)	Remote Method Invocation (RMI)
Schnittstellendefinition	IDL (C-ähnlich)	IDL (C++-ähnlich)
Sprachabbildung	C	C, C++, Java, Ada, Smalltalk, ...
Basisdienste	Threads, RPC, Directory, Security, Time	Object Request Broker (ORB)
Standard Dienste	Distributed File System (DFS), Diskless Service, PC Integration	Name, Lifecycle, Event, Persistence, Interoperability, Concurrency, Transaction, Externalisation, Relationship, Security, Time, Licensing, Properties, Query
erweiterte Dienste	Transaktionsmonitore (Encina), ...	Backup, Operational Control, Mail, Printing, Workflow
Internetfähig	Aufwendig	Ja (Java)
Verfügbarkeit	Hoch (viele Hersteller, viele Plattformen)	Hoch. (viele Hersteller, viele Plattformen, viele Programmiersprachen)
<b>Tabelle 8.1:</b> Vergleich von OSF DCE und OMG OMA		

Das OMA-Modell und die daraus abgeleiteten Spezifikationen erfüllen in weiten Teilen die Anforderungen, die sich aus dem PoliFlow Umfeld ergeben. Besonders die Aspekte der Kommunikation und der Interoperabilität zwischen heterogenen Rechner- und Netzwerksystemen wird durch OMA gut unterstützt. Durch das objektorientierte Modell wird überdies die Integration und

Wiederverwendung von funktionalen Bausteinen bzw. Komponenten erleichtert. Ein Schwachpunkt sind sicherlich zur Zeit noch einige Lücken bei der Spezifikation der allgemeinen Dienste. Interoperable ORB-Implementierungen sind bereits kommerziell auf dem Markt verfügbar.

Gegenüber dem OSF DCE hat OMG OMA den Vorteil, daß ein komponentenbasiertes Design unterstützt wird, dadurch können die funktionalen Bausteine wiederverwendet werden oder durch bessere Komponenten ersetzt werden. Gerade diese komponentenorientierte Architektur ist für die Realisierung eines Vorgangssteuerungssystems besonders gut geeignet, denn es kann je nach Systemplattform, Anforderungen und Bedarf die Vorgangssteuerung und die Benutzeranbindung optimal an die Gegebenheiten im Anwendungsszenario angepaßt werden. Weiterhin ist OMG OMA ideal geeignet Altsoftware zu integrieren, da einerseits im Standard bereits viele Abbildungen in Programmiersprachen definiert sind und andererseits Abbildungen in beliebige Programmiersprachen konzeptionell vorgesehen ist.

### 8.1.3 Bewertung

DCE ist nur beschränkt geeignet, da es aufgrund der Fixierung auf die Programmiersprache C zu Problemen bei der Integration von Anwendungen und Altsoftware kommen kann.

CORBA ist für die Realisierung gut geeignet. CORBA ist mittlerweile ein ausgereifter Standard und durch die Unabhängigkeit sowohl von der Betriebssystemplattform wie auch der Programmiersprache ist die einfache Integration von Anwendungen und Altsoftware gegeben.

Als wesentlicher Schwachpunkt bei der Evaluierung von Middleware-Systemen wurde das Fehlen von geeigneten Sicherheitsmechanismen zur Realisierung einer sicheren Vorgangssteuerung in weit verteilten Systemen identifiziert. Zusätzliche Anforderungen die Sicherheit werden vor allem durch dynamische Autorisierung, Rollenkonzepte und Stellvertreterregelungen, die in Vorgangssteuerungssystemen unabdingbar sind, gestellt.

Die Assoziation einer Person mit Aktivitäten, ergibt eine Dynamik, die mit bisherigen Sicherheitssystemen nicht effizient bewältigt werden kann. Die in IPVR.TE.1 untersuchten Technologien<sup>1</sup> setzen z.B. in der Regel ACL-basierte Mechanismen zur Zugriffskontrolle auf Dienste und Daten ein. Die entsprechenden Einträge in den ACL's werden entweder an Personen, Gruppen oder Diensten festgemacht, deswegen kann nur eine statische Modellierung der Zugriffskontrolle erfolgen. Dies steht im Widerspruch zu der Flexibilität von Vorgangssteuerungssystemen.

<sup>1</sup>Diese Aussage trifft vor allem auf das OSF DCE zu. Für die Technologien die auf der OMG OMA basieren können z.Z. noch keine definitiven Aussage gemacht werden, da die Spezifikation des Sicherheitsdienstes noch nicht abgeschlossen ist. Es ist allerdings abzusehen, daß im Rahmen von OMG OMA ein flexibleres Framework für die Realisierung von Sicherheitsdiensten spezifiziert wird.

Neben der dynamischen Zuordnung von Rollen zu Personen ergeben sich auch aus der Möglichkeit zur Bestimmung von Stellvertretern weitere Anforderungen an das Zugriffskontrollsystem. Es muß dem Stellvertreter ermöglicht werden alle Dienste und Daten in Anspruch zu nehmen, die zur erfolgreichen Bearbeitung notwendig sind, aber es muß auch gewährleistet sein, daß eine stellvertretende Person darüberhinausgehende Privilegien der zu vertretenden Person nicht in Anspruch nehmen kann.

## 8.2 Multimedia-Technologien

Das Schlagwort „Multimedia“ umfaßt sehr viele Technologien mit den unterschiedlichsten Einsatzgebieten. Es wird heutzutage überall dort verwendet, wenn Bilder, Sprache oder Video eingesetzt werden. Es ist dabei nicht relevant, wie die verschiedenen Medien eingesetzt werden, d.h. es ist bereits „Multimedia“, wenn eine Benutzeroberfläche mit animierten Icons verschönert wird, andererseits zählen auch so aufwendige Applikationen wie Desktop-Konferenzsysteme zu den Multimedia-Technologien. Daher können Multimedia-Technologien nicht im allgemeinen bewertet werden, sondern es ist nur sinnvoll gleiche Klassen von Multimedia-Anwendungen zu evaluieren.

In diesem Kapitel wird eine Klassifikation eingeführt mit der sicherlich nicht alle Multimedia-Technologien eingeordnet werden können. Die Klassifikation enthält hauptsächlich die Technologieklassen, die für die Integration von Multimedia und Workflow bedeutend sind. Nach der Klassifikation wird auf die einzelnen Technologien genauer eingegangen.

### 8.2.1 Klassifikation

Bei der Integration von Multimedia und Workflow soll der Informationsaustausch, der bei den verschiedenen Geschäftsvorgängen anfällt, vereinfacht bzw. natürlicher gestaltet werden. Es können zwei verschiedenen Arten des Informationsaustausches unterschieden werden. Es gibt den bei Workflow Management Systemen üblichen asynchronen Informationsaustausch und den durch Konferenzsysteme erst möglichen synchronen multimedialen Informationsaustausch.

#### **Asynchroner Informationsaustausch**

Der asynchrone Informationsaustausch ist bei Workflow-Managementsystemen üblich, er umfaßt im wesentlichen die Weiterleitung von Dokumenten, die zur Bearbeitung der Vorgänge notwendig sind. Die Einführung von „Multimedia“ in diesem Bereich bedeutet vor allem die Möglichkeit zur Verwendung von Multimedia-Dokumenten. Damit Multimedia-Dokumente verwendet werden können müssen gewisse Voraussetzungen erfüllt werden. Dazu gehören unter anderem: Arbeitsplatzrechner die sowohl Graphik-, Video- und Audiodaten darstellen können, entsprechende

Werkzeuge zur Erzeugung, Bearbeitung und Wiedergabe von Multimedia-Dokumenten und neben der getrennten Behandlung der verschiedenen Medientypen sogenannte Verbunddokumente, die eine typenübergreifende Verarbeitung und Präsentation möglich machen.

### **Synchroner Informationsaustausch**

Zu den Multimedia-Technologien, die dem synchronen Informationsaustausch zugeordnet werden, gehören Desktop Konferenzsysteme. Dabei geht es vor allem um Kommunikation, und Kooperation zwischen Menschen, dazu zählt sowohl direkte Audio- und Videokommunikation als auch das gemeinsame Bearbeiten von Dokumenten.

### **Klassifikationskriterien**

Multimedia-Technologien, die dem Informationsaustausch dienen, können in zwei Klassen eingeteilt werden. Der wesentliche Unterschied zwischen den beiden Klassen ist der zeitliche Zusammenhang in dem die Informationen verarbeitet werden.

Somit gibt es die Klasse der asynchronen Multimedia-Technologien und der synchronen Multimedia Technologien. Im folgenden Abschnitt werden die asynchronen Technologien beschrieben, im Abschnitt 8.2.3 wird auf die synchronen Technologien eingegangen.

## **8.2.2 Asynchrone Multimedia Technologien**

Der Begriff Multimedia Dokument faßt alle Arten der Speicherung und Darstellung von Informationen und Informationstypen zusammen. Es sind vor allem die relativ neuen Medientypen Bild, Audio und Video interessant, die Typen Text und Vektorgraphik (CAD u.ä.) zählen eher zu den Standard-Medientypen. In diesem Bericht wird vor allem auf die neuen Medientypen (Bild, Audio, Video) und ihre speziellen Anforderungen an die Systeminfrastruktur eingegangen.

### **Medientyp: Bild**

Dieser Medientyp umfaßt alle Arten von nichtstrukturierten Bildinformationen, d.h. vor allem keine Vektorgraphiken und CAD-Daten. Bilder werden meistens nur als Informationsgrundlage für menschliche Betrachter verwendet. Es existieren sehr unterschiedliche Bildformate. Es ist aber nicht abzusehen, daß sich ein Standard für die Darstellung und Speicherung von Bildern durchsetzen wird. Es ist vielmehr davon auszugehen, daß die Werkzeuge zur Erstellung und Darstellung verschiedene Bilddatenformate verarbeiten und konvertieren können. Dadurch wird es möglich sein eine maximale Interoperabilität zwischen Bildproduzenten und Konsumenten zu erreichen.

### Medientyp: Audio

Der Medientyp Audio umfaßt alle digitalen Darstellungen von Ton. Abhängig von der Qualität kann es zu einem großen Unterschied bei dem Ressourcenbedarf kommen. Die Qualität kann durch drei Parameter angegeben werden: Kanalanzahl, Samplingrate und Sampleauflösung. Aus diesen Parameter kann dann die Bandbreite die zur isochronen Audioübertragung benötigt wird ermittelt werden.

- Kanalanzahl:

Ton kann mit mehreren Kanälen aufgezeichnet werden. Entweder um Stereo bzw. Raumklang zu erhalten, oder um den gleichen Inhalt in mehreren Sprachen zur Verfügung zu stellen.

- Samplingrate:

Die Aufnahme­frequenz gibt an wieoft der Ton gesampelt wird. Desto höher die Frequenz um so höher ist die Qualität. Audio-CDs werden z.B. mit 44100 Hz digitalisiert. Beim digitalen Telefonieren wird normalerweise mit 8000 Hz digitalisiert.

- Sampleauflösung:

Die Sampleauflösung gibt an mit welcher Genauigkeit der Ton digitalisiert wird. Der Ton auf Audio-CDs wird mit 16 Bit Auflösung digitalisiert. Für normalen Telefonverkehr wird meist 8 Bit verwendet.

- Bandbreite:

Mit diesen drei Parametern kann nun die Bandbreite, die eine Audioaufzeichnung benötigt, berechnet werden. Dazu kann folgende Formel verwendet werden:

$$Bandbreite_{Audio} \left[ \frac{KByte}{s} \right] = \frac{Kanalanzahl \times Rate \left[ \frac{Sample}{s} \right] \times Auflösung \left[ \frac{Bit}{Sample} \right]}{8 \left[ \frac{Bit}{Byte} \right] \times 1024 \left[ \frac{Byte}{KByte} \right]}$$

Aus der Bandbreite und der Dauer der Audiosequenz kann der Speicherbedarf berechnet werden:

$$Speicherbedarf_{Audio} [KByte] = Bandbreite_{Audio} \left[ \frac{KByte}{s} \right] \times Dauer_{Audio} [s]$$

### Medientyp: Video

Mit Video werden alle Bewegtbild-Sequenzen zusammengefaßt. Die Qualität von Video kann durch vier Parameter angegeben werden: Bildrate, Auflösung, Farbtiefe und Kompression. Aus diesen Parametern kann die Bandbreite die zur isochronen Übertragung von Video notwendig ist berechnet werden. Über die Dauer der Videosequenz kann auch der Speicherplatz bestimmt werden, falls das Video gespeichert werden soll.

- **Bildrate:**

Die Bildrate bestimmt wie flüssig die Bewegungen in der Videosequenz dargestellt werden. Je höher die Rate ist desto eher entsteht der Eindruck eines Bewegtbildes. Für das menschliche Auge ist ab 25 [Bilder/s] dieser Eindruck gegeben. Für Videokonferenzen reichen allerdings auch geringere Bildraten, um den Eindruck von Bewegung zu übermitteln, so sind 10 -15[Bilder/s] durchaus ausreichend.

- **Auflösung:**

Die Auflösung legt die 2-D Dimensionen des Videos fest. Desto größer die Auflösung ist desto mehr Details kann das Video darstellen.

- **Farbtiefe:**

Die Farbtiefe legt fest wie genau die realen Farben abgebildet werden. Eine Farbtiefe von 8 Bit (256 Farben) ist heute auf den meisten Arbeitsplatzrechnern mindestens üblich, diese Farbtiefe reicht für Video allerdings nicht aus. Als True Color, also für das Auge Echtfarbdarstellung, wird die Farbtiefe von 24 Bit (~16 Mill. Farben) bezeichnet, für medizinische Anwendungen kann aber selbst diese Farbtiefe noch zu gering sein.

- **Kompression:**

Ein besonders wichtiger Parameter für die Aufzeichnung, Übertragung, Speicherung und Darstellung von Video ist die Kompression, da bei Video eine Unmenge von Informationen anfallen, die ohne Kompression nur unter extrem hohen Kosten weiterverarbeitet werden können. Bei der Videokompression kann zwischen prinzipiell zwei verschiedenen Arten von Kompression unterschieden werden: die Kompression innerhalb der Daten eines Bildes und der Kompression zwischen mehreren aufeinanderfolgenden Bildern. Bei der Kompression von Bildern ist es nicht notwendig verlustfrei zu komprimieren, da das menschliche Auge gewisse Verluste bzw. Änderungen nicht registriert. Es ist daher möglich Algorithmen zu verwenden, bei denen sich das Originalbild nicht verlustfrei komprimiert/dekomprimiert wird. Dadurch können ungleich höhere Kompressionsfaktoren erzielt werden, wie mit klassischen verlustfreien Kompressionsverfahren.

- **Bandbreite:**

Die Bandbreite die Video zur isochronen Übertragung benötigt kann wie folgt berechnet werden:

$$Bandbreite_{Video} \left[ \frac{KByte}{s} \right] = \frac{Auflosung \left[ \frac{Punkte}{Bild} \right] \times Farbtiefe \left[ \frac{Bit}{Punkt} \right] \times Rate \left[ \frac{Bilder}{s} \right]}{8 \left[ \frac{Bit}{Byte} \right] \times 1024 \left[ \frac{Byte}{KByte} \right]}$$

Die Kompression wurde bei dieser Berechnung nicht berücksichtigt, da der Kompressionsfaktor im allgemeinen nicht konstant ist, sondern je nach Bilderinhalt variiert.



- Speicherbedarf:

Der Speicherbedarf kann unter Berücksichtigung der Dauer des Videos berechnet werden:

$$\text{Speicherbedarf}_{\text{Video}}[\text{KByte}] = \text{Bandbreite}_{\text{Video}} \left[ \frac{\text{KByte}}{\text{s}} \right] \times \text{Dauer}_{\text{Video}}[\text{s}]$$

Auch beim Speicherbedarf ist die Kompression noch nicht mit eingerechnet.

### **Verbunddokumente**

Die Verwendung von Verbunddokumenten (*Compound Documents*) ist ein weiterer Schritt zur Integration und einfacheren Handhabung der verschiedenen Medientypen. Durch die Verwendung von Verbunddokumenten ist es für den Benutzer nicht mehr erforderlich selber zwischen den Werkzeugen/Anwendungen für die verschiedenen Medien zu wechseln, sondern alle Medien werden zusammen präsentiert und das Verbunddokumenten ist für die korrekte Verwendung der Erstellungs-, Bearbeitungs- und Darstellungswerkzeugen verantwortlich. Die Begriffe „Editing in place“ und „In place activation“ stehen für die Vorteil von Verbunddokumenten. Der Benutzer muß sich nicht mehr bewußt sein mit welchen Medien und Werkzeugen er arbeitet, somit ist die Verwendung von Verbunddokumenten ein wichtiger Schritt zum effektiven und effizienten Umgang mit Multimedia-Daten.

### **Anforderungen an Multimedia Dokumente**

Auf unterschiedlichen Ebenen sind Anforderungen an Multimedia-Dokumente zu identifizieren. Es gibt Anforderungen, die der Nutzer von Multimedia-Dokumenten an Multimedia-Dokumente stellt, andererseits stellt der Einsatz von Multimedia-Dokumenten Anforderungen an das Rechner-system und es gibt auch Anforderungen die aus der Integration von Multimedia-Dokumenten und Workflow Management Systemen ergibt.

### **Nutzeranforderungen**

Die Nutzer erwarten verschiedene Vorteile bei der Verwendung von Multimedia-Dokumenten:

- Logisch zusammengehörige Daten, auch bei unterschiedlichem Medientyp, sind zusammenhängend darzustellen.
- Medienbrüche sind zu vermeiden, d.h. alle Medien sollen in elektronisch gespeicherter Form als Daten vorliegen und alle gewünschten Operation müssen mit diesen Daten möglich sein.

### **Systemanforderungen**

An das System werden durch Multimedia-Dokumente neue Anforderungen gestellt. Vor allem die zeitabhängigen Medien (Video/Audio) und ihre Eigenschaften erfordern besondere Berücksichtigung:

- Für die Verarbeitung und Darstellung der Medien sind Dienstgütezusicherungen notwendig, damit die benötigten Ressourcen zur richtigen Zeit im richtigen Umfang verfügbar sind. Zu den Ressourcen gehören u.a.:
  - CPU Leistung: Die CPU muß in konstanten Intervallen Rechenleistung bereitstellen, um die anfallenden Daten der kontinuierlichen Medienströme zu verarbeiten.
  - Kommunikationsbandbreite: Besonders Video benötigt neben der kontinuierlichen Bereitstellung auch noch eine hohe Bandbreite, die meist nicht durch konventionelle Technik bereitgestellt werden kann.
  - Sekundärspeicherzugriff: Wenn Multimedia-Dokumente von Sekundärspeicher gelesen oder auf Sekundärspeicher geschrieben werden, müssen sowohl die Bandbreite wie auch die Verzögerung garantiert werden bzw. entsprechend vorreserviert sein.
- Die Synchronisation der einzelnen Medien untereinander muß durch das System ermöglicht werden, dabei sollten bestimmte maximal Werte nicht überschritten werden (vgl. Tabelle 8.2, "Dienstgüte für Medien," auf Seite 152).

### **Integrationsbedarf**

Bei der Integration von MM-Dokumenten in ein Workflow Management System ergeben sich einige Standardanforderungen, die immer auftreten wenn Multimedia verwendet werden soll und einige spezielle Workflow bezogene Anforderungen:

- Allgemeine Anforderungen:
  - Die Arbeitsplätze müssen multimedialfähig sein, dazu gehören vor allem die Fähigkeiten Audio abzuspielen und Video darzustellen. Falls Multimedia-Dokumente erzeugt werden sollen müssen die Arbeitsplätze auch die Möglichkeit zur Aufnahme von Audio und Video bieten.
  - Die Arbeitsplätze müssen über einen ausreichend dimensionierten Netzanschluß verfügen damit die anfallenden Bandbreiten für Audio und vor allem Video bewältigt werden können.
- Spezielle Anforderungen:
  - Die MM-Dokumente dürfen nicht plattformabhängig sein, da die relevanten Workflow Management Systeme in weitverteilten und heterogenen Systemumgebungen arbeiten müssen, ist es notwendig, daß auch die Dokumente auf allen Systemen bearbeitet werden können.

- Es muß ein Datenhaltungssystem vorhanden sein, in dem die Multimedia-Daten gehalten werden. Da im Regelfall die Endgeräte der Benutzer nicht über die Ressourcen verfügen viele Multimedia-Dokumente aufgrund der großen Datenmenge zwischenspeichern. Zu dem Datenhaltungssystem gehören unter anderem spezielle Medienserver für Video, Audio usw.
- Zum effizienten und transparenten Datenzugriff muß ein systemweites Schema für die Dokumente zur Verfügung stehen.

### **Standards für Verbunddokumentes**

Durch Verbunddokumente wird sich die Interaktion mit Dokumenten und Anwendungen sich grundlegend ändern. Bisher mußte der Anwender, wenn er ein Dokument erstellt hat, das verschiedenen Medien enthielt, mit vielen verschiedenen Anwendungen in unterschiedlichen Fenstern arbeiten. Jede Anwendung stellte dem Anwender die Medieninformation der Einzelmedien wie z.B. Text, Tabellen, Formulare, Bilder, Animationen oder digitalisiertes Video zur Betrachtung und Bearbeitung zur Verfügung. Der Benutzer ist sich in einer solchen Arbeitsumgebung immer bewußt, daß er mit verschiedenen Medien, Anwendungen und Daten arbeitet, um ein Dokument zu erstellen.

Angenommen die Formate wären kompatibel, könnte die Arbeit in einem einzigen Dokument zusammengefaßt werden, um es anzuzeigen oder zu drucken. Dazu müßte der Anwender die ausgewählten Daten in das gemeinsame Dokument, das von einer Anwendung verwaltet wird, übertragen. Dies kann durch Operationen unterstützt werden, die als „Cut and Paste“ bekannt sind, allerdings müssen diese Operationen von der Quell- und Zielanwendung unterstützt werden. Wenn Änderungen an den Daten notwendig sind, werden sie mit der ursprüngliche Anwendung durchgeführt und durch explizite Aktualisierungsoperationen propagiert, z.B. durch wiederholte „Cut and Paste“ Transferoperationen.

Als Alternative zu dieser Vorgehensweise, wird durch die Verbunddokumente ein Modell eingeführt, daß das Dokument und die darin enthaltenen Daten in den Mittelpunkt stellt und nicht mehr die Anwendungen. Ein Verbunddokument ist so gesehen ein vereinheitlichter Container für Daten mit verschiedenen Medientypen. Die Mechanismen, die die Inhalte des Container verwalten merken sich die Beziehungen zwischen den Daten und den Anwendungen mit denen die Daten erzeugt und bearbeitet werden können. Das Verbunddokument verwalten ebenfalls die Darstellung von Daten und die Benutzerinteraktion mit den Daten und Anwendungen. Die Container stellen Schnittstellen und Laufzeitunterstützung zur Verfügung, um den Datenaustausch zwischen den Anwendungen zu unterstützen.

Dem Benutzer wird das Verbunddokument in einem Fenster, als funktionelle und nahtlose Integration mit allen anderen wichtigen Anwendungen, präsentiert, dies umfaßt die Darstellung, Bearbeitung, Speicherung und Druck des Verbunddokuments. Durch diese Technologie erhält der

Benutzer ein vereinfachtes Interaktionsmodell, um mit Multimedia-Daten umzugehen und sie innerhalb und zwischen Verbunddokument zu verschieben. Durch Verbunddokumente besteht nicht mehr die Notwendigkeit explizit zwischen Anwendungen und Fenstern zu wechseln, um die Daten mit verschiedenen Medientypen zu bearbeiten.

Die wichtigsten Konzepte für das Datenmanagement und die Darstellung von Verbunddokumenten umfaßt Querverweise (*linking*), Einbettung (*embedding*), mehrstufige Schachtelung (*nested containment*), direkte Bearbeitung und die Kontrollinfrastruktur

Daten werden in Verbunddokumente entweder durch Querverweise oder durch Einbettung eingefügt. Bei Querverweisen wird nur eine Referenz auf die Originaldaten eingefügt, die Daten können sich an beliebigen anderen Orten bzw. Dokumenten befinden. Querverweise stellen sicher, daß alle Änderungen an den Daten konsistent zur Verfügung gestellt werden, z.B. wenn die referenzierten Daten dargestellt, wiedergegeben (Video/Audio) oder gedruckt werden sollen. Außerdem verringert sich die Speicherplatzanforderungen von Daten die in mehreren Dokumenten verwendet werden. Ein Nachteil von Querverweisen ist das bei Änderungen des Zugriffspfades auf die Originaldaten der Querverweis inkonsistent wird.

Bei der Einbettung werden die Daten und die Beziehung zu den Anwendungen physikalisch in dem Verbunddokument gespeichert. Eingebettete Daten können ohne Probleme mit dem Verbunddokument verschoben, kopiert und weitergegeben werden und weiterhin bearbeitet werden, solange die dazu notwendige Anwendung zur Verfügung stehen. Jedoch steigt mit jeder Kopie der benötigte Speicherplatz und die Daten der Kopien sind völlig voneinander entkoppelt, d.h. wenn in einer Kopie die Daten geändert werden, ist es für das System nicht möglich die Daten in der andern Kopie entsprechend zu aktualisieren. Der Benutzer hat bei Einbettung dafür zu sorgen, daß die Daten in den verschiedenen Kopien konsistent bleiben, falls dies erforderlich ist.

Die mehrstufige Schachtelung erlaubt, daß referenzierte und eingebettete Daten wiederum referenzierte und eingebettete Daten enthalten können. Verschachtelung ermöglicht, daß komplexe Dokumente logisch gruppiert werden können, um diese Gruppen als Einheit zu handhaben.

Die direkte Manipulation vereinfacht das Interaktionsmodell, bei der Einbringung und Bearbeitung im Verbunddokument. „Drag and Drop“ erlaubt dem Anwender die Daten durch direkte Mausinteraktion zu verschieben und zu plazieren, sogar zwischen verschiedenen Gruppen in einem Dokument und auch zwischen verschiedenen Dokumenten. Visuelles Bearbeiten (*visual editing*) bietet ein ähnliches und sehr einfaches, auf ein einzelnes Fenster bezogenes Interaktionsmodell, um die vielen verschiedenen Anwendungen zu handhaben mit denen die unterschiedlichen Daten des Verbunddokuments bearbeitet werden. Wenn Daten in einem Verbunddokument ausgewählt werden bewirkt dies die Veränderung der Menüs, Paletten, Werkzeugleisten und anderer Kontrollfunktionen die für die Bearbeitung der speziellen Daten erforderlichen sind. Dies geschieht durch die Ersetzung, Überlagerung und Erweiterung der entsprechenden Dialogobjekte

des Dokumentenhauptfensters. Der Benutzer kann dadurch die Daten im Verbunddokument bearbeiten (*editing in place*) und muß nicht zwischen verschiedenen Anwendungsfenstern wechseln, um Daten verschiedener Medientypen zu bearbeiten.

Das Zusammenspiel zwischen den Anwendungen, die mit den Daten des Verbunddokuments arbeiten, wird durch die Kontrollinfrastruktur geregelt. Die Kontrollumgebung aktiviert die Anwendungen, wenn die Daten, die ihnen zugeordnet sind, ausgewählt werden. Im folgenden wird auf drei Standards für Verbunddokumente näher eingegangen: OpenDoc, OLE und HTML.

- **OpenDoc**

Die Component Integration Laboratories (CI Labs) arbeitet an einem Standard für Verbunddokumente und Komponentensoftware. Die CI Labs sind ein unabhängiges, nicht kommerzielles Konsortium, bei dem Apple, IBM, Taligent, Novell/WordPerfect und SunSoft Mitglieder sind. Apple und IBM haben die meiste Software beigesteuert, zu der unter anderem gehören: a) Apple's Verbunddokumentenstandard und Implementierung b) Bento für die Speicherung strukturierter Daten und die Open Scripting Architecture (OSA) und c) IBM's System Object Model (SOM). Die CI Labs verwalten und vermarkten den OpenDoc Standard, stellt Zertifikate für OpenDoc konforme Implementierungen aus und lizenziert den OpenDoc Quellcode.

Verbunddokumente werden in OpenDoc aus Teilen (*Parts*) zusammengesetzt, die vom Benutzer angeordnet werden. *Parts* enthalten Daten von einem oder mehreren Typen, die auch Multimedia Formate wie digitalisierte Sprache oder Video sein können. Ein Verbunddokument stellt einen Basiscontainer zur Verfügung in dem die *Parts* angeordnet werden können. Benutzer bearbeiten und greifen auf die *Parts* über *Part Handler* zu. Die *Part Handler* bestehen aus Funktionen zum Darstellen, Bearbeiten und Speichern der *Parts*. Es gibt *Part Handler* sowohl zum Bearbeiten (*Editor Part Handler*) als auch nur zum Anzeigen und Drucken (*Viewer Part Handler*). Rahmen (*Frames*) sind Darstellungsbereiche in einem Anwendungsfenster, die die Inhalte der *Parts* dem Benutzer anzeigen. OpenDoc *Frames* können beliebige zusammenhängende Formen annehmen, z.B. Kreise, Dreiecke, Rauten usw. und sie können sich in dem Verbunddokument überlappen. OpenDoc kann verschiedene Versionen der *Parts* verwalten indem die Änderungen zwischen den Versionen gespeichert wird. Kooperative Gruppenarbeit kann somit unterstützt werden.

OpenDoc enthält einen Automatisierungstechnologie, die als *Open Scripting Architecture (OSA)* bezeichnet wird. Durch die *OSA* können *Parts* so manipuliert und koordiniert werden, daß sie ohne Benutzerinteraktion zusammenarbeiten können. Die *OSA* ist inhaltsorientiert, da die *Part Handler* über Skripte gesteuert werden können. Jeder Typ eines *Parts* in einem Verbunddokument hat sein eigenes Modell wie der Inhalt, die Daten und Operationen beschrieben werden.

Operationen, die oft polymorph gegenüber dem Typ des *Parts* sind, schließen unter anderem Aktionen wie die Selektion, das Löschen und das Setzen von Eigenschaften ein. *Part Handler* lassen zur Laufzeit die Datenobjekte die sie enthalten und die Operationen die sie anbieten registrieren. *OSA* definiert ein Standardnomenklatur von semantischen Ereignissen, Datennoten und Operationsverben, um die Konsistenz zwischen verschiedenen Applikationen und Skriptsprachen zu fördern. Während der Skriptausführung benutzt der OpenDoc Ereignisverteiler die Eintragungen, um die semantischen Ereignisse an die zuständigen *Part Handler* weiterzuleiten. OpenDoc erlaubt es auch Sequenzen von Benutzerereignissen aufzuzeichnen, die in *OSA* konforme Skripte umgesetzt werden können. So können Vorgänge automatisiert werden, z.B. für sich ständig wiederholende Aktionen.

Die OpenDoc Verbunddienste (*Compound Services*, früher auch *Parts Framework* genannt) stellen die Softwareinfrastruktur zur Verfügung, um OpenDoc Verbunddokumente verarbeiten zu können. Diese Dienste unterstützen Plattformunabhängigkeit, sowie Interoperabilität mit Anwendungen die nicht dem OpenDoc Standard entsprechen. Die wichtigsten Dienste sind die Komponentenregistratur (für die Automatisierung), die persistente Datenspeicherung, der Datenaustausch und die Ressourcenaushandlung.

*Bento* ist das System von Apple zur Speicherung von strukturierten Daten. Die verschiedenen *Parts* eines Verbunddokuments werden in einem Bento-Container als eine Sammlung von Strömen dargestellt, die jeweils persistent gespeichert werden. Zwischen den verschiedenen Strömen können Referenzen angelegt werden. *Bento* erlaubt die Integration und dynamische Wiederverwendung von OpenDoc *Parts* durch Endbenutzer und Anwendungs-*Part Handler* in dem Dokument. *Parts* können geschachtelt werden. So können komplexe Hierarchien geschaffen werden. *Bento* überwacht die Identität und Platzierung der *Parts* in den Dokumenten und veranlaßt das Laden aus dem persistenten Speicher in den Hauptspeicher und umgekehrt, nach den Anforderungen der *Part Handler*. Ebenfalls verwaltet *Bento* Indizes, mit denen komplexe Beziehungen zwischen Dokumentteilen ausgedrückt werden können. Diese werden nicht nur von OpenDoc zur Navigation verwendet werden, sondern können auch von jeder anderen Anwendung, die das *Bento*-Modell versteht, zur Navigation genutzt werden.

Das Datenaustauschmodell von OpenDoc basiert auf OpenDoc's *Bento* Speicherdienst. Es werden die gleichen Operationen zum Austausch verwendet, die zur Speicherung in Dateien benutzt werden. Der Datenaustausch kann innerhalb von Dokumenten, aber auch zwischen Dokumenten durch folgende Operationen stattfinden:

- *Drag and Drop*
- Copy and Paste
- Referenzieren (*Linking*)

Das Datentransfermodell erlaubt die Verlagerung von ganzen *Parts* in einer Operation, auch wenn diese eingebettet und geschachtelt sind. Es erlaubt auch die Platzierung in bereits bestehenden mehrfach geschachtelten *Parts*.

Referenzierung, die dritte Datentransferoperation, die OpenDoc unterstützt, basiert auf dem *publish and subscribe* Modell. Benutzer registrieren Teile von Dokumenten als öffentlich zugänglich, andere Anwendungen können diese Teile durch Referenzieren verwenden. Das Datentransfermodell unterstützt auch die Propagierung von Änderungsereignissen vom Produzenten der Änderung zum Konsumenten der Daten. Dies funktioniert auch wenn die Konsumenten nur über Referenzen auf die Daten zugreifen.

Mit dem *System Object Model (SOM)* von IBM wird der *Object Management Service (OMS)* von OpenDoc realisiert. Der *OMS* bildet das kritische Element zur Gewährleistung der Interoperabilität zwischen den OpenDoc *Parts*, den *Part Handlern* und anderen Softwarekomponenten. SOM ist ein sprachunabhängiger und CORBA konformer Object Request Broker der lokale und nicht lokale Kommunikation zwischen OpenDoc Objekten realisiert.

- **OLE 2.0**

Object Linking and Embedding (OLE) besteht sowohl aus standardisierten Spezifikationen wie auch standardisierten Implementierungen von Microsoft. OLE 2 erfüllt auch die Anforderungen für Verbunddokumente, wie sie im allgemeine Teil und bei OpenDoc beschrieben wurden. Die Architektur und die Implementierung unterscheidet sich allerdings.

Wie OpenDoc basiert auch die Technologie von OLE 2 auf einer komponentenbasierten Architektur. Diese Architektur baut auf dem *Component Object Model (COM)* Standard auf und stellt die Interoperabilität auf der binären Ebene zwischen Anwendungskomponenten, die durchaus in verschiedenen Sprachen programmiert wurden, sicher. COM unterstützt ein Client/Server Modell, bei dem die Klienten den Dienstnutzern entsprechen, OLE-Objekte sind Daten und OLE-Server sind Anwendungen die OLE-Objekte und die dazugehörigen Dienste implementieren.

Wichtige Elemente in COM sind die Schnittstellen (*Interfaces*), die eine Menge von zusammengehörigen Funktionen zusammenfassen, wie z.B. „Drag and Drop“. OLE Dienstschnittstellen werden in COM separat von den Implementierungen in den Serveranwendungen verwaltet. In COM, greifen die Klienten ausschließlich über Zeiger, in virtuelle Tabellen, auf die Schnittstellen zu. Die virtuellen Tabellen bestehen wiederum aus Zeigern auf Funktionen. Die Implementierung dieser Funktionen bestimmen schließlich das Dienstverhalten bzw. die Methoden der Dienste. Alle OLE-Objekte haben eine oder mehrere Schnittstellen, die jeweils Zeiger auf Funktionstabellen bereitstellen.

Der OLE Verbunddokumentenstandard spezifiziert wie Server ihre Objekte gruppieren, so daß Container in beliebigen Anwendungen diese Objekte in den eigenen Dokumenten verwenden können. OLE unterscheidet dabei zwischen zwei Datentypen in Verbunddokumenten, dies sind die Präsentationsdaten, die dazu verwendet werden die Datenobjekte auf dem Bildschirm oder sonstigen Ausgabegerät darzustellen und die Basisdaten die eine Anwendung zur Bearbeitung der Datenobjekte benötigt. Eingebettete Objekte (*embedded objects*) enthalten beide Typen und somit ist es möglich sie zu aktivieren und direkt zu bearbeiten (*editing in place*). Referenzierte Objekte (*linked objects*) enthalten nur die Darstellungsdaten, die Ursprungsdaten werden nur im Ursprungsobjekt gespeichert.

Die OLE Verbunddokumenten Schnittstellen sind in folgenden funktionale Gruppen unterteilt:

- Verbundobjekte
- Verbunddokumente
- Referenzierung
- Datentransfer und Zwischenspeicherung
- Drag and Drop
- Persistente Speicherung
- Vor-Ort-Aktivierung
- Automation

Alle OLE-Objekte müssen die Verbundobjekt-Schnittstelle enthalten, sie spezifiziert wie die Klienten mit dem Objekt kommunizieren. Durch diese Abfrageschnittstelle kann sich der Klient informieren welche weiteren Schnittstellen das Objekt unterstützt und außerdem die Eigenschaften des Objekts dynamisch erfragen, ohne Vorkenntnisse über das Objekt zu haben. Für die Inter-Prozeßkommunikation werden leichtgewichtige Remote Procedure Calls (RPCs) verwendet, die blockierend sind (synchron) und einen transparenten Datentransfer in oder zwischen Prozessen auf einer Plattform gewährleisten.

Das hierarchische Speichermodell von OLE verwendet *Storages* zur Verwaltung der Verbunddokumente. Die *Storages* sind in der Form von Verzeichnissen angeordnet, damit eine effektive Navigation möglich ist. *Storages* enthalten einen Strom je OLE-Objekt. Die Ströme entsprechen normalen Dateien. Die OLE-Implementierung des OLE-Speichermodells wird als Verbunddateien (*compound files*) bezeichnet. OLE-fähige Anwendungen verwenden diese Verbunddateien zum effizienten und flexiblen Zugriff auf die Daten. Wenn Änderungen gemacht werden, können durch die Verbunddateien diese Änderungen direkt in das Objekt übernommen werden. Oder die Änderungen werden getrennt von der alten Version verwaltet bis der Benutzer explizit die Speicherung der gemachten Änderungen forciert (transaktionsori-



entiert). Wie bei *Bento* müssen zur Anwendungsentwicklung, zur Datenverwaltung und Speicherung keine Verbunddateien verwendet werden, dann geht allerdings die Performanz und die Integrationseigenschaft verloren.

Durch das Modell für den Datentransfer in OLE können Daten einheitlich durch *Drag and Drop*, *Copy and Paste* oder APIs ausgetauscht werden. Das OLE-Transfermodell wird als *Uniform Data Exchange* bezeichnet und unterscheidet zwischen der Vorbereitungsphase für den Datenaustausch und der eigentlichen Transferoperation. Von den Quelldaten wird ein Objektzeiger zu den Datensinken transferiert. Nachdem der Zeiger übertragen wurde wird das OLE Protokoll nicht weiter benötigt und der tatsächliche Datentransfer wird durch die Objekte direkt durchgeführt. Die Quelle stellt das Datenobjekt zur Verfügung, regelt die Operationsinitiierung bzw. Operationsterminierung und kontrolliert das Benutzeroberflächeninterface. Die Senke empfängt das Datenobjekt, überprüft die Brauchbarkeit des Formats und legt die Operationen fest die ausgeführt werden wenn das Datenobjekt über dem Zielfenster abgelegt wird. Die Objektbeschreibung enthält Informationen zur Objektdarstellung und zum Medientyp, um eine geeignete Transportmethode auszuwählen, z.B. eine Datei im Sekundärspeicher, einen Datenstrom oder gemeinsamen Speicher.

Die OLE Automation erlaubt es Anwendungen ihre funktionalen Schnittstellen offenzulegen, dies hat den Effekt, daß andere Anwendungen die Schnittstellen dynamisch und programmgesteuert abfragen und aufrufen könne. Im Gegensatz zur *OSA* von OpenDoc bietet OLE-Automation weder ein standardisiertes Vokabular zur einheitlichen Programmierung von Makro- und Skriptsprachen noch unterstützt es die Aufzeichnung von Aktivitätssequenzen der Anwender.

- **HTML**

Die Hypertext Markup Language (HTML) dient der Präsentation von Multimedia Daten. HTML wird als Standardseitenbeschreibungssprache im Internet/WWW verwendet. HTML kann auch für lokale Präsentationen verwendet werden z.B. für Präsentationen auf CDs.

Mit der HTML können nicht nur multimediale Seiten beschrieben werden, sondern es können auch verschiedene Seiten und deren Inhalte mit einander verknüpft werden. Dazu wird in HTML ein Referenzierungskonzept (*Linking*) verwendet.

HTML ist im Gegensatz zu OpenDoc oder OLE nicht zur WYSIWIG-Bearbeitung geeignet.

Neben der Seitenbeschreibung und der Informationsvernetzung enthält die HTML auch noch Sprachelemente für Dialoge. Damit können HTML-Formulare erstellt werden und auf Serverseite verarbeitet werden. Somit sind mittels HTML einfache Interaktionen möglich.

In HTML können nicht nur verschiedene Medientypen und -formate verwendet werden sondern es kann auch Anwendungslogik integriert werden. HTML stellt allerdings nur den optischen Rahmen für die Programme zur Verfügung, die Programmiersysteme selber sind nicht in HTML definiert. Von Relevanz sind z.Z. vor allem Java, Javascript und ActiveX.

### **Schwachstellenanalyse**

OpenDoc und OLE sind Standards für Multimedia Verbunddokumente, die den meisten Anforderungen gerecht werden, die durch die Integration in Vorgangssteuerungssysteme entstehen. HTML kann nicht direkt mit OLE und OpenDoc verglichen werden, da es sich um eine Seitenbeschreibungssprache handelt und nur in sehr begrenztem Rahmen Interaktion mit dem Anwender zulässt. Im Detail betrachtet gibt aber für die Standards folgende Vor- und Nachteile:

#### **OLE**

- Vorteile
  - Bereits hohe Verfügbarkeit auf PC-Basis
  - Viele Standard PC Anwendungen unterstützen OLE
- Nachteile
  - Proprietärer Standard
  - Nicht für allen relevanten Systemarchitekturen im vollen Funktionsumfang erhältlich
  - Beschränkte Funktionalität (im Vergleich mit OpenDoc)
  - Aufwendigere Anwendungsimplementierung (im Vergleich mit OpenDoc)

#### **OpenDoc**

- Vorteile
  - OpenDoc ist ein offener Standard (nicht proprietär)
  - OpenDoc ist auf vielen Rechnerarchitekturen im vollen Funktionsumfang verfügbar
  - Besser geeignet für Component Ware
- Nachteile
  - Noch in der Entwicklungsphase, daher noch in wenigen Anwendungen verfügbar

#### **HTML**

- Vorteile
  - HTML ist ein offener Standard
  - HTML ist auf vielen Rechnerarchitekturen im vollen Funktionsumfang verfügbar
- Nachteile
  - Bietet nur begrenzte Interaktionsmöglichkeiten

- Bietet nur Rahmen für Component Ware, z.B. Java, Java Beans

Ein besonders schwerwiegender Nachteil, der sowohl OpenDoc wie auch OLE betrifft, ist die fehlende Möglichkeit zur Modellierung von Sicherheitsaspekten. Dies erschwert bzw. verhindert noch den direkten Einsatz von Verbunddokumenten nach dem OpenDoc- bzw. OLE-Standard in Vorgangssteuerungssystemen, da gerade in Büro- und Verwaltungsumfeld es besonders wichtig ist zwischen Dokumententeile bezüglich des Zugriffsschutzes und der Bearbeitungsrechte zu differenzieren. Selbst bei einem so einfachen Formular wie dem Urlaubsantrag muß es möglich sein die verschiedenen Felder für Unterschriften so zu schützen damit nur die autorisierten Personen dort ihre Signatur plazieren können. Bei OpenDoc und OLE sollen diese Sicherheitsaspekte in Zukunft von den darunterliegenden Objektsystemen zur Verfügung gestellt werden. OpenDoc soll auf OMA/CORBA der OMG abgebildet werden. OLE soll auf COM basieren. COM ist ebenfalls ein proprietärer Standard von Microsoft. Inwieweit dadurch die Sicherheitsanforderungen erfüllt werden können kann erst geprüft werden wenn die dazugehörigen Implementierungen verfügbar sind.

### **8.2.3 Synchroner Multimedia Technologien**

Desktop Konferenzen bezeichnen weniger eine Technologie sondern vielmehr eine Arbeits- und Kommunikationsweise. Desktop Konferenzen bezeichnen die synchrone Kommunikation und Kooperation von verschiedenen, unter Umständen an weit entfernten Orten arbeitenden Personen. Die Kommunikation und Kooperation erfolgt mittels rechnergestützten Werkzeugen, die wie jede andere Anwendung mittels der Benutzerschnittstelle bedient werden. Desktop Konferenzsysteme setzen sich aus verschiedenen funktionalen Komponenten zusammen. Nicht jedes Desktop Konferenzsystem muß alle Komponenten enthalten, allerdings ist als Mindestmaß ein Kommunikationskanal und das Sessionmanagement notwendig.

#### **Sessionmanagement**

Das Sessionmanagement kann unterschiedlich komplex sein. Die einzelnen Komponenten können ein implizites Sessionmanagement beinhalten. Es kann allerdings auch ein komplexes Management geben, bei dem verschiedene Komponenten einheitlich koordiniert werden, Rechte zu überwachen sind, Einladungen für Konferenzen verschickt werden müssen und Eingaberechte für replizierte Anwendungen nach festgelegten Politiken zu vergeben sind.

Zum Zeitpunkt der Untersuchungen im Rahmen des Arbeitspaktes gab es noch keinen Standard wie ein gemeinsames Management einzelner Komponenten bzw. zwischen verschiedenen Komponenten realisiert werden kann. Daraus ergibt es sich, daß es nur komplett Lösungen gibt und es heute noch nicht möglich ist beliebige Komponenten eines Desktop Konferenzsystems auszutauschen.

## Video/Audio

Applikationen für Audio und Video Verbindungen sind die auffälligsten Komponenten in einem Desktop Konferenzsystem. Von diesen Komponenten werden auch die härtesten Anforderungen gestellt. Hierfür ist vor allem die synchrone Datenübertragung zu nennen, um sie zu gewährleisten ist die Reservierung von Dienstgütern (Quality of Service, QoS) notwendig. Bei der Dienstgüte handelt es sich um die CPU-Zeit, reservierter Speicherplatz für Puffer und Netzwerkbandbreite.

- Verzögerung/Varianz

Die Übertragungsqualität von Video und Audio wird mit zwei Werten maßgeblich beeinflusst: Die Verzögerung und die Varianz der Verzögerung.

Die Verzögerung bestimmt die Zeit die zwischen Aufnahme und Wiedergabe eines kontinuierlichen Medientyps vergeht. Die Verzögerung ergibt sich aus den Verarbeitungszeiten die in den an der Übertragung beteiligten Komponenten anfallen. Zu den Komponenten gehören im wesentlichen die Aufnahmeeinheit und die Wiedergabeeinheit, die zur Digitalisierung und gegebenenfalls zur Kompression bzw. Dekompression verwendet werden. Als zeitkritischste Komponente zählt die Netzwerkverbindung, hier können je nach Verbindungstyp und Verbindungslänge erhebliche Verzögerungen entstehen.

Die Varianz gibt an wie stark die Verzögerungszeit von der durchschnittlichen Verzögerung abweicht. Desto höher die Varianz ist je mehr Ressourcen müssen auf den Systemen zur Verfügung gestellt werden, da Video/Audio kontinuierlich wiedergegeben werden muß. Die Ressourcen zur Zwischenspeicherung werden zum Ausgleich der Varianz benötigt.

- Synchronisation von Audio und Video

Damit bei der Verwendung von kontinuierlichen Medientypen der Gleichlauf zwischen den Medien und innerhalb eines Medienstroms gewährleistet werden kann müssen die Medienströme synchronisiert werden.

Wenn der Transport von Daten in einem Strom zur Verringerung der Varianz synchronisiert wird, wird von Intraströmsynchronisation gesprochen. Die Intraströmsynchronisation verringert den Ressourcenbedarf der für den Transport notwendig wird, da weniger Zwischenpuffer notwendig sind.

Die Synchronisation von mehreren verschiedenen Medienströmen wird als Interströmsynchronisation bezeichnet. Bei der Interströmsynchronisation wird dafür gesorgt das alle Ströme im zeitlichen Gleichlauf präsentiert werden, dazu muß die Verzögerung der verschiedenen Medienströme aneinander angepaßt werden.

- Video

Die Übertragung von Videodaten fallen sehr hohe Datenmengen pro Zeiteinheit an und diese müssen mit möglichst konstanter Verzögerung transferiert werden. Das Datenvolumen kann durch Kompression verringert werden, dazu sind spezielle Verfahren notwendig, da herkömm-

liche Verfahren nicht effizient genug sind, die Unmenge von Daten in kurzer Zeit bei konstantem Durchsatz zu verarbeiten. Im Gegensatz zu „normalen“ Daten ist es aber bei Videodaten nicht notwendig die Daten verlustfrei zu komprimieren. Durch eine angemessene Verwendung von mathematischen Transformationen und der Berücksichtigung der Fähigkeiten bzw. Empfindlichkeit des menschlichen Auges können effiziente Algorithmen verwendet werden, um Videodaten zu komprimieren. Im Gegensatz zu der normalen Bildkompression kann bei der Videokomprimierung auch aufgrund der Eigenschaften aufeinanderfolgenden Bildern (große Ähnlichkeit) komprimiert werden.

Für die Videodatenübertragung gibt es verschiedene (Kompressions-)Standards die anhand des wichtigen Merkmals der benötigten Bandbreite unterschieden werden können. Die eine Gruppe von Standards arbeitet mit konstanten Bandbreiten für die Übertragung der Videodaten. Das hat vor allem bei leitungorientierten, exklusiv nutzbaren Datendirektverbindungen Vorteile. Es wird die komplette Bandbreite genutzt, es kommt zu keiner Überlastung der Verbindung. Der Nachteil dieser Übertragungsform ist das die Qualität der Übertragenen Bilder nicht konstant ist. Um einen konstante Bandbreite zu erhalten muß je nach Bildinhalten unterschiedlich stark komprimiert werden und dies führt zu einer sich verändernden Bildqualität.

Bei konstanter Bildqualität ändert sich die benötigte Bandbreite und Ressourcenbelastung ständig. Deswegen ist es schwierig die richtige Ressourcenreservierung durchzuführen, denn teilweise werden die Ressourcen nicht genutzt und teilweise kommt es zur Überlastung der zur Verfügung stehenden Ressourcen. Wenn allerdings eine konstante Bildqualität gefordert ist muß diese Randbedingung hingenommen werden.

- **Audio**

Die Verwendung von Audio erfordert eine noch optimiertere Ressourcenzuteilung. Im Gegensatz zu Video macht sich bei Audio ein noch so geringer Jitter sofort bemerkbar. Das Datenaufkommen ist normalerweise im Vergleich mit Video eher geringer einzuschätzen, dafür müssen die Audiodaten aber mit möglichst geringer Varianz übertragen werden.

### **Application Sharing**

Neben der direkten audiovisuellen Kommunikation zählt das Application Sharing ebenfalls zu den Kommunikationsmitteln einer Desktop Konferenz. Beim Application Sharing wird jedem Konferenzteilnehmern eine replizierte Darstellung einer prinzipiell beliebigen Anwendung dargestellt. Alle am Application Sharing beteiligten Personen sehen den gleichen Zustand der Applikation und können in der Konferenz über die Daten und Zustand der Anwendung reden.

Für die Eingabe kann nicht die gleiche Politik wie für die Darstellung der Applikationen verwendet werden, denn bei der Eingabe muß geregelt sein wer, wann, was eingeben darf, da der Großteil der Anwendungen nur die Eingabe von einem Benutzer verarbeiten kann. Meistens wird ein Token zwischen den Benutzern vergeben, daß das Recht zur Eingabe regelt.

Neben normalen Anwendungen, wie Editoren, Textverarbeitungssysteme oder Tabellenkalkulationen, gibt es Anwendungen die normalerweise nur in Verbindung mit Applikation Sharing sinnvoll an einem rechnergestützten Arbeitsplatz verwenden lassen. Diese Applikationen dienen als weiteres Kommunikationsmittel und können die Eingaben von verschiedenen Benutzern gleichzeitig verarbeiten, z.B. Whiteboards.

Zur Unterstützung der Kommunikation über die Inhalte die mittels des Application Sharing allen Konferenzteilnehmern zugänglich gemacht werden ist ein Telepointing bzw. Telemarking Werkzeug notwendig. Ein solcher elektronischer Zeigefinger dient dazu den Fokus auf den Punkt zu bringen, zumindest soweit er sich innerhalb der aktuell replizierten Anwendungen befindet.

### **Applications on Shared Data**

Beim Application Sharing wird die graphische Darstellung einer Applikation bei den Teilnehmern der Konferenz repliziert. Dies wird gemacht, um die Inhalte gemeinsam zu bearbeiten bzw. um über sie zu diskutieren. Es haben also alle Teilnehmer ständig die gleiche Sicht auf die gleichen Daten.

Für Desktop Konferenzen gibt es auch Anwendungen bei denen jeder Teilnehmer zwar auf den gleichen Daten, nicht aber auf der gleichen Sicht der Daten arbeitet. So kann es sinnvoll sein ein Konferenzprotokoll zu führen zu dem jeder online Zugriff hat, was aber nicht bedeuten muß, daß jeder zu jedem Zeitpunkt die gleiche Sicht auf die Daten hat.

### **Anforderung an Desktop-Konferenzsysteme**

Es werden auf verschiedenen Ebenen Anforderungen an ein Desktop-Konferenzsystem gestellt. Es werden zuerst die Anforderungen bezüglich des Kommunikationsbedarfs erfaßt, als nächstes werden die Anforderungen beschrieben, die die Integration betreffen und schließlich werden die wichtigsten Qualitätsanforderungen aufgezählt.

### **Kommunikationsbedarf**

Bei der Abwicklung von Geschäftsprozessen besteht Bedarf an informeller, spontaner bzw. kreativer synchroner Kommunikation. Dieser Bedarf wird bisher nicht durch Workflows Management Systeme unterstützt. Dazu im Gegensatz steht die formelle asynchrone Kommunikation der klassischen Workflow Management Systeme.

Die synchrone Kommunikation kann in zwei Klassen eingeteilt werden, die für Geschäftsprozesse relevant sind:

- Spontane Kommunikation: z.B. Rückfragen, Krisenmanagement
- Geplante Kommunikation: z.B. Besprechungen, Diskussion

## **Integrationsbedarf**

Vorgänge sollen in ihrer Gesamtheit von dem Workflow Management System abgewickelt werden, daher ist es notwendig sowohl strukturierte als auch unstrukturierte Aktivitäten zu unterstützen. Bisher konnten durch Workflow Management Systeme nur strukturierte Arbeitsformen unterstützt werden, zukünftig sollen durch die Integration von Telekooperationstechniken auch unstrukturierte Arbeitsformen Berücksichtigung finden. Die Integration sollte dabei so weit gehen, daß den Benutzern der Übergang von strukturierter Arbeit zu unstrukturierter Arbeit und umgekehrt nahtlos möglich ist.

Für die nahtlose Integration müssen daher verschiedene Anforderungen erfüllt sein:

- Das Desktop Konferenzsystem muß die Organisationsstruktur der Workflow Management Umgebung verwenden können bzw. ein Konverter muß den Namensraum der Organisationsstruktur auf den Namensraume des Desktop Konferenzsystems abbilden können.
- Die Desktop Konferenzsysteme müssen unabhängig von der Systemplattform sein, da die relevanten Workflow Management Systeme, in die sie integriert werden sollen, ebenfalls in weitverteilten und heterogenen Systemumgebungen arbeiten. Daher ist es notwendig, daß auch die Konferenzsysteme in der heterogenen Umgebung ihre Funktionalität erbringen.
- Das Konferenzsystem muß über eine offene Programmierschnittstelle zu steuern sein. Folgende Funktionen sind erforderlich:
  - Lokation und Bereitschaft zur Teilnahme an einer Konferenz klären.
  - Konferenz aufbauen, unter Angabe der Teilnehmer und der zu verwendenden Medien (Audio/Video) und Qualitäten.
  - Neue Teilnehmer müssen zu einer laufenden Konferenz hinzugenommen werden können, sowohl Teilnehmer als auch System gesteuert.
  - Teilnehmer müssen aus einer laufenden Konferenz ausscheiden können, sowohl Teilnehmer als auch System gesteuert.
  - Die Konferenz muß beendet werden können.
- Es müssen Möglichkeiten zur Konfiguration der Funktionalität bestehen. Die Möglichkeit zur Konfiguration ist notwendig, damit nur die Komponenten und Funktionen des Desktop Konferenzsystems von den Benutzern verwendet werden, die nach der Vorgangsdefinition erlaubt sind. So kann es z.B. Sicherheitsvorschriften geben die es verbieten das vertrauliche Informationen/Dokumente nicht beliebigen anderen Benutzern über das Application Sharing zugänglich gemacht werden dürfen.

### Qualitätsanforderungen

Der Begriff „Quality of Service“ (QoS) fasst mehrere Parameter zusammen, die die Dienstgüte von Multimedia Strömen beschreiben. Die folgende Aufzählung ist nicht vollständig enthält aber die wichtigsten Parameter:

- Isochrone Datenübertragung: Intraströmsynchronisation (Vermeidung von Jitter), Interstromsynchronisation (Synchronisation von verschiedenen Medien, (vgl. Tabelle 8.2))

Media		Mode, Application	QoS
Video	animation	correlated	+/- 120 ms
	audio	lip synchronization	+/- 80 ms
	image	overlay	+/- 240 ms
		non overlay	+/- 500 ms
	text	overlay	+/- 240ms
		non overlay	+/- 500ms
Audio	animation	event correlation	+/- 80 ms
	audio	tightly coupled (stereo)	+/- 11 $\mu$ s
		loosley coupled (dialog mode with various participants)	+/- 120 ms
		loosely coupled (e.g. background music)	+/- 500 ms
	image	tightly coupled (music with notes)	+/- 5 ms
		loosely coupled (slide show)	+/- 500 ms
	text	text annotation	+/- 240 ms
	pointer	audio relates to showed item	- 500/+750 ms

**Tabelle 8.2:** Dienstgüte für Medien

- Verzögerung (Delay): Echtzeitkommunikation
- Bandbreite: Hochgeschwindigkeitsverbindungen, Rechnersystemdurchsatz
- Mehrpunktverbindungen: Multicastfähigkeit des Netzwerkes, Gruppenbildung

### Produktbeschreibung und Bewertung

Es werden hauptsächlich zwei Desktop Konferenz Systeme untersucht: Berkom MMC und Com-muniquen von Insoft. Dazu im Vergleich wird CIO-JVTOS beschrieben.

- BERKOM MMC



Der BERKOM Multimedia Collaboration Service (MMC Version 3.3beta) [Gleser et al., 1994] ist ein vollwertiges Desktop Konferenzsystem, da der MMC alle wichtigen funktionalen Komponenten enthält, die eingeführt wurden. Außerdem erfüllt es viele wichtige Anforderungen die an Desktop Konferenzsysteme gestellt werden, wenn sie in Workflow Management Systemen integriert werden sollen.

- **Sessionmanagement:**

Der MMC stellt ein sehr umfangreiches Management von Konferenzen bereit. Neben der Mindestfunktionalität, wie Konferenzöffnung, Einladung, Hinzunahme und Ausschluß von Teilnehmern können auch Konferenzen zusammengelegt und getrennt werden. Außerdem können mit dem Sessionmanagement Rollen vergeben werden, denen besondere Funktionalität und Rechte zugeordnet sind, wie Initiator, Konferenzleiter, Tokenhalter (Application Sharing), Teilnehmer, Beobachter. Es besteht auch die Möglichkeit Konferenzen offen oder geschlossen zu halten. Dies bedeutet das entweder nur explizit eingeladene Benutzer an einer Konferenz teilnehmen können oder beliebige Benutzer sich in eine laufende Konferenz einklinken können.

Der MMC stellt auch ein Art von Telefonbuch bereit, daß alle benötigten Information über alle potentiellen MMC-Benutzer enthält.

- **Audio/Video-Kommunikation:**

Für die Audio/Video Kommunikation sind verschiedene Optionen möglich. Es kann ein AV Mode eingestellt werden der angibt welche Art der audiovisuellen Kommunikation gewählt werden soll, es sind vier verschiedene Arten möglich:

- Keine audiovisuelle Kommunikation
- Nur Audio Kommunikation
- Nur Video Kommunikation
- Vollständige audiovisuelle Kommunikation

Außerdem kann die Politik angegeben werden wie die Dienstgüte der Kommunikation zu den Konferenzteilnehmern ausgehandelt werden:

- Individuell: Die beste Audio/Video Verbindung zu jedem Teilnehmer wird gewählt
- Gleich: zwischen allen Teilnehmern wird die gleiche (geringste) Dienstgüte gewählt.
- Vorgegeben: Falls eine vorgegebene Dienstgüte nicht zu allen Teilnehmern möglich ist kommt die Konferenz nicht zustande.

- **Application Sharing:**

Das Application Sharing ist in den MMC nahtlos integriert, d.h. es wird über den MMC angegeben welche Anwendungen repliziert werden, ebenfalls wird mittels des MMC ange-

geben welche Rechte die Teilnehmer zur Eingabe in die Anwendung haben. Auch die Mechanismen und Politiken zur Weitergabe der Rechte wird über das Sessionmanagement des MMC kontrolliert.

Das Application Sharing des MMC verfügt über Telepointer, die einzelnen Telepointer sind an jeweils genau ein Fenster gebunden, dies ist für Applikationen mit mehreren Fenster nur suboptimal.

Zur Zeit stehen zwei verschiedene Werkzeuge zum Application Sharing zur Verfügung, HP SharedX<sup>1</sup> und xy.

- **Bewertung:**

Grundsätzlich ist der BERKOM MMC gut geeignet für Desktop Konferenzen, bei genauerer Prüfung sind folgende Punkte besonders zu erwähnen:

Für den MMC sollen API's zur Verfügung gestellt werden, die Protokolle und deren Primitiven sind bereits veröffentlicht und es scheint, daß damit eine Integration in ein Vorgangsteuerungssystem möglich sein wird.

Da der MMC eine eigene Benutzerverwaltung enthält, muß eine Konverter zwischen dem Workflow Organisationsdienst und der MMC Verwaltung benutzt werden. Überdies muß sichergestellt sein, daß die beiden Dienste zueinander konsistent gehalten werden.

Die Dienstkonfiguration ist nur eingeschränkt möglich.

Der MMC ist auf verschiedenen Unix und Windows Plattformen verfügbar.

Neben den bisher aufgeführten funktionalen Komponenten gehört zum MMC auch ein Whiteboard, das nicht über das Application Sharing repliziert wird, sondern ein eigenes Protokoll verwendet.

- **Insoft Communique**

Insoft Communique Version 4.0 [InSoft, 1994a], [InSoft, 1994b], [InSoft, 1994c] ist ein vollwertiges Desktop Konferenzsystem, Communique enthält alle wichtigen funktionalen Komponenten.

- **Sessionmanagement:**

Communique 4.0 stellt ein sehr intuitives Sessionmanagement bereit. Es wird die Konferenzöffnung, die Einladung, die Hinzunahme und der Ausschluß von Teilnehmern unterstützt.

Communique stellt nur ein sehr einfache Art eines Telefonbuchs bereit, daß Information

---

<sup>1</sup>Bei der Verwendung von SharedX ist die beschriebene Funktionalität und Mächtigkeit der Eingabekontrolle teilweise eingeschränkt.

über Communique Teilnehmer bereithält. Neben diesem Telefonbuch bietet Communique auch eine aktive Netzwerküberwachung an und zeigt potentielle Konferenzteilnehmer an, die zur Zeit im Netzwerk aktiv sind.

- Audio/Video-Kommunikation:

Die audiovisuelle Kommunikation kann anhand sehr vieler Optionen beeinflusst werden:

Es kann angegeben werden zu welchen Teilnehmern Audio bzw. Videoverbindungen unterhalten werden sollen.

Für Video und Audio können getrennt von einander die Rate, Bandbreiten bzw. die Qualität angegeben werden. Es ist möglich sowohl nach maximaler Qualität und nach minimaler Verzögerung zu optimieren.

Es ist möglich den Grad der Synchronität zwischen Audio und Video festzulegen<sup>1</sup>.

- Application Sharing:

Das Application Sharing von Communique ist sehr einfach gehalten. Je Benutzer können nur drei Applikationen (X-Window basiert) repliziert werden. Die Applikationen können nur beim Applikationsstart repliziert werden, nachträgliches replizieren ist nicht möglich.

Es sind keine Mechanismen zur Eingabekontrolle vorgesehen, d.h. alle Teilnehmer können ungehindert Eingaben tätigen, dies kann zu Konflikten führen die von den Teilnehmern selbst aufgelöst werden müssen („durch Zurufen“).

Telepointer werden nicht zur Verfügung gestellt. Ebenso sind die privaten Mauszeiger der Teilnehmer bei den anderen Teilnehmern nicht sichtbar.

- Bewertung:

Grundsätzlich ist Communique 4.0 gut für Desktop Konferenzen geeignet, bei genauerer Prüfung sind folgende Punkte besonders zu erwähnen:

Für Communique und die dazugehörige Programmierumgebung OpenDVE sind API's verfügbar. Die Protokolle und deren Primitiven sind zugänglich und es ist damit eine Integration in ein Vorgangssteuerungssystem möglich.

Da Communique eine gute API zum Management der Benutzerverwaltung zur Verfügung stellt kann der Organisationsdienst eines Vorgangssteuerungssystems an Communique angebunden werden.

Die Dienstkonfiguration ist sehr gut möglich, da alle Komponenten (in OpenDVE Syntax: plugins) des Konferenzsystems hinzugefügt, ersetzt und weggelassen werden können. Es ist auch möglich auf der Basis von OpenDVE eigene Konferenzkomponenten zu realisieren, um sie innerhalb von Communique zu verwenden.

---

<sup>1</sup>Im Praxistest wurde kein Unterschied zwischen den Extrem (eng bzw. lose Synchronisation) festgestellt. Die Tests wurden allerdings auch nur in einem lokalen Netzwerk durchgeführt.

Communique ist auf verschiedenen Systemplattformen verfügbar. Sowohl unter den Unix-Derivaten von SUN (Solaris und SunOS), Hewlett Packard, IBM, Digital Alpha ist Communique verfügbar, als auch unter Windows/NT und IBM kompatiblen PC's (486/Pentium).

Als Netzwerk-Technologien können Ethernet, ISDN, FDDI, Frame Relay und ATM verwendet werden.

Neben den bisher aufgeführten funktionalen Komponenten gehört zu Communique auch ein Whiteboard, das nicht über das Application Sharing repliziert wird, sondern ein eigenes Protokoll verwendet. Außerdem ist auch ein textbasiertes Protokollwerkzeug verfügbar.

- CIO JVTOS

CIO Joint Viewing and Tele-Operation Service (JVTOS Version 3.0) ist ein vollwertiges Desktop Konferenzsystem. Es erfüllt die wichtigsten Anforderungen die an Desktop Konferenzsysteme gestellt werden, wenn sie in Workflow Management Systeme integriert werden sollen.

- Sessionmanagement:

CIO JVTOS stellt ein umfangreiches Management von Konferenzen bereit. Es bietet folgende Funktionalität: Konferenzöffnung, Einladung, Hinzunahme und Ausschluß von Teilnehmern. Außerdem können durch das Sessionmanagement Rollen vergeben werden denen eine besondere Semantik und Rechte zugeordnet sind, wie die Konferenzleitung und das Halten des Eingaberechts in replizierten Anwendungen.

- Audio/Video-Kommunikation:

Die audiovisuelle Kommunikation kann anhand von Optionen beeinflusst werden:

Es kann angegeben werden zu welchen Teilnehmern Audio bzw. Videoverbindungen unterhalten werden sollen.

Das Video und Audio kann in mehreren Qualitäten den anderen Teilnehmern angeboten werden. Als Empfänger von Audio und Video kann aus einer der angegebenen Qualitäten eine ausgesucht werden.

- Application Sharing:

Das Application Sharing ist in JVTOS integriert, d.h. es wird über die Benutzerschnittstelle von JVTOS angegeben welche Anwendungen repliziert werden, ebenfalls ist es auch möglich anzugeben welche Rechte die Teilnehmer für die Eingabe in eine replizierte Anwendung haben. Es können auch Multimedia-Applikationen repliziert werden die kontinuierliche Medien (Audio/Video) enthalten. Die Replizierung von Applikationen ist jederzeit möglich.

JVTOS verfügt über ein sehr komfortables Konzept zur Verwendung von Telepointern. Die

einzelnen Telepointer können in allen Fenster, die durch das Application Sharing angeboten werden, benutzt werden. Sie sind also nicht an Fenster- oder Applikationsgrenzen gebunden, dadurch ist eine sehr natürliche und intuitive Verwendung der Telepointer möglich.

- **Bewertung:**

CIO JVTOS ist gut geeignet für Desktop Konferenzen, bei genauerer Prüfung sind folgende Punkte besonders zu erwähnen:

Für JVTOS sind zur Zeit keine API zur Verfügung gestellt.

Der Konferenzaufbau ist zur Zeit nur über das User-Interface möglich.

Die Dienstkonfiguration ist nur eingeschränkt möglich.

JVTOS 3.0 ist auf verschiedenen Unix und Windows Systemplattformen verfügbar.

### **Schwachstellenanalyse**

Die untersuchten Konferenzsysteme/Groupwareumgebungen, Berkom MMC, Insoft Communique und CIO-JVTOS erfüllen im wesentlichen die Anforderungen die an solche Werkzeuge gestellt werden:

- Konferenzverwaltung
- Video und Audio zwischen den Konferenzteilnehmern zu übertragen unter Berücksichtigung der speziellen Anforderungen von kontinuierlichen Medienströmen.
- Application Sharing, um Dokumente und weitergehende Informationen zum Konferenzthema allen Teilnehmern anbieten zu können.

Als monolithische Einzelanwendung betrachtet sind die Werkzeuge voll funktionsfähig und effizient einsetzbar. Für den Einsatz bzw. die Integration in Vorgangssteuerungssysteme fehlen den Werkzeugen noch einige wichtige Eigenschaften und Merkmale:

- Die verwendeten Kommunikationsprotokolle müssen sich an Standards halten bzw. sind zu standardisieren, damit auch zwischen Desktop Konferenzsystemen verschiedener Hersteller audiovisuelle Konferenzen stattfinden können. Dies ist besonders in weit verteilten und sehr heterogenen Systemumgebungen notwendig, da dort meist der Einsatz von funktionell gleichen Produkten verschiedener Hersteller unumgänglich ist. Für die Übertragung von Video gibt es durch H.261 bereits einen Standard der von vielen Produkten verwendet wird.
- Die meisten Konferenzsysteme verwenden Kommunikationsprotokolle die keine konstante Medienqualität garantieren. Die Qualität ist daher stark von dem zugrundeliegendem Netzwerk und dessen Eigenschaften und Kapazitäten abhängig. Für eine befriedigende Lösung muß entweder das Netzwerk so stark überdimensioniert sein, daß es kaum zu Engpässen kommt, dies

ist aber keine realistische Möglichkeit. Die Verwendung von Kommunikationsprotokollen und Netzwerken bei denen Ressourcenreservierung durchgeführt werden kann ist wohl die bessere Alternative.

- Zur nahtlosen Integration der Kommunikationswerkzeuge müssen API's zur Steuerung verfügbar sein. Bei einer guten Integration können durch das Vorgangssteuerungssystem viele Informationen zum Konferenzaufbau und Informationsaustausch beigesteuert werden.
- Neben der Steuerungsmöglichkeit über API's müssen zusätzliche Systemdienste für die Integration mit Vorgangssteuerungssystemen verfügbar sein. Dazu gehört unter anderem die Möglichkeit die Anwesenheit und den Arbeitsplatz von Teilnehmern zu ermitteln und ein Terminkalender für die rechnerunterstützte Koordination von Konferenzen.
- Die ungenügende Unterstützung von Sicherheitsanforderungen ist eine weitere Schwachstelle der untersuchten Kommunikationssysteme. Beim Berkom MMC und bei CIO JVTOS sind bereits Sicherheitsmechanismen ansatzweise realisiert, dazu gehört vor allem die Unterstützung von Rollen (Konferenzleiter, aktiver Teilnehmer, Beobachter usw.) und beim Application Sharing die Möglichkeit zur gezielten Vergabe der Eingabeberechtigung (floor control).
- Es fehlt die Möglichkeit zur Übernahme bzw. Einhaltung eines externen Organisationsschemas (Vorgangssteuerungssystem) mit Vertraulichkeitsabstufungen. Dies bezieht sich sowohl auf die kontinuierlichen Medien (Video und Audio), als auch auf Dokumente, die über das Application Sharing angeboten werden. Beim Application Sharing kann ein „Copy and Paste“ von Dokumenten nicht unterbunden werden und somit kann keine sichere Kopien/Versionenkontrolle möglich.

## 8.3 Zusammenfassung

In dem Arbeitsbereich „Technologie Evaluierung“ wurden zwei verschiedene Technologiebereiche untersucht. Es wurde der Bereich der verteilten Systemdienste untersucht, der für ein inhärent verteiltes Workflow Management System besondere Bedeutung hat. Weiterhin wurde untersucht wie sich neue Medientypen in ein Workflow Management Systeme integrieren lassen und welche technologischen Defizite einer Integration heutzutage noch entgegenstehen.

Die Untersuchung der Multimedia Technologien fokussierte auf zwei Schwerpunkte: Verbunddokumente und Konferenzsysteme. Es hat sich gezeigt, daß die Basisfunktionalität in beiden Bereichen bereits vorhanden ist. Es wurden allerdings bei der Integration dieser Technologien in Workflow Management Systeme Schwachstellen aufgedeckt, die die Verbindung der Technologien noch behindern.

Bei der Untersuchung der verteilten Systemplattformen hat sich gezeigt, daß das “Distributed Computing Environment (DCE)” nur beschränkt geeignet ist, da es aufgrund der Fixierung auf die Programmiersprache C zu Problemen bei der Integration von Anwendungen und Altsoftware kommen kann. Außerdem ist DCE nicht geeignet, um in Intranetzen bzw. dem Internet eingesetzt zu werden. CORBA ist für die Realisierung gut geeignet. CORBA ist mittlerweile ein ausgereifter Standard und durch die Unabhängigkeit sowohl von der Betriebssystemplattform wie auch der Programmiersprache ist die Integration von Anwendungen und Altsoftware gegeben. Als wesentlicher Schwachpunkt bei der Evaluierung von Middleware-Systemen wurde das Fehlen von geeigneten Sicherheitsdiensten und -mechanismen zur Realisierung einer sicheren Vorgangsteuerung in weit verteilten Systemen identifiziert. Aufgrund dieses Defizits wurde in der zweiten Phase des Projekts PoliFlow die Thematik in dem Gebiet der “Sicheren Workflow Systeme” vertieft (vgl. auch Kapitel 5).

## 9 Zusammenfassung und Bewertung

Das Verbundprojekt PoliFlow (1.5.1994-31.12.1998) war ein Forschungsprojekt der Förderinitiative POLIKOM, die vom BMBF gefördert wurde. Zielsetzung von PoliFlow war die effektive Unterstützung von weitverteilten Vorgängen in der öffentlichen Verwaltung. Hierfür wurden Lösungen entwickelt, bei der verschiedene innovative Technologien wie Workflow-Management, DMS, Groupware- und synchrone Telekooperationsdienste über Intra- bzw. Internet integriert wurden. Projektpartner des IPVR in PoliFlow waren Hewlett-Packard GmbH, das IAT der Universität Stuttgart sowie die Oberfinanzdirektion Berlin als Anwender.

Die Aufgaben des IPVR innerhalb des Projektes lagen in der wissenschaftlichen und technologischen Begleitung des Projektes, mit dem Themenschwerpunkt Workflow-Management. Das Projekt wurde von den Abteilungen Anwendersoftware und Verteilte Systeme getragen, die Verwaltung lag bei der Abteilung Infrastruktur. Über den gesamten Zeitraum arbeiteten 4-5 wissenschaftliche Mitarbeiter/innen an mehreren Arbeitsbereichen und wurden dabei von zahlreichen wissenschaftlichen Hilfskräften sowie durch Studien- und Diplomarbeiten unterstützt.

### **Motivation und Problemstellungen**

Zum Zeitpunkt der Antragsstellung war Workflow-Management noch eine relativ neue Technologie, deren Potential jedoch bereits erkannt wurde. Neben den Chancen, die diese Technologie für die öffentliche Verwaltung eröffnete, mußten jedoch auch einige Schwächen bestehender Systeme zur Kenntnis genommen werden, die einem sofortigen Einsatz, in gewünschtem Umfang noch im Wege standen.

Das bestehende Anwendungsfeld stellt besondere Anforderungen u.a. an die zuverlässige Ausführung langlebiger Prozesse, an die flexible Unterstützung unstrukturierter oder variabler Prozesse sowie an die Integration synchroner Telekooperationsdienste in die Vorgangssteuerung. Neben der Evaluierung bestehender Technologien und der detaillierten Analyse des Anwendungsfeldes wurden daher bereits in der ersten Phase Lösungsansätze für diese Problemstellungen erarbeitet.

Weiterhin zeigte die Analyse, daß bei unternehmensweiten und -übergreifenden Workflows besondere Anforderungen zur Gewährleistung der Sicherheit und zum Betrieb in heterogenen Systemumgebungen berücksichtigt werden müssen. Um diese beiden Aspekte mitzubehandeln, wurden die Arbeiten in den Bereichen Zuverlässigkeit und Technologieevaluierung in der zweiten Projektphase ersetzt.



### **SWATS - Innovatives Workflow Management im Internet**

Für eine organisationsweite Einführung müssen die Systeme standortübergreifend verfügbar sein, und alle Anwender müssen einen einfachen und benutzerfreundlichen Zugang haben.

Das 'Stuttgarter Workflow- und Telekooperations-System' (SWATS) basiert auf der Workflow-Engine des Produktes 'Changengine' (Hewlett-Packard). Zusammen mit den am IPVR entwickelten Komponenten wurde ein innovatives WFMS realisiert, das in PoliFlow als Basistechnologie eingesetzt wurde.

Die Architektur von SWATS basiert auf der Internet-Technologie, um auf heterogenen Teilnetzwerken Intra- und Extranetze aufbauen zu können. Java-Applets überwinden die Heterogenität auf Client-Seite und gestatten eine nahtlose Integration von Informationen und Applikationen. Sie werden zur Realisierung der Benutzeroberfläche und der Aktivitäten des Workflow-Systems eingesetzt, die damit nicht vorinstalliert werden müssen. Die Middleware CORBA überbrückt Server-seitige Heterogenität und erlaubt die verteilte und interoperable Komposition und Steuerung eines offenen und erweiterbaren Workflow-Systems.

### **Forschungsschwerpunkte**

Im Bereich 'Flexibilität und Anpassungsfähigkeit von WFMS' wurden Konzepte auf mehreren Ebenen entworfen. Zum einen wurde durch die Definition beliebiger, komplexer Kontrollstrukturen die Flexibilität der Modellierung erhöht, zum anderen können einzelne Workflows aber auch noch während der Ausführung beliebig angepaßt werden. Durch eine entsprechende Modellierung können die einzelnen Workflows über Anpassungsrechte und Constraints geschützt werden, wodurch die Konsistenz und Integrität der Workflows garantiert wird. Über einen graphischen Editor werden die Benutzer bei der Durchführung von Anpassungen zusätzlich unterstützt.

Im Bereich 'Sicherheit in WFMS' wurde ein Konzept entwickelt, mit dem auch in organisationsübergreifenden Workflows die individuellen Sicherheitspolitiken und -mechanismen der einzelnen Organisationen gewährleistet werden können. Durch ein dynamisches, Aktivitäten-basiertes Sicherheitsmodell wird sichere und domänenübergreifende Vorgangsbearbeitung praktikabel. Sicherheitspolitiken müssen nicht mehr statisch-administrativ realisiert werden, sondern werden speziell im Kontext der Vorgänge definiert und zur Instanzlaufzeit in den jeweiligen Sicherheitsdomänen umgesetzt.

Beide Lösungen wurden weitestgehend unabhängig von spezifischen Modellen oder Architekturen konzipiert, wodurch eine hohe Übertragbarkeit erreicht wurde.

### **Projektmanagement und Öffentlichkeitsarbeit**

Durch den großen Verbund von Projektpartnern innerhalb und außerhalb der Universität stellte das Projekt besondere Ansprüche an die Koordination der Arbeitsbereiche. Durch entsprechendes Management ist es gelungen die gesteckten Ziele zu erreichen und den Rahmenplan zu erfüllen.

Im Rahmen des Projektes wurden über 20 Studien- und Diplomarbeiten bearbeitet, über 20 weitere Studenten/innen beteiligten sich am Projekt als wissenschaftliche Hilfskräfte. In den einzelnen Forschungsbereichen wurde Kontakte zu verschiedenen Organisationen (WFMC, OMG, GI-AK-Workflow) gepflegt, und neben Kooperationen mit Industriepartnern (HP-Labs, Palo Alto) entstand ein erfolgreicher wissenschaftlicher Austausch mit weiteren Universitäten (Münster, Ulm, Erlangen).

Durch die aktuelle Thematik und Technologie sowie die große politische Bedeutung des Projektes wurde eine entsprechende Öffentlichkeitsarbeit erforderlich. Neben Veröffentlichungen in der Presse wurden die wissenschaftlichen Ergebnisse auf zahlreichen nationalen und internationalen Konferenzen präsentiert und publiziert. Die praktische Relevanz der Ergebnisse wurde bei mehr als zehn Systempräsentationen bestätigt (u.a. CeBIT '96 und '98, POLIKOM-Konferenzen). Insgesamt fanden alle durchgeführten Maßnahmen ein breites Interesse und große Bestätigung bei Anwendern, Herstellern und Wissenschaftlern.