

**Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik**

Einsatz von Workflow-Management-Systemen zur
Unterstützung unternehmensübergreifender
Kooperationsformen

Diplomarbeit

Leipzig, August 1998

vorgelegt von
Drawehn, Jens

1	Einleitung	5
2	Computergestützte Gruppenarbeit	6
2.1	Begriffe	6
2.2	Klassifikationen von Groupware	8
2.3	Die Groupware-Kategorie der Workflow-Management-Systeme	11
3	Das Forschungsfeld Workflow-Management	13
3.1	Begriffe	13
3.1.1	Grundlegende Begriffe	13
3.1.2	Prozesse und ihre Struktur	15
3.1.3	Anwendungen und Daten	17
3.2	Entwicklung von Workflow-Management-Systemen	18
3.3	Aufgaben und Ziele von Workflow-Management	19
3.4	Einsatz von Workflow-Management-Systemen	19
4	Workflow-Management-Systeme	21
4.1	Anforderungen an Workflow-Management-Systeme	21
4.1.1	Aspekte von Workflow-Management-Anwendungen	21
4.1.2	Benutzerschnittstellen von Workflow-Management-Systemen	22
4.1.3	Nichtfunktionale Anforderungen	23
4.2	Architektur von Workflow-Management-Systemen	26
5	Standardisierung: Die Workflow Management Coalition	28
5.1	Die Workflow Management Coalition	28
5.2	Grundlegende Funktionen eines Workflow-Management-Systems	29
5.3	Generische Struktur von Workflow-Management-Systemen	30
5.4	Das Referenzmodell der WfMC	31
5.5	WAPI – Die Schnittstellen des Referenzmodells	32
5.5.1	Interface 1: Prozeßdefinitionen	32
5.5.2	Interface 2: Client-Anwendungen	33
5.5.3	Interface 3: Invoked Applications	33
5.5.4	Interface 4: Interoperabilität	34
5.5.5	Interface 5: Administration und Überwachung	34
5.6	Interoperabilität	34
5.6.1	Interoperabilitätsszenarien von Workflow-Management-Systemen	35
5.6.2	Stufen der Interoperabilität von Workflow-Management-Systemen	38
5.7	Stand der Arbeit der WfMC	39
6	Gegenwärtiger Stand der Entwicklung	41
6.1	Beispiele für Workflow-Management-Systeme	41
6.1.1	Oracle InterOffice 4.1	41
6.1.2	HP AdminFlow	44
6.1.3	IBM FlowMark	46
6.1.4	Weitere Workflowsysteme	47

6.2	Workflow-Interoperabilität der vorgestellten Produkte	47
6.3	Klassifikationsansätze von Workflow-Management-Systemen	48
6.3.1	Klassifikation anhand des Workflow-Metaschema-Modells	48
6.3.2	Klassifikation bezüglich der Art der Datenbankverwendung	50
6.3.3	Einordnung nach Art der Vorgabebestimmung und Bezugsobjekt	51
6.4	Fazit	52
7	Interoperabilität von Workflow-Management-Systemen	53
7.1	Interoperabilitätsstrategien von Software	53
7.2	Eine Typologie zwischenbetrieblicher Anwendungen	54
7.3	Nutzung von Internet-Technologien	56
7.4	Workflow-Management und Middleware	57
7.4.1	Die Object Management Group	58
7.4.2	CORBA	58
7.4.3	Die Workflow Management Facility	64
7.4.4	Workflow-Management und die OMG	65
8	Workflow-Interoperabilität – ein Beispiel	67
8.1	Ein Beispielszenario: Das Projekt PoliFlow	67
8.1.1	Der Prozeß „Auftragsvergabe“	67
8.1.2	Implementation der benötigten Aktivitätstypen	68
8.2	Workflow-Interoperabilität mit InterOffice	69
8.2.1	Das IDL-Interface	70
8.2.2	Implementation des CORBA-Servers	71
8.2.3	Implementation des CORBA-Clients	72
8.3	Einordnung des Beispielprojekts	72
8.3.1	Unterstützte Interoperabilitätsszenarien	72
8.3.2	Unterstützte Interoperabilitätsstufen	74
8.3.3	Unterstützung zwischenbetrieblicher Anwendungen	75
8.3.4	Grenzen	76
9	Kurzzusammenfassung	77
	Literaturverzeichnis	78
	Anhang 1 IDL-Interface zur Prozeßkommunikation	81
	Anhang 2 Java-Beispielroutine des CORBA-Servers	86
	Anhang 3 Der Prozeß ‚Auftragsvergabe‘	88
	Anhang 4 Inhalt der CD-ROM	91
	Erklärung	92

Abbildungsverzeichnis

Abbildung 1: Die Interdisziplinarität von CSCW	6
Abbildung 2: Raum-Zeit-Matrix nach Johansen	9
Abbildung 3: Klassifikation von Groupware nach Unterstützungsfunktionen	10
Abbildung 4: Zustände von Prozeßinstanzen	17
Abbildung 5: Ausschnitt des Workflow-Metaschemas von InterOffice 4.1	22
Abbildung 6: Komponentenarchitektur eines Workflow-Management-Systems	26
Abbildung 7: Schichtenarchitektur eines Workflow-Management-Systems	27
Abbildung 8: Workflow System Characteristics	29
Abbildung 9: Generic Workflow Product Structure	30
Abbildung 10: Workflow Reference Model - Components & Interfaces	31
Abbildung 11: Chained Services Model	35
Abbildung 12: Nested Subprocesses Model	36
Abbildung 13: Peer-to-peer Model	37
Abbildung 14: Parallel Synchronised Model	37
Abbildung 15: Dynamische Formulare mit InterOffice	43
Abbildung 16: Komponenten von AdminFlow	45
Abbildung 17: Workflow-Management-Systeme mit Datenbanken	50
Abbildung 18: Interoperabilitätsstrategien von Software	54
Abbildung 19: Typologie zwischenbetrieblicher Anwendungen	55
Abbildung 20: Die Referenzarchitektur OMA	58
Abbildung 21: Interaktion zwischen Objekten mittels ORB	59
Abbildung 22: Workflow-Interoperabilität mit InterOffice	70
Abbildung 23: Oberfläche des CORBA-Clients	72

Tabellenverzeichnis

Tabelle 1: Unterschiede zwischen Workflow-Management-Systemen und anderer Groupware	12
Tabelle 2: Generationen von Workflow-Management-Systemen	18
Tabelle 3: Verfügbare Dokumente der WfMC	40
Tabelle 4: Vergleich von Workflow-Metaschema-Modellen	49
Tabelle 5: Klassifikation von Workflow-Management-Systemen	52
Tabelle 6: CORBA Services	61
Tabelle 7: CORBA Facilities	63
Tabelle 8: Domain Interface Task Forces	63
Tabelle 9: Routinen des IDL-Interfaces zur Prozeßsteuerung	71
Tabelle 10: Produktübergreifende Ausführung eines Subprozesses	74

1 Einleitung

Das Thema Workflow-Management hat in den letzten Jahren ein beständig steigendes Interesse aus Forschung und Anwendung erfahren. Weit über 100 Anbieter haben eine Vielzahl von Workflowprodukten auf den Markt gebracht, deren unterschiedliche Ansätze eine gemeinsame Betrachtung erschweren. Die vorliegende Arbeit will einen Überblick über den erreichten Stand auf dem Forschungsgebiet Workflow-Management geben und untersuchen, inwieweit sich die gewonnenen Erkenntnisse in den Produkten wiederfinden. Besondere Beachtung findet dabei die Frage, ob die Produkte neue Informationsinseln darstellen oder Interaktionen mit anderen Anwendungen, speziell anderen Workflow-Management-Systemen, möglich sind. Letztere Eigenschaft ist für den effizienten Einsatz der Systeme deshalb wichtig, weil sie nur dann auch für Vorgänge eingesetzt werden können, die Unternehmensgrenzen überschreiten.

Kapitel 2 beschreibt das Forschungsgebiet der computergestützten Gruppenarbeit und stellt die Besonderheiten von Workflow-Management-Systemen gegenüber anderer Groupware heraus. In Kapitel 3 wird das Forschungsfeld Workflow-Management vorgestellt. Hier sind auch die für die weitere Bearbeitung des Themas notwendigen Begriffe definiert. Kapitel 4 beschreibt die technologischen Anforderungen an Workflow-Management-Systeme und einen Architekturvorschlag, der diesen Anforderungen gerecht wird. In Kapitel 5 wird ausführlich auf die Arbeit der Workflow Management Coalition (WfMC) eingegangen. Das Referenzmodell der WfMC wird beschrieben, wobei entsprechend dem Thema der Arbeit die Interoperabilitätsschnittstelle genauer betrachtet wird. Am Schluß des Kapitels befindet sich eine Zusammenfassung des aktuellen Stands der Arbeit der WfMC. Kapitel 6 zeigt anhand von konkreten Produkten, was heutige Workflow-Management-Systeme können und wie sie die theoretischen Anforderungen erfüllen. Anschließend werden einige Klassifikationsansätze für Workflow-Management-Systeme vorgestellt. In Kapitel 7 wird beschrieben, welche Möglichkeiten heute für den Datenaustausch zwischen Softwareprodukten genutzt werden und welche Bedeutung der Middleware-Standard CORBA für Workflow-Management hat. Abschließend wird im Kapitel 8 an einem Beispielprojekt gezeigt, wie mit einem konkreten Produkt Interoperabilität erreicht werden kann.

2 Computergestützte Gruppenarbeit

In diesem Abschnitt soll zunächst einmal geklärt werden, was computergestützte Gruppenarbeit (engl. Computer Supported Cooperative Work – CSCW) ist. Dann werden verschiedene Klassifikationen von CSCW-Anwendungen, sogenannter Groupware, vorgestellt. Schließlich wird der Bereich des Workflow-Managements als Teilbereich des Forschungsgebietes CSCW herausgestellt.

2.1 Begriffe

Der Begriff **Computer Supported Cooperative Work (CSCW)** wurde von Irene Greif und Paul Cashman geprägt. Im Jahre 1984 organisierten sie ein Forschertreffen, die erste internationale Konferenz zum Thema CSCW fand 1986 in Austin statt. Mit dieser Konferenz wurden einem weiten Kreis von Wissenschaftlern die Dimensionen der neuen Forschungsdisziplin bewußt.¹

Das Forschungsgebiet CSCW befaßt sich mit der Rolle der Informations- und Kommunikationstechnologie in Arbeit und Gesellschaft und ist ein interdisziplinäres Forschungsgebiet, wie aus Abbildung 1 ersichtlich wird.

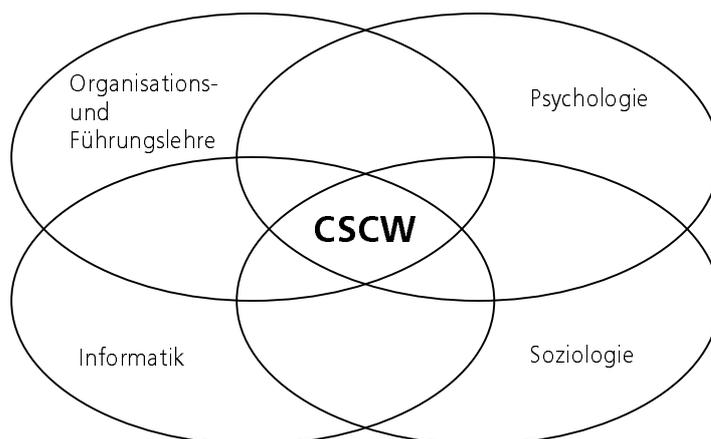


Abbildung 1: Die Interdisziplinarität von CSCW²

¹ [Teufel et al. 95], Seite 17

² [Teufel et al. 95], Seite 19

Die Informatik erforscht in diesem Zusammenhang die Zusammenarbeit von Mensch und Computer sowie die Unterstützung von Informationsbedürfnissen in der Arbeitswelt, sie ist mit den entsprechenden Forschungsbereichen Mensch-Maschine-Interaktion und Büroinformationssysteme beteiligt. Die Sozialpsychologie setzt sich mit Gruppenaspekten auseinander und untersucht die Auswirkungen von CSCW-Anwendungen auf den Menschen. Die Organisations- und Führungslehre deckt die betriebswirtschaftlichen Aspekte ab und stellt unter anderem Kosten/Nutzen-Überlegungen an. Das Zusammenwirken dieser Disziplinen aus den unterschiedlichen Forschungsgebieten ergibt das Forschungsfeld CSCW.

Die folgenden Begriffe dienen als Grundlage für die Definition von CSCW. Der Begriff der Gruppe führt zu dem der Gruppenarbeit, der wiederum für die Definition von CSCW, also computergestützter Gruppenarbeit, notwendig ist. Deshalb wird zunächst eine Definition des Begriffs **Gruppe** sowie deren besondere Formen Arbeitsgruppe und Team gegeben.

„Von einer Gruppe spricht man, wenn zwei oder mehrere Personen interagieren und dabei eine gegenseitige Beeinflussung stattfindet.“³

Dienen die Interaktionen zwischen den Personen dazu, eine bestimmte Aufgabe zu erfüllen, dann handelt es sich bei der Gruppe um eine **Arbeitsgruppe**. Verfolgen die Gruppenmitglieder neben der gemeinsamen Aufgabe auch ein gemeinsames Ziel, dann spricht man von einem **Team**.

Der Begriff der **Gruppenarbeit** kann jetzt wie folgt definiert werden:

„Gruppenarbeit ist die Summe aller aufgabenbezogenen Tätigkeiten, welche von Gruppenmitgliedern ausgeführt werden, um zielbezogene Aufgaben zu erfüllen und somit Gruppenziele zu erreichen.“⁴

Für die Durchführung von Gruppenarbeit sind **Gruppenprozesse** notwendig. Diese kann man den drei Kategorien der **Kommunikation, Koordination und Kooperation** zuordnen. Dabei sind Kommunikationsprozesse die Voraussetzung für Koordinationsprozesse und diese wiederum Voraussetzung für Kooperationsprozesse. Damit ist Kommunikation, also die Verständigung mehrerer Personen untereinander, die Grundlage jeder Gruppenarbeit. Bezieht sich die Kommunikation in einer Gruppe auf die Abstimmung aufgabenbezogener Tätigkeiten, spricht man von Koordination. Koordination,

³ [Teufel et al. 95], Seite 9

⁴ [Teufel et al. 95], Seite 11

die der Vereinbarung gemeinsamer Ziele dient, wird als Kooperation bezeichnet.⁵

Der Begriff der Gruppenarbeit ist nun gegeben und kann für eine Definition des Forschungsgebiets CSCW herangezogen werden.

„Computer Supported Cooperative Work (CSCW) ist die Bezeichnung des Forschungsgebietes, welches auf interdisziplinärer Basis untersucht, wie Individuen in Arbeitsgruppen oder Teams zusammenarbeiten und wie sie dabei durch Informations- und Kommunikationstechnologie unterstützt werden können.“⁶

Software, in der die Theorien des Forschungsgebietes CSCW implementiert sind, bezeichnet man als **Groupware**. Wichtig ist die Abgrenzung von Groupware gegenüber **Mehrbenutzer-Anwendungen**, mit denen mehrere Benutzer gleichzeitig, aber voneinander unabhängig arbeiten können.⁷

2.2 Klassifikationen von Groupware

Es gibt eine Vielzahl von Kriterien zur Klassifikation von Groupware. Einige wichtige Kriterien mit den zugehörigen Klassifikationsschemata werden hier vorgestellt. Ein einheitliches und umfassendes Schema ist deshalb schwierig, weil die meisten Anwendungen verschiedene Unterstützungsfunktionen bieten und dadurch auch verschiedene Arbeitsformen ermöglichen.

Dem vorigen Abschnitt kann man entnehmen, daß Kommunikationsprozesse die Grundlage für Gruppenarbeit und damit für Groupware sind. Die Charakteristika dieser Prozesse können als Kriterien für eine Groupware-Klassifikation aus technologischer Sicht dienen. Wählt man örtliche und zeitliche Verteilung der Kommunikationsprozesse aus, erhält man eine Raum-Zeit-Matrix, wie sie in Abbildung 2 zu sehen ist. Weitere mögliche Kriterien sind beispielsweise die verwendeten Medientypen, die Anzahl der Kommunikationspartner oder ob die Kommunikation explizit oder implizit erfolgt.⁸

⁵ [Teufel et al. 95], Seite 11 f

⁶ [Teufel et al. 95], Seite 17

⁷ [Teufel et al. 95], Seite 22

⁸ [Teufel et al. 95], Seite 23 f

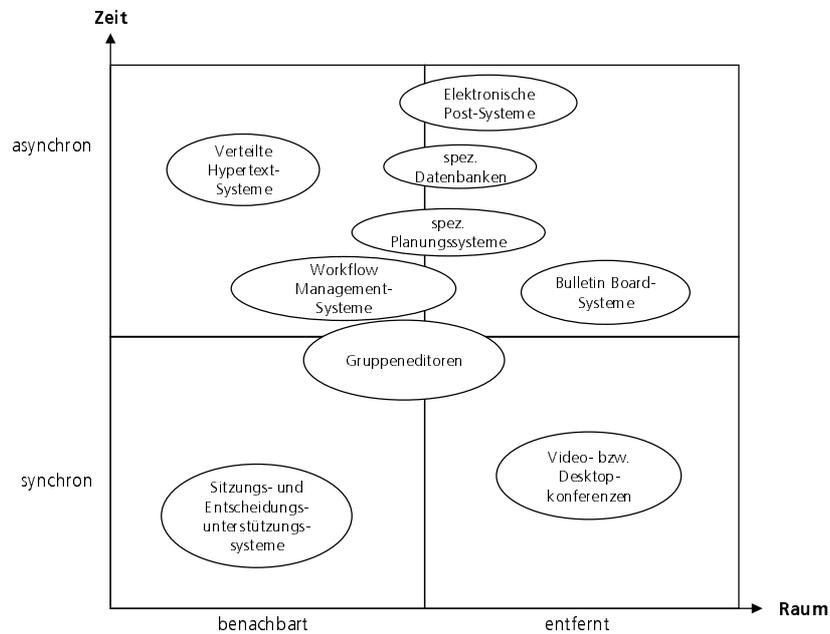


Abbildung 2: Raum-Zeit-Matrix nach Johansen⁹

Die wenigsten Applikationen lassen sich eindeutig einer der vier Kategorien der Matrix zuordnen, weil sie mehrere Kommunikationstechniken unterstützen. Das Schema läßt sich auf verschiedene Art erweitern. Man kann die Koordination als weitere Dimension hinzufügen. Denkbar ist weiterhin, die beiden Achsen im mittleren Bereich jeweils durch die Kategorien „unterschiedlich, aber bekannt“ zu verfeinern.

Ein anderer Ansatz klassifiziert Groupware nach der Art der angebotenen Unterstützungsfunktionen für Gruppenprozesse. Aus den Definitionen dieser Prozesse ergibt sich, daß Groupware immer Kommunikationsunterstützung bietet. Die einzelnen Anwendungen legen aber unterschiedliche Schwerpunkte bezüglich der Unterstützungsfunktionen. Dementsprechend lassen sie sich, wie in Abbildung 3 gezeigt, in einem Dreieck anordnen.

⁹ [Teufel et al. 95], Seite 25

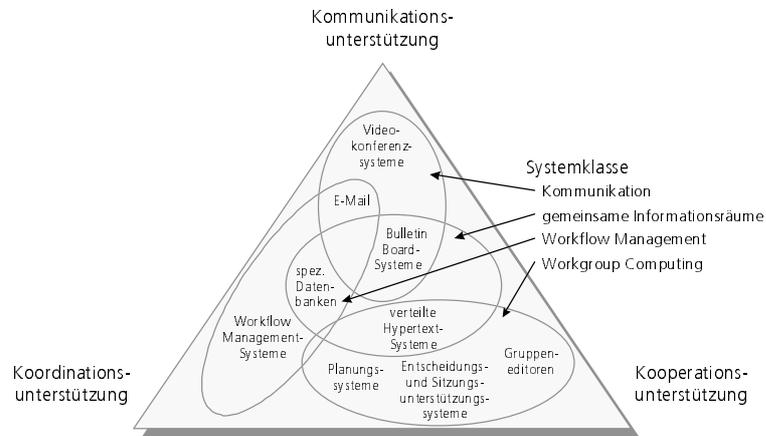


Abbildung 3: Klassifikation von Groupware nach Unterstützungsfunktionen¹⁰

In diesem Schema lassen sich vier einander überlappende Systemklassen bilden.

In die **Systemklasse Kommunikation** fallen Anwendungen, die den expliziten Informationsaustausch ermöglichen, zum Beispiel Videokonferenz- und E-Mail-Systeme.

Die **Systemklasse gemeinsame Informationsräume** umfaßt solche Systeme, die die implizite Kommunikation unterstützen, wie zum Beispiel verteilte Hypertextsysteme.

In der **Systemklasse Workgroup Computing** sind Anwendungen zusammengefaßt, die die Kooperation in Arbeitsgruppen und Teams unterstützen. In den Gruppen treten dabei vor allem Aufgaben mit mittleren bzw. geringen Strukturierungsgraden und Wiederholungsfrequenzen auf. Typische Vertreter sind Gruppeneeditoren und Sitzungsunterstützungssysteme.

Die **Systemklasse Workflow-Management** legt den Schwerpunkt auf die Koordination von Aufgaben. Dazu werden unter anderem Techniken eingesetzt, die aus dem Bereich der E-Mail-Systeme und spezieller Datenbanksysteme stammen.

¹⁰ [Teufel et al. 95], Seite 27

2.3 Die Groupware-Kategorie der Workflow-Management-Systeme

Im vorigen Abschnitt wurde bereits eine Abgrenzung von Workflow-Management-Systemen zu anderer, kommunikativ/kooperativer Groupware vorgenommen, die jetzt vervollständigt wird.

Workflow-Management-Systeme sind prozeßorientiert. Das unterscheidet sie von anderer Groupware, die z.B. datenorientiert oder funktionsorientiert sein kann.

Grundsätzlich gibt es viele Bereiche, in denen man sich den erfolgreichen Einsatz von Workflow-Management-Systemen vorstellen kann. Besonders geeignet sind sie zur Unterstützung von Vorgängen, die mindestens eine der folgenden Eigenschaften aufweisen.¹¹

- Eine hohe Anzahl von Personen oder Anwendungen ist in den Vorgang einbezogen.
- Der Vorgang weist einen hohen Strukturierungsgrad auf.
- Der Vorgang hat eine hohe Wiederholungsrate und erfordert wenige Ausnahmebehandlungsmechanismen.

Workflow-Management betrachtet die kausalen und zeitlichen Abhängigkeiten von Teilschritten, die zusammen einen Beitrag zum Unternehmensziel leisten. Workflow-Management-Systeme befassen sich also mit der Steuerung, Verwaltung und Koordination von Vorgängen in Unternehmen. Sie unterstützen die Koordination aktiv, andere Groupware bietet lediglich Informationen, die den Benutzer bei seinen Koordinationstätigkeiten unterstützen.

Während zu Workflow-Management-Systemen immer ein Vorgangsschema gehört, ist das bei anderen Systemen aufgrund der Unterstützung schlecht formalisierbarer Vorgänge oft nicht der Fall. Dennoch bieten einige dieser Systeme sogenannte Templates an, die dann vom Benutzer relativ einfach zur Laufzeit geändert werden können. Workflow-Management-Systeme unterstützen vorrangig asynchrone Kommunikationsformen. Synchrone Zugriffe auf dieselben Ressourcen werden meist durch andere Systeme unterstützt. Tabelle 1 faßt wesentliche Unterschiede zwischen Workflow-Management-Systemen und kommunikativ/kooperativer Groupware zusammen.

¹¹ [Teufel et al. 95], Seite 183

	Workflow-Management-Systeme	kommunikativ/ kooperative Groupware
Koordination	Aktivität	Objekt
Koordinationsunterstützung	aktiv	passiv
Schemakonzept	ja	bedingt
Schemakonzept zur Laufzeit änderbar	nur teilweise	teilweise
Synchronizität	asynchron	synchron
Prozeßverlauf	bekannt	unbekannt

Tabelle 1: Unterschiede zwischen Workflow-Management-Systemen und anderer Groupware¹²

¹² [Jablonski et al. 97], Seite 381

3 Das Forschungsfeld Workflow-Management

Das sich aus dem Forschungsgebiet CSCW heraus entwickelnde Forschungsfeld Workflow-Management ist noch sehr jung. Die Workflow Management Coalition, deren Ziel die Standardisierung in diesem Bereich ist, wurde 1993 gegründet.

In diesem Abschnitt werden die wesentlichen Konzepte auf dem Gebiet des Workflow-Managements erläutert. Dabei sind die Begriffe weitgehend von der Workflow Management Coalition (WfMC) übernommen, auf deren Arbeit in Abschnitt 5 näher eingegangen wird. Weiterhin werden die Entwicklung von Workflow-Management-Systemen, Aufgaben und Ziele von Workflow-Management sowie Fragen des Einsatzes der Systeme betrachtet.

3.1 Begriffe

3.1.1 Grundlegende Begriffe

Ein **Arbeitsvorgang** oder auch **Vorgang** ist ein Geschehen in der Realität, mit dem bestimmte Ziele verfolgt werden. Ein **Geschäftsprozeß** ist ein Vorgang in einer Organisation, der direkt zu einem Unternehmensziel beiträgt. Ein Geschäftsprozeß setzt sich meist aus mehreren Vorgängen zusammen, die keinen direkten Beitrag zum Unternehmensziel leisten müssen.

Ein **Prozeß** im Sinne des Workflow-Managements ist ein Vorgang, zu dem es eine formale Beschreibung gibt, die eine automatisierte Bearbeitung ermöglicht. Diese Beschreibung wird als **Prozeßdefinition** bezeichnet. Zur Erstellung von Prozeßdefinitionen benötigt man eine formale Sprache mit entsprechenden Ausdrucksmitteln, eine **Workflow-Sprache**¹³, deren Produkte die Prozeßdefinitionen sind. Es gibt verschiedene Workflow-Sprachen, die sich hinsichtlich der Mächtigkeit ihrer Ausdrucksmittel unterscheiden. Dabei gruppieren sie sich in der Regel nicht in ausdruckschwache und reichhaltige Sprachen, sie sind vielmehr bezüglich bestimmter Aufgaben geeignet oder ungeeignet. Da Workflow-Sprachen formale Sprachen sind, kann man zu ihnen ein Metaschema angeben. Dieses **Workflow-Metaschema** beschreibt die Eigenschaften aller Prozeßdefinitionen der Workflow-Sprache. Es wird durch

¹³[Jablonski et al. 97], Seite 492

die Definition einer Ablaufsemantik vervollständigt, wodurch ein **Workflow-Metaschema-Modell** gebildet wird.¹⁴

Als **Aktivität** bezeichnet man einen logischen Schritt innerhalb eines Vorgangs. Aktivitäten sind Instanzen eines bestimmten **Aktivitätstyps**, der die statischen Eigenschaften der Aktivitäten beschreibt. Man unterscheidet zwischen manuellen und automatisierten Aktivitäten. Manuelle Aktivitäten können Teil von Geschäftsprozessen sein. Ihre Ausführung ist aber nicht automatisiert. Automatisierte Aktivitäten werden auch als **Workflow-Aktivitäten** bezeichnet. Sie erfordern menschliche und/oder maschinelle Ressourcen.

Die WfMC definiert den Begriff **Workflow** wie folgt:

“The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.”¹⁵

Ein Workflow ist also der zumindest teilweise automatisierte Ablauf von Aktivitäten, die Bestandteil von Geschäftsprozessen sind. Workflows sind dabei überwiegend als ergonomische und weniger als technische Prozesse zu verstehen, da die Ausführung der Aktivitäten in der Regel eine Interaktion von Mensch und Maschine erfordert.¹⁶

Das Forschungsfeld **Workflow-Management** befaßt sich mit der vollständigen oder teilweisen Automatisierung von Arbeitsvorgängen. In diesem Zusammenhang umfaßt es Handlungen wie das Organisieren, Planen, Entscheiden, Kontrollieren, Steuern und Führen.¹⁷ Software, die zur Ausführung dieser Aufgaben dient, wird von der WfMC als **Workflow-Management-System** bezeichnet und wie folgt definiert:

“A system that defines, creates and manages the execution of workflows through the use of software, ...”¹⁸

Unter einer **Instanz** versteht man eine konkrete Ausführung eines Prozesses oder einer Aktivität durch ein Workflow-Management-System. Man unterscheidet demzufolge **Prozeßinstanzen** und **Aktivitätsinstanzen**. Eine Instanz wird vom Workflow-Management-System erzeugt und verwaltet, wenn der zugehörige Prozeß bzw. die zugehörige Aktivität ausgeführt wird.

¹⁴ [Jablonski et al. 97], Seite 124

¹⁵ [WfMC Glossary], Seite 7

¹⁶ [Jablonski et al. 97], Seite 490

¹⁷ [Jablonski et al. 97], Seite 491

¹⁸ [WfMC Reference Model], Seite 7

Aus der Sicht eines Prozesses unterscheidet man die **Build Time**, in der die Prozeßdefinition festgelegt wird und die **Run Time**, in der die Prozeßinstanzen erzeugt und verwaltet werden.

Eine Aktivitätsinstanz verlangt die Durchführung einer bestimmten Arbeit, was entweder durch menschliche Benutzer, sogenannte **Workflowteilnehmer**, oder durch maschinelle Ressourcen geschieht. Jeder Workflowteilnehmer hat eine **Arbeitsliste**, in der die für ihn relevanten Tätigkeiten aufgeführt sind.

In der Regel werden die Aktivitäten nicht direkt den Workflowteilnehmern zugeordnet. Jeder Teilnehmer bekommt eine oder mehrere **Rollen** zugewiesen. Diese Rollen werden den Aktivitäten zugeordnet.

Das Verfolgen und Aufzeichnen von Ereignissen im Workflow-Management-System wird als **Workflow-Monitoring** bezeichnet.

Eine **Workflow Engine** bietet eine Ausführungsumgebung für Prozeßinstanzen. Sie ermöglicht deren Ausführung basierend auf den zugehörigen Prozeßdefinitionen. Die Fähigkeit von zwei oder mehr Workflow Engines zur Kommunikation und Zusammenarbeit bezeichnet man als **Workflow-Interoperabilität**.

Ein Service, der eine oder mehrere Workflow Engines beinhaltet und die Erzeugung, Verwaltung und Ausführung von Instanzen ermöglicht, heißt **Workflow Enactment Service**. Workflow-Interoperabilität ist sowohl innerhalb eines Workflow Enactment Service möglich als auch zwischen Workflow Engines, die sich in verschiedenen Workflow Enactment Services eines Workflow-Management-Systems befinden.

Während der Ausführung von Workflows durch ein Workflow-Management-System können verschiedenartige Fehler auftreten. Diese Fehler lassen sich in **Systemfehler** und **semantische Fehler** unterteilen. Systemfehler werden durch die dem System zugrundeliegende Technologie verursacht. Semantische Fehler entstehen durch Ausnahmesituationen, die nicht auf technologisches Versagen zurückzuführen sind und eine normale Fortsetzung des Prozesses verhindern.

3.1.2 Prozesse und ihre Struktur

Dem vorigen Abschnitt kann man entnehmen, daß ein Prozeß ein formal beschriebener Vorgang ist, der sich aus Aktivitäten zusammensetzt. Es gibt verschiedene Formalisierungsansätze, die als Grundlage für die

Prozeßmodellierung dienen können, z.B. Petrinetze¹⁹, State/Activity-Charts²⁰ und Prozeßalgebren²¹. Sie stehen aber nicht im Zentrum dieser Arbeit, da deren Schwerpunkt eher auf den Aspekten der Softwarearchitektur von Workflow-Management-Systemen liegt.

Die Strukturierung von Prozessen wird durch das Konzept der Subprozesse unterstützt. Ein Prozeß, der innerhalb eines anderen Prozesses ausgeführt wird, heißt **Subprozeß**. An der Spitze einer so gebildeten Prozeß-Hierarchie steht ein **Top-Level-Prozeß**. Man unterscheidet zusammengesetzte und atomare Prozesse. Ein **atomarer Prozeß** ist nicht weiter zerlegbar, enthält also nur einen logischen Schritt und ist demzufolge einer Aktivität gleichzusetzen.

Aktivitätsinstanzen können entweder gleichzeitig oder nacheinander ausgeführt werden. Man bezeichnet diese beiden Formen der Abarbeitung als **sequentielles** bzw. **paralleles Routing**.

Innerhalb von Workflows gibt es Verzweigungspunkte. Wenn der Kontrollfluß auf mehrere Aktivitätsinstanzen aufgeteilt wird, die alle gleichzeitig ausgeführt werden, dann handelt es sich um einen **AND-Split**. Der Punkt, an dem dieser Kontrollfluß wieder zusammengeführt wird, heißt **AND-Join**. Bei einer Verzweigung, von der jeweils nur ein Pfad ausgeführt wird und damit den Kontrollfluß übernimmt, spricht man von einem **OR-Split**. Die Zusammenführung der Alternativen heißt **OR-Join**. Wenn zwischen Punkten, die sich in verschiedenen Pfaden befinden, Informationen ausgetauscht werden, dann spricht man von **Synchronisationspunkten**.

Unter einer **Iteration** versteht man einen Aktivitätszyklus, der eine oder mehrere Workflow-Aktivitäten wiederholt ausführt. Zu einer Iteration gehört im Allgemeinen eine Abbruchbedingung. Die Aktivitäten werden so lange wiederholt, bis die Bedingung erfüllt ist.

Während ihrer Abarbeitung kann eine Instanz verschiedene Zustände einnehmen. Eine Übersicht über diese Zustände und die zugehörigen Übergänge bietet Abbildung 4. Die Zustände *Running* und *Active* unterscheiden sich dadurch, daß im ersten Fall bestimmte Aktivitäten gestartet werden können und im zweiten Fall mindestens eine Aktivität tatsächlich gestartet wurde.

¹⁹ [Baumgarten 96]; [Oberweis 96]

²⁰ [Wodtke 97]

²¹ [Baeten/Weijland 90]; [Milner 89]; [Hennessy 88]

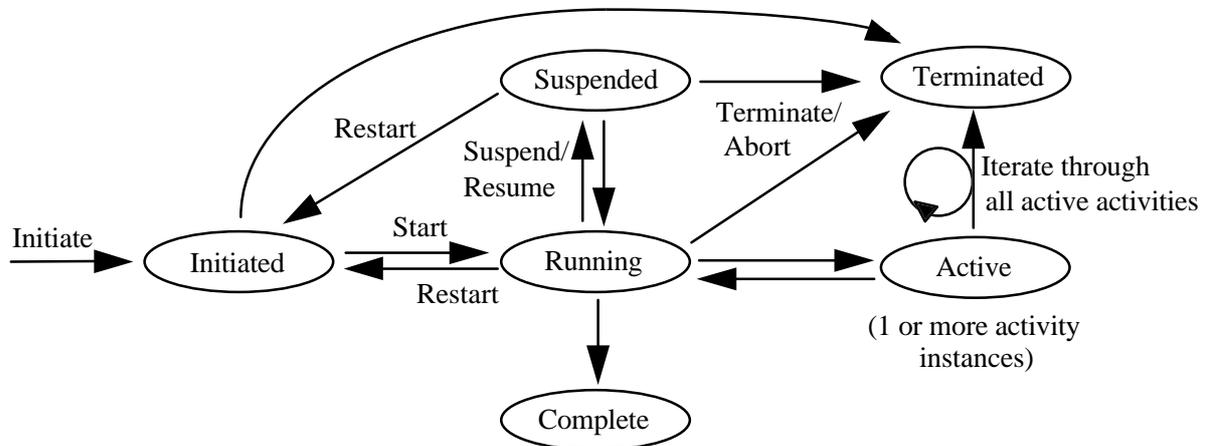


Abbildung 4: Zustände von Prozeßinstanzen²²

3.1.3 Anwendungen und Daten

Eine **Workflow-Applikation** oder Workflow-Anwendung ist eine Software, die mit einem Workflow-Management-Systemen interagiert und damit die Abarbeitung von Aktivitäten unterstützt. Workflow-Anwendungen übernehmen keine Steuerungsaufgaben, sie stellen lediglich Funktionen bereit, die vom Workflow-Management-System zur Erledigung bestimmter Aufgaben benötigt werden.²³

Eine **Client-Applikation** ist eine Software, die mit einem Workflow-Management-System interagiert, indem sie verschiedene Dienste in Anspruch nimmt. Dabei kann es sich um die Handhabung der Arbeitsliste, die Erzeugung von Prozeßinstanzen und anderes handeln.

Eine **Workflow-Management-Anwendung** ist eine implementierte und eingeführte Lösung zur Steuerung von Workflows mit einem Workflow-Management-System. Sie umfaßt die Komponenten Workflow-Management-System, Prozeßdefinitionen, Instanzdaten, eingesetzte Akteure und Workflow-Applikationen sowie Benutzerschnittstellen für Anwender und Betreiber des Systems.²⁴

Anwendungsdaten gehören zu einer Workflow-Anwendung und stehen dem Workflow-Management-System nicht zur Verfügung. **Workflowrelevante Daten** werden vom Workflow-Management-System dazu genutzt, den Status

²² [WfMC Reference Model]

²³ [Jablonski et al. 97], Seite 490

²⁴ [Jablonski et al. 97], Seite 491

von Instanzen zu bestimmen. **Workflowkontrolldaten** werden innerhalb des Workflow-Management-Systems verwaltet und sind für die Anwendungen normalerweise nicht zugänglich. Ein Beispiel für Workflowkontrolldaten sind Informationen über den Status von Instanzen.

3.2 Entwicklung von Workflow-Management-Systemen

Die Entwicklung von Workflow-Management-Systemen wird in vier Generationen gegliedert, wie Tabelle 2 zu entnehmen ist. Die Generationen stellen verschiedene Entwicklungsstufen der Workflowtechnologie dar, denen keine festen Zeitabschnitte zugeordnet sind. Die zur Zeit am weitesten entwickelten Systeme sind der dritten Generation zuzuordnen. Sie besitzen zwar Schnittstellen und Austauschformate, diese sind jedoch proprietär. Der Übergang in die vierte Generation setzt diesbezüglich Standards voraus, die noch nicht existieren. Dieses Problem will die Workflow Management Coalition, auf deren Arbeit in Abschnitt 5 näher eingegangen wird, lösen.

Generation	Hauptmerkmale
1.	Workflow-Management-Systeme als funktionaler Teil anderer Applikationen Statisch implementierte Workflowdefinitionen
2.	Workflow-Management-Systeme als eigenständige Applikationen Workflowdefinitionen sind mit Hilfe von Skriptsprachen anpaßbar
3.	Workflow-Management-Systeme mit generischen Diensten Workflowdefinitionen sind anpaßbar mit Hilfe von GUIs
4.	Workflow-Management-Systeme als sogenannte <i>Embedded enabler</i> Standardisierte Schnittstellen und Austauschformate

Tabelle 2: Generationen von Workflow-Management-Systemen²⁵

²⁵ [Teufel et al. 95], Seite 185

An anderer Stelle wird die Entwicklung von Workflow-Management-Systemen ebenso gegliedert. Zusätzlich wird jedoch den Systemen ab der 3. Generation die Verwendung von Datenbanksystemen zur Speicherung vorgangsbezogener Daten zugeschrieben.²⁶

3.3 Aufgaben und Ziele von Workflow-Management

Grundlegende Funktionen eines Workflow-Management-Systems beim Einsatz in einem Unternehmen oder in einer Behörde sind

- die Organisationsmodellierung
- die Definition von Vorgangstypen
- die Ausführung der Vorgänge und
- das Vorgangsmanagement.²⁷

Die Einführung eines Workflow-Management-Systems in einer Organisation stellt einen erheblichen Aufwand dar, der durch den Einsatz des Systems gerechtfertigt werden muß. Je nach Art des Unternehmens und der zu unterstützenden Prozesse können sich unterschiedliche Effekte ergeben. Einige Vorteile, die grundsätzlich durch den Einsatz von Workflow-Management-Systemen erreicht werden, sind²⁸:

- Erhöhte Produktivität: Zeiteinsparung wird durch die Vermeidung von Transport- und Liegezeiten und Parallelarbeit möglich.
- Nachweisbarkeit: Die Dokumentation von Abläufen wird von den Systemen übernommen. Das ist vor allem dort von Bedeutung, wo diese Dokumentation vorgeschrieben ist, zum Beispiel in Behörden.
- Qualitätssicherung: Das System überwacht die Ausführung von Aktivitäten und sorgt dafür, daß sie auch wirklich erledigt werden oder meldet zumindest, daß die Erledigung noch aussteht.
- Auskunftsfähigkeit: Der aktuelle Bearbeitungsstand eines Vorgangs kann jederzeit ermittelt werden.

3.4 Einsatz von Workflow-Management-Systemen

Workflow-Management-Systeme sind die technologische Basis für die rechnergestützte Ausführung der Geschäftsprozesse eines Unternehmens. Voraussetzung für den Einsatz solcher Systeme ist eine geeignete Modellierung der Geschäftsprozesse, woraus sich die wesentlichen Anforderungen des

²⁶ [Böhm/Schulze 95]

²⁷ [Böhm/Schulze 95]

²⁸ [Böhm/Schulze 95]

Unternehmens an ein Workflow-Management-System ableiten lassen. Workflow-Management-Systeme sind zum heutigen Zeitpunkt keine Standardsoftware. Sie unterscheiden sich in vielerlei Hinsicht und sind deshalb für die Bewältigung bestimmter Aufgaben unterschiedlich gut geeignet. Die Auswahl eines geeigneten Systems ermöglicht eine einfache Implementation der Prozeßmodelle und minimiert die notwendigen Anpassungen und Erweiterungen der Systeme. Eine Entscheidungshilfe stellen Kriterienkataloge dar, die anhand vieler Einzelkriterien einen Überblick über den Leistungsumfang der Produkte geben.

Grundsätzlich ist es denkbar, ein Workflow-Management-System lediglich zur Unterstützung der bestehenden Vorgänge eines Unternehmens einzusetzen. Damit wird aber das Potential der Systeme nicht voll genutzt. Bei entsprechender Planung wird durch den Einsatz der Systeme in der Regel auch ein Business Process Reengineering, eine Umgestaltung der bestehenden Prozesse, ermöglicht.

4 Workflow-Management-Systeme

In diesem Kapitel soll der Frage auf den Grund gegangen werden, welche Eigenschaften Software aufweisen muß, damit sie als Workflow-Management-System bezeichnet werden kann. Dazu werden in Abschnitt 4.1 Anforderungen aufgezählt, die sich zwingend aus dem Einsatzzweck der Systeme ergeben und die deshalb durch ein Workflow-Management-System erfüllt werden müssen. Ausgehend von diesen Anforderungen wird in Abschnitt 4.2 ein Vorschlag für eine allgemeine Architektur von Workflow-Management-Systemen entwickelt, die als Richtlinie für eine erfolgreiche Implementierung eines solchen Systems dienen kann.

4.1 Anforderungen an Workflow-Management-Systeme

In diesem Abschnitt werden die grundlegenden Anforderungen an Workflow-Management-Systeme beschrieben. Man unterscheidet funktionale und nichtfunktionale Anforderungen.²⁹ Die funktionalen Anforderungen bestehen zum einen in der Implementation eines Workflow-Metaschemas, das bestimmte Aspekte beinhaltet. Darauf wird in Abschnitt 4.1.1 eingegangen. Zum anderen muß ein Workflow-Management-System bestimmte Benutzerschnittstellen bieten, die in Abschnitt 4.1.2 beschrieben sind. Abschnitt 4.1.3 schließlich beschreibt die nichtfunktionalen Anforderungen an Workflow-Management-Systeme, die sich nach den Gesichtspunkten Offenheit, Zuverlässigkeit und Analysierbarkeit gruppieren lassen.

4.1.1 Aspekte von Workflow-Management-Anwendungen

Das **Workflow-Metaschema** eines Workflow-Management-Systems beschreibt die Eigenschaften aller Prozeßdefinitionen, die mit diesem System erstellt und verarbeitet werden können. Ein Workflow-Metaschema beinhaltet verschiedene Aspekte, mit denen die Verwaltung und Steuerung der Prozesse beschrieben wird. Abbildung 5 zeigt einen Ausschnitt aus dem Workflow-Metaschema der Workflowkomponente von Oracle InterOffice 4.1 .

²⁹ [Jablonski et al. 97], S. 219 ff

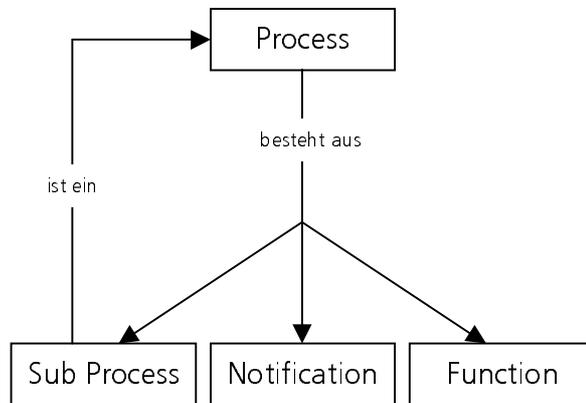


Abbildung 5: Ausschnitt des Workflow-Metaschemas von InterOffice 4.1

Die folgenden Aspekte sollten von jedem System unterstützt werden:

- Der **Funktionsaspekt** bestimmt, welche Aktionen innerhalb von Prozessen ausgeführt werden. Damit wird die Zusammensetzung von Prozessen aus Subprozessen und Aktivitäten beschrieben.
- Der **Verhaltensaspekt** beschreibt den Kontrollfluß in und zwischen Prozessen. Das System sollte in der Lage sein, dabei auftretende Probleme zu behandeln.
- Der **Informationsaspekt** beschreibt den Datenfluß. Damit wird es möglich, Workflowkontrolldaten und workflowrelevante Daten zu verwalten.
- Für die Zuweisung von Aufgaben an Akteure wird der **Organisationsaspekt** benötigt. Zu diesem Zweck muß das System über ein Abbild der Organisationsstruktur verfügen.
- Der **Operationsaspekt** beschreibt den Aufruf von nicht zum Workflow-Management-System gehörenden Anwendungsprogrammen. Dabei sollten dem System bestimmte Eigenschaften der Anwendungen zugänglich gemacht werden, die für das System von Interesse sind, z.B. Transaktionseigenschaften.

Wie diese Aspekte implementiert werden, bleibt den Herstellern der Systeme überlassen. Für auf bestimmte Anwendungsgebiete spezialisierte Systeme können weitere Aspekte hinzukommen.

4.1.2 Benutzerschnittstellen von Workflow-Management-Systemen

Ein wesentliches Merkmal von Workflow-Management-Systemen ist der hohe Interaktionsgrad mit den Benutzern des Systems. Anhand der Aufgaben dieser

Benutzer lassen sich Schnittstellen festlegen, die jedes System zur Verfügung stellen muß, um einen Einsatz des Systems zu ermöglichen.

- Die Entwicklerschnittstelle gestattet die Erstellung, Verwaltung und Analyse von Prozeßdefinitionen. Neben den dafür benötigten Tools muß dem Entwickler die Möglichkeit gegeben sein, Schnittstellen zu anderen Systemen im Workflow-Management-System zu implementieren.
- Die Anwenderschnittstelle oder Arbeitsliste präsentiert dem Benutzer die zu erledigenden Aufgaben und ermöglicht deren Ausführung sowie das Erstellen neuer Aufgaben und das Abbrechen oder Weiterleiten von Instanzen. Auch Laufzeitänderungen von Abläufen sollten unterstützt werden.
- Administratoren erledigen die Benutzerverwaltung, die Konfiguration des Systems sowie die Überwachung und Steuerung der Abläufe. Das System muß die entsprechenden Schnittstellen bieten und außerdem die Erstellung von Statistiken und Historien gestatten.

4.1.3 Nichtfunktionale Anforderungen

4.1.3.1 Offenheit

In diesem Abschnitt sind Anforderungen an die Offenheit von Workflow-Management-Systemen, d.h. bezüglich der Erweiterbarkeit, Plattformunabhängigkeit, Verteilbarkeit, Skalierbarkeit und der Interoperabilität zusammengefaßt.

Grundsätzlich sollen die Systeme bestehende Lösungen, die sich bereits bewährt haben, ergänzen und sie nicht ersetzen. Frei zugängliche, dokumentierte Schnittstellen der Systemkomponenten ermöglichen die Integration in bestehende Hard- und Softwareumgebungen. Damit wird auch die Erweiterbarkeit der Funktionalität des Systems gesichert: vorhandene Komponenten können ergänzt und neue hinzugefügt werden.

Workflow-Management-Systeme werden auch in verteilten Einrichtungen eingesetzt, z.B. in verschiedenen Niederlassungen eines Unternehmens. Deshalb werden sie in der Regel als verteilte Systeme realisiert. In den meisten Unternehmen findet man heute heterogene Umgebungen bezüglich der eingesetzten Hardware und der Betriebssysteme vor. Da der Einsatz von Workflow-Management-Systemen nur dann Sinn macht, wenn er möglichst viele Unternehmensteile einbezieht, müssen die Systeme auch diesbezüglich offen sein, das heißt einzelne Komponenten müssen auf unterschiedlichen Plattformen lauffähig sein. Um Plattformunabhängigkeit zu gewährleisten, bietet sich die Verwendung von Middleware auf der Basis aktueller Standards an. In Abschnitt 7.4 wird auf die Middleware-Architektur CORBA näher eingegangen.

Wichtig ist weiterhin die Skalierbarkeit des Systems, da die benötigte Leistungsfähigkeit zu Beginn des Einsatzes nur sehr grob abgeschätzt werden kann. Die Skalierbarkeit des Systems kann durch folgende Konzepte erreicht werden³⁰:

- Replikation von Komponenten
- Splitten von Komponenten auf mehrere Server
- Migration von Komponenten zwischen verschiedenen Servern

Verteilbarkeit und Skalierbarkeit müssen auch zur Laufzeit gewährleistet sein, da ein Stillstand des Systems nach Möglichkeit zu vermeiden ist und sich die Einsatzbedingungen während der Laufzeit mit hoher Wahrscheinlichkeit ändern.

Die Unterstützung des Operationsaspekts stellt an Workflow-Management-Systeme die Anforderung der Offenheit gegenüber anderen, externen Anwendungen, die zur Ausführung bestimmter Aufgaben genutzt werden. Für eine vollständige Integration ist aber auch eine gleichberechtigte Kopplung von Workflow-Management-Systemen mit anderen Anwendungen notwendig. Das ist zum Beispiel der Fall, wenn ein Workflow-Management-System mit einer anderen Groupware gekoppelt werden soll, da beide Systeme unterschiedliche, unter Umständen lang andauernde Aufgaben unter verschiedenen Gesichtspunkten ausführen, die sich gegenseitig beeinflussen können.

4.1.3.2 Zuverlässigkeit

In einem Workflow-Management-System laufen in der Regel viele Prozesse parallel ab. Deshalb sind für die fehlerfreie Arbeit des Systems Mechanismen erforderlich, die Inkonsistenzen verhindern. In Fehlersituationen müssen diese Mechanismen die Fehlerbehandlung übernehmen und auf schnellstem Weg wieder einen konsistenten Systemzustand herstellen. Diese Forderung legt den Einsatz eines Datenbanksystems nahe. Datenbanksysteme benutzen zur Vermeidung von Inkonsistenzen das Konzept der Transaktionen, das die folgenden Eigenschaften beinhaltet³¹:

- Atomizität: Eine Transaktion wird entweder vollständig oder gar nicht ausgeführt.
- Konsistenz: Eine Transaktion überführt das System von einem konsistenten Zustand in einen anderen konsistenten Zustand.
- Isolation: Transaktionen beeinflussen sich nicht gegenseitig.
- Dauerhaftigkeit: Das Resultat einer Transaktion ist persistent.

³⁰ [Jablonski et al. 97], Seite 227

³¹ [Lockemann 93], S. 405

Die Anwendung des Datenbank-Transaktionskonzepts auf Workflow-Management-Systeme erscheint vielversprechend, führt aber ohne Änderungen am Konzept zu Problemen, da zwischen klassischen Datenbanktransaktionen und Workflows einige wesentliche Unterschiede bestehen:

- Dauer: Das Transaktionskonzept ist für sehr kurze Operationen gedacht. Workflows dauern in der Regel viel zu lange, um für die gesamte Zeit ihrer Ausführung irgendwelche Ressourcen blockieren zu können, ohne daß es zu Verklemmungen kommt.
- Struktur: Workflows können im Gegensatz zu Transaktionen hierarchisch aufgebaut sein.
- Atomizität: Das Zurücksetzen von Aktivitäten im Fehlerfall ist für arbeitsintensive Workflows nicht geeignet.
- Isolation: Transaktionen konkurrieren um Ressourcen, Workflows hingegen können kooperativ sein. Weiterhin wird die Isolation von Transaktionen durch Serialisierung erreicht, was für Workflows nicht möglich ist.

Aufgrund der angeführten Unterschiede wird klar, daß einige Eigenschaften des Transaktionskonzepts erweitert und andere abgeschwächt werden müssen. Eine Erweiterung ist beispielsweise bezüglich der Transaktionsstruktur notwendig. Abgeschwächt werden muß hingegen die Isolationseigenschaft, die in Datenbanken mittels Serialisierbarkeit gesichert wird.

Genauere Betrachtungen zur Anwendung des Transaktionskonzepts in Workflow-Management-Systemen legen die Verwendung eines Datenmodells mit sogenannten Kontextelementen nahe. Dabei wird unter Verweis auf die sehr unterschiedliche Lebensdauer von Transaktionen und Workflows formuliert:

"Quite obviously, phenomena which are 9 orders of magnitude away from simple modules, need different programming techniques and different styles of talking about consistency, for example."³²

4.1.3.3 Analysierbarkeit

Das Verhalten eines Workflow-Management-Systems muß in jeder Situation nachvollziehbar sein. Monitorprogramme sind für die Kontrolle des Systemzustands, z.B. die Ermittlung von Fehlerzuständen und Engpässen, notwendig. Eine Historie ermöglicht die Analyse beendeter Workflowinstanzen. Mittels Logfiles lassen sich die Aktionen des Systems jederzeit nachvollziehen.

³² [Reuter/Schwenkreis]

Für den Aufbau einer Historie ist es notwendig, den Zustand beliebiger Objekte zu unterschiedlichen Zeitpunkten zu kennen. Deshalb sollten Veränderungen eines Objekts nicht einfach den alten Wert überschreiben, sondern eine neue Version dieses Objektes erzeugen. Dieser Mechanismus wird als Objektversionierung bezeichnet.

4.2 Architektur von Workflow-Management-Systemen

Ausgehend von den im Abschnitt 4.1 genannten funktionalen und nichtfunktionalen Anforderungen kann die Architektur von Workflow-Management-Systemen auf einer abstrakten Ebene beschrieben werden.³³ Eine funktionale Komponentenarchitektur ordnet jedem zu unterstützenden Aspekt eine Komponente zu. Die Zusammenfassung dieser Komponenten zu einem System entspricht der Workflow-Engine. Den Benutzerschnittstellen werden ebenfalls Komponenten zugeordnet, die in einem Werkzeugsystem zusammengefaßt werden. Die Interaktion der Aspektkomponenten erfolgt über eine Koordinationskomponente, die auch die Verbindung der Workflow-Engine zum Werkzeugsystem herstellt. Durch die Koordinationskomponente wird die Anzahl der Schnittstellen innerhalb der Architektur erheblich verringert.

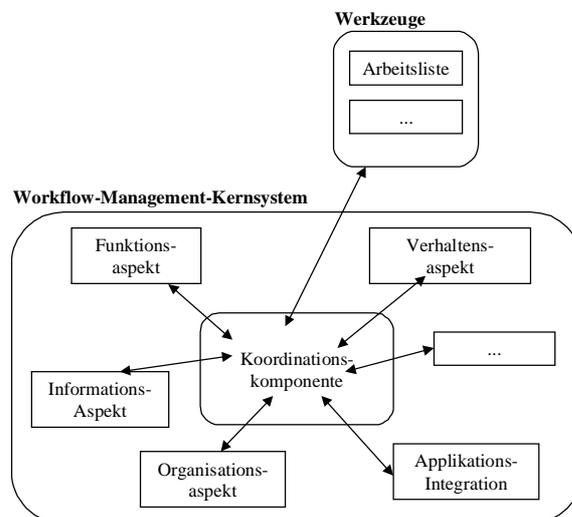


Abbildung 6: Komponentenarchitektur eines Workflow-Management-Systems³⁴

³³ [Jablonski et al. 97], Seite 231 ff

³⁴ [Jablonski et al. 97], Seite 235

Zur Unterstützung der nichtfunktionalen Anforderungen wird eine Schichtenarchitektur vorgeschlagen, in deren oberster Schicht die funktionalen Module des Systems untergebracht sind. Die unterste Schicht wird durch Basisdienste wie Betriebssystem und Datenbanksystem gebildet. In den dazwischenliegenden Schichten der Architektur kommt sogenannte Middleware zum Einsatz. Der Begriff Middleware wird wie folgt definiert:

Middleware ist anwendungsneutrale Systemsoftware, die zwischen Anwendungsprogrammen und Betriebssystem, Datenbanksystemen und anderen Ressourcen-Managern läuft. Sie stellt Infrastrukturdienste zur Verfügung, auf deren Basis Applikationen ortstransparent entwickelt und betrieben werden können. Solche Dienste sind unter anderem Kommunikations-, Datenmanagement-, Prozeßmanagement-, Transaktionsmanagement- sowie Netzwerk- und Systemmanagement-Dienste.³⁵

Die Module der funktionalen Schicht können so ohne Berücksichtigung ihrer Verteilung entwickelt werden. Die Middleware-Schicht für Client-Server-Kommunikation dient der Transparenz bezüglich Verteilung und Heterogenität. Auch die Interoperabilität zwischen verschiedenen Middleware-Produkten wird hier realisiert. Die nächste Schicht sorgt mit Hilfe von Replikationsmechanismen für die Verfügbarkeit des Systems. In der Skalierbarkeitsschicht erfolgt eine geeignete Partitionierung der Daten. Durch diese Architektur wird ein komplexes Problem in mehrere weniger komplexe Teilprobleme zerlegt.

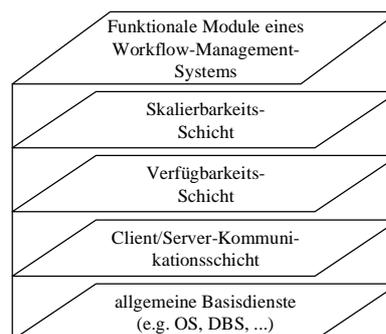


Abbildung 7: Schichtenarchitektur eines Workflow-Management-Systems³⁶

³⁵ [Jablonski et al. 97], Seite 487

³⁶ [Jablonski et al. 97], Seite 241

5 Standardisierung: Die Workflow Management Coalition

Die in diesem Abschnitt benutzten Begriffe sind im Glossar der WfMC festgelegt, ihre Bedeutung kann aber auch Abschnitt 3 dieser Arbeit entnommen werden, der sich weitgehend an die Begriffsbestimmungen der WfMC hält. Die Abbildungen in diesem Abschnitt stammen ebenfalls aus den Dokumenten der WfMC.

5.1 Die Workflow Management Coalition

Seit Beginn der 90er Jahre hat eine Vielzahl von Herstellern ebenso viele verschiedene Workflow-Management-Systeme hervorgebracht. Die Unterschiede zwischen den einzelnen Workflow-Management-Systemen haben ihre Ursache zu einem großen Teil in der unterschiedlichen Herkunft der Systeme. Einige Systeme wurden von Anfang an als Workflowprodukte entwickelt. Es gibt aber auch Produkte, deren Ursprung z.B. ein Dokumentenmanagementsystem, Datenbanksystem oder E-Mail-System ist, in das Workflow-Funktionalität erst im Nachhinein integriert wurde. Aus diesem Grund ist es schwierig, das grundlegende Konzept von Workflow-Management-Systemen zu identifizieren. Angesichts der Gefahr der Entwicklung neuer Informationsinseln kamen Anwender und Hersteller relativ schnell zu der Einsicht, daß die Schaffung von Standards notwendig ist.

Die WfMC wurde im August 1993 gegründet und ist eine internationale Organisation von Firmen und Einrichtungen, die Workflowprodukte anbieten, nutzen oder analysieren. Die Organisation hat über 100 Mitglieder und stellt das wichtigste Organ zur Schaffung von Standards auf dem sich schnell entwickelnden Markt der Workflowprodukte dar.

Die Ziele der WfMC sind die Schaffung einer einheitlichen Terminologie für den Bereich Workflow-Management und die Festlegung von Standards für die Interoperabilität von verschiedenen Workflowprodukten. Seit Juni 1996 liegen die Begriffsdefinitionen der Workflow Management Coalition in der Version 2.0 vor.³⁷

Die WfMC entwickelte ein Referenzmodell für Workflow-Management-Systeme. Das Modell legt allgemeine Eigenschaften der Systeme, deren

³⁷ [WfMC Glossary]

funktionale Bereiche und die benötigten Schnittstellen zwischen diesen Bereichen fest. Das Modell enthält nur wenige Angaben zum internen Aufbau der Systeme und konzentriert sich auf die Standardisierung der Schnittstellen.

5.2 Grundlegende Funktionen eines Workflow-Management-Systems

Workflow-Management-Systeme bieten Unterstützungsfunktionen für den Build Time-Bereich, in dem die Prozesse definiert und möglicherweise auch modelliert werden, und für den Run Time-Bereich, in dem die Ausführung der Prozeßinstanzen kontrolliert wird und die Interaktion mit den Benutzern und mit den Workflow-Anwendungen stattfindet.

Abbildung 8 zeigt die Zusammenhänge zwischen den Bereichen. Einige Workflow-Management-Systeme erlauben Änderungen der Prozeßdefinitionen aus dem Run Time – Bereich heraus, was durch den unterbrochenen Pfeil verdeutlicht wird.

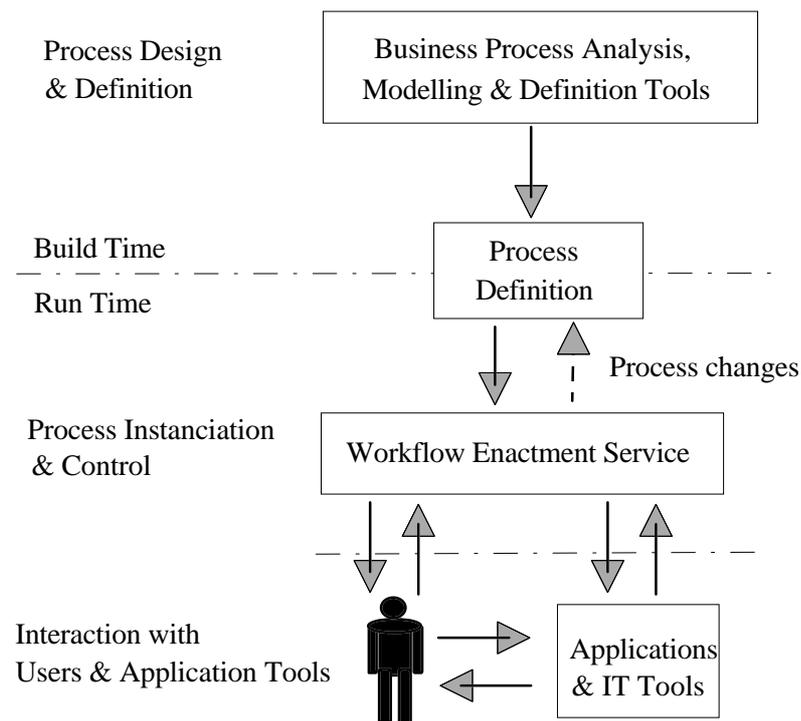


Abbildung 8: Workflow System Characteristics

Der Austausch von Prozeßdefinitionen zwischen verschiedenen Systemen soll standardisiert werden, die Erzeugung der Definitionen wird von den Bemühungen der WfMC jedoch nicht erfaßt.

5.3 Generische Struktur von Workflow-Management-Systemen

Abbildung 9 zeigt das von der WfMC erstellte allgemeine Modell eines Workflow-Management-Systems. Dieses Modell beschreibt den Aufbau der meisten auf dem Markt befindlichen Produkte und soll die Grundlage für die Entwicklung verschiedener Stufen der Interoperabilität bieten.

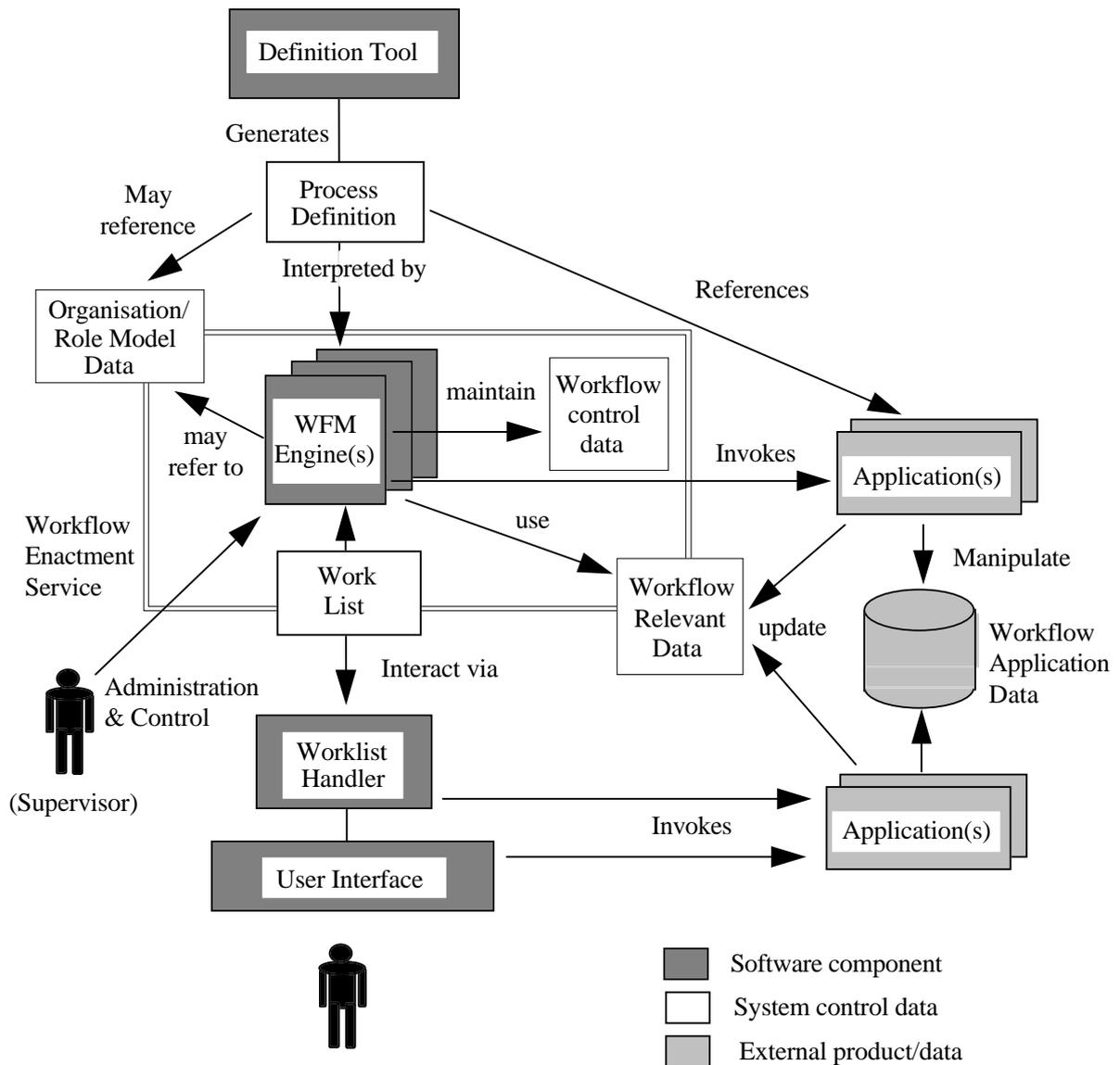


Abbildung 9: Generic Workflow Product Structure

Das Modell legt die wichtigsten funktionalen Komponenten eines Workflow-Management-Systems und die Schnittstellen zwischen ihnen fest. Der Workflow Enactment Service ist die zentrale Komponente des Systems. Er verarbeitet die Prozeßdefinitionen und erzeugt und verwaltet die Prozeßinstanzen.

5.4 Das Referenzmodell der WfMC

Aus dem allgemeinen Modell eines Workflow-Management-Systems entwickelte die WfMC ein Referenzmodell, indem sie die Schnittstellen festlegte, die für die Interoperabilität verschiedener Systeme notwendig sind. Während Abbildung 9 weitgehend vollständig den Aufbau von Workflowprodukten beschreibt, konzentriert sich das in Abbildung 10 dargestellte Referenzmodell auf die abstrakte Darstellung der Interoperabilitätsschnittstellen.

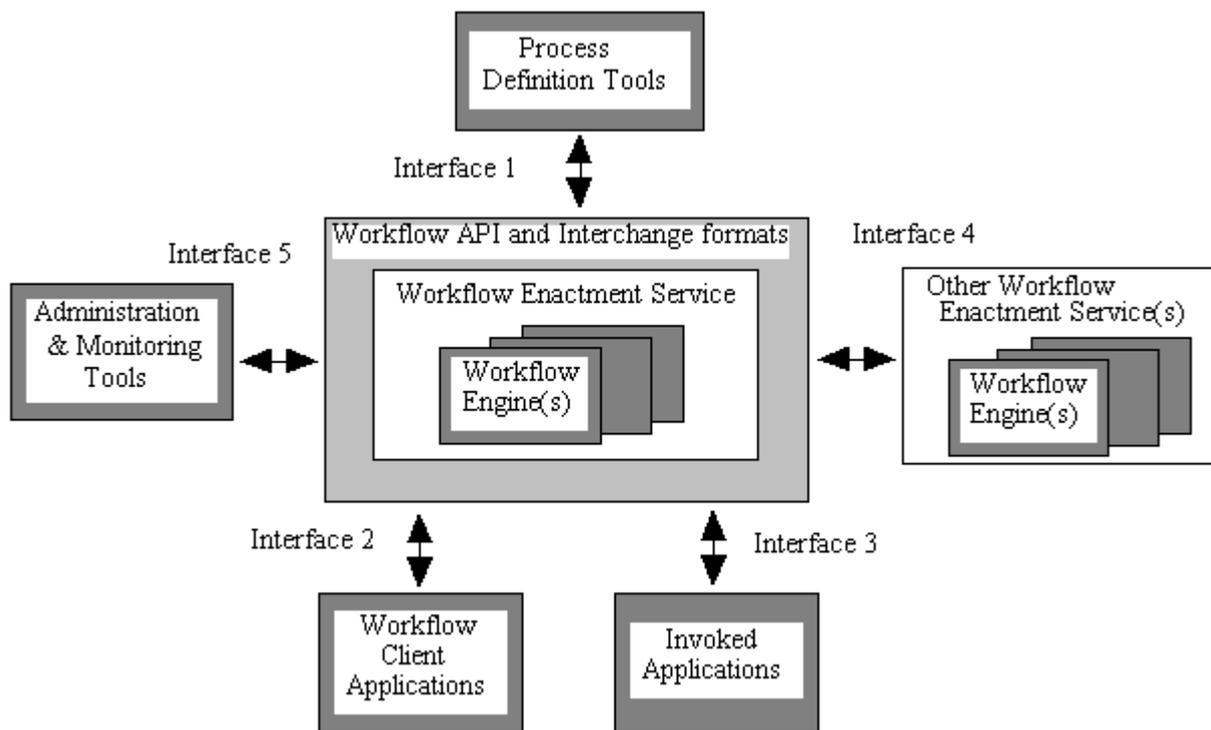


Abbildung 10: Workflow Reference Model - Components & Interfaces

Der Workflow Enactment Service stellt eine logische Einheit dar, kann jedoch zentral oder verteilt implementiert werden. Verteilte Systeme müssen die Workflow Engines synchronisieren und den Informationsaustausch zwischen ihnen gewährleisten. In einem homogenen Workflow Enactment Service sind

diese Operationen anbieterspezifisch. Wenn mehrere unterschiedliche Produkte verwendet werden, ist eine standardisierte Schnittstelle zwischen den Workflow Engines notwendig. Diese Aufgabe übernimmt das Interface 4.

5.5 WAPI – Die Schnittstellen des Referenzmodells

Die fünf Schnittstellen des Workflow Enactment Service bezeichnet man zusammen als WAPI – Workflow Application Interfaces and Interchange Formats. Man kann WAPI auch als eine Schnittstelle mit 5 funktionalen Bereichen ansehen, da einige Funktionen in mehreren Bereichen benötigt werden. Dieser Abschnitt beschreibt die einzelnen Schnittstellen und die Schlußfolgerungen, die sich daraus für die beteiligte Software ergeben. Eine Übersicht über den derzeitigen Entwicklungsstand der Schnittstellen findet sich in Abschnitt 5.7.

5.5.1 Interface 1: Prozeßdefinitionen

Das Interface 1 des Referenzmodells besteht aus einem Satz aufzurufender Funktionen und ist für den Austausch von Prozeßdefinitionen zuständig. Ohne eine standardisierte Schnittstelle zum Austausch der Prozeßdefinitionen ist die gemeinsame Verwendung verschiedener Produkte nur möglich, wenn für jede Kombination aus Modellierungs- und Ausführungsumgebung ein Übersetzungsverfahren implementiert wird. Die Verwendung von Interface 1 reduziert diesen Aufwand darauf, das jedes Produkt seine internen Daten gemäß der Schnittstellendefinition zur Verfügung stellt.

Es wird nichts über die Software ausgesagt, mit der die Definitionen erstellt werden. Sie kann entweder Bestandteil des Workflowproduktes oder eine eigenständige Anwendung sein. Im ersten Fall liegen die erstellten Prozeßdefinitionen innerhalb eines Systems vor und für die Ausführung der zugehörigen Prozesse ist ein Zugriff über eine standardisierte Schnittstelle nicht erforderlich. Handelt es sich jedoch um verschiedene Produkte, dann können die Definitionen über das Interface 1 ausgetauscht werden. Es ermöglicht den Anwendern, Modellierungssoftware und Runtime-Umgebung unabhängig voneinander zu wählen. Dadurch können Produkte eingesetzt werden, die für die jeweilige Aufgabe optimal geeignet sind. Weiterhin können die gleichen Prozeßdefinitionen von verschiedenen Workflowprodukten genutzt werden, was eine Voraussetzung für den gleichzeitigen Einsatz verschiedener Workflow-Management-Systeme zur Lösung umfassender Aufgaben ist.

Die Arbeit der WfMC zum Interface 1 konzentriert sich auf folgende Aufgaben:

- Erstellen eines Metamodells für Prozeßdefinitionen, das in der Lage ist, die Bestandteile von Prozeßdefinitionen sowie ihre Eigenschaften und

Beziehungen auszudrücken. Die WfMC entwickelt ein solches Metamodell, das grundlegende, relativ einfache Definitionen erfaßt.

- Festlegen einer Menge von Funktionsaufrufen, die das Bereitstellen und Abrufen von Prozeßdefinitionen ermöglichen.

5.5.2 Interface 2: Client-Anwendungen

Im Referenzmodell der WfMC findet die Interaktion zwischen den Client-Anwendungen und dem Enactment Service durch das Interface 2 statt, das als Arbeitsliste (englisch *worklist*) organisiert ist. Die Einträge in dieser Liste symbolisieren Aufgaben, die von einem bestimmten Nutzer oder einer Gruppe von Nutzern zu erledigen sind. Die Aktivierung der einzelnen Einträge in der Arbeitsliste wird entweder durch die Client-Anwendung oder den Benutzer gesteuert.

Die Workflow Engine fügt Einträge in die Arbeitsliste ein, wenn während der Abarbeitung einer Prozeßinstanz eine Aktion durch einen Benutzer erforderlich ist. Die Client-Anwendung benutzt die Arbeitsliste, um diese Einträge auszulesen und dem Benutzer zur Abarbeitung anzubieten. Neben diesen grundlegenden Funktionen gibt es weitere, die das Entfernen abgearbeiteter Einträge, das Markieren von Einträgen als zwischenzeitlich nicht ausführbar usw. ermöglichen.

Eine Client-Anwendung kann grundsätzlich mit mehreren unterschiedlichen Enactment Services interagieren. Dem Nutzer gegenüber kann sie die Einträge in beliebiger Form präsentieren, z.B. als eine einheitliche Liste. Die Schnittstelle wird auf verschiedene Kommunikationsmechanismen abgebildet, um der Vielfalt der bestehenden Systeme Rechnung zu tragen.

5.5.3 Interface 3: Invoked Applications

Workflow-Management-Systeme interagieren mit Anwendungen, um die verschiedensten Aufgaben auszuführen. Dabei ist es unmöglich, mit allen möglichen Anwendungen direkt zu interagieren, weil prinzipiell alle möglichen Anwendungen auf allen Plattformen in allen Netzwerkumgebungen für eine Interaktion in Frage kommen.

Manche Workflow-Management-Systeme interagieren mit Anwendungen eines bestimmten Typs, bei dem die Daten einfach übergeben werden können und ihre Struktur bekannt ist, z.B. Textverarbeitungen. Andere Möglichkeiten sind standardisierte Austauschmechanismen sowie Anwendungsagenten, welche die Unterschiede der Anwendungen verbergen. Es ist auch möglich, *workflowfähige Anwendungen* zu entwickeln, die zur Interaktion mit dem Workflow Enactment Service eine standardisierte Schnittstelle benutzen.

Die Arbeit der WfMC besteht zunächst darin, einen Katalog von Interfacetypen und eine Menge von Schnittstellen zu entwickeln, die in zukünftigen, workflowspezifischen Anwendungen genutzt werden können.

Im einfachsten Fall erfolgt die Einbindung der Anwendungen lokal in einer Workflow Engine. Die Information über die Anwendungen befindet sich in den Prozeßdefinitionen. Vereinbarungen über Bezeichnung und Adressierung der Anwendungen sind dann ebenfalls lokal bezüglich der Workflow Engine. In einer heterogenen Umgebung müssen die Informationen über die Anwendungen gegebenenfalls zwischen den Engines ausgetauscht werden, wofür Interface 4 zuständig ist.

5.5.4 Interface 4: Interoperabilität

Interface 4 soll verschiedenen Workflow-Management-Systemen die Möglichkeit bieten, über die Systemgrenzen hinweggehende Arbeitsvorgänge zu unterstützen. Über die Schnittstelle werden Aktivitäten und Subprozesse gestartet, Prozeßdefinitionen, workflowrelevante und Anwendungsdaten sowie Status- und Steuerungsinformationen ausgetauscht und die Ausführung von Instanzen an bestimmten Punkten synchronisiert.

Genauere Angaben zu den von der WfMC festgelegten verschiedenen Stufen der Interoperabilität und den damit einhergehenden Konsequenzen sind im Rahmen dieser Arbeit von größerer Bedeutung und finden sich in Abschnitt 5.6.

5.5.5 Interface 5: Administration und Überwachung

Interface 5 des Referenzmodells dient als Standardschnittstelle für Administrations- und Überwachungsaufgaben. Dadurch wird es möglich, verschiedene Workflow Enactment Services einheitlich zu überwachen und damit einen vollständigen Überblick über den Status der Abläufe in einer Organisation zu erhalten. Die Schnittstelle bietet Funktionen für die Bereiche Benutzermanagement, Rollenmanagement, Aufzeichnung von Ereignissen, Ressourcenkontrolle und Management von Statusinformationen.

5.6 Interoperabilität

Die WfMC beschreibt verschiedene Szenarien, die sich auf die systemübergreifende Ausführung von Prozessen beziehen. Einen Überblick darüber gibt Abschnitt 5.6.1. Weiterhin unterscheidet die WfMC acht Stufen der Interoperabilität von Workflow-Management-Systemen, die in Abschnitt 5.6.2 aufgeführt sind.

5.6.1 Interoperabilitätsszenarien von Workflow-Management-Systemen³⁸

Die WfMC will Interoperabilität zwischen Workflow-Management-Systemen erreichen, ohne dabei die Leistungsfähigkeit der Produkte einzuschränken. Deshalb wurden verschiedene Interoperabilitätsszenarien entworfen. Dabei ist zu erwarten, daß die Produkte zunächst nur die relativ einfachen Szenarios unterstützen, in denen einzelne Teile von Prozessen zwischen verschiedenen Systemen übertragen werden. Ein engeres Zusammenwirken mehrerer Workflow-Engines hingegen erfordert unter anderem die gemeinsame Benutzung derselben Prozeßdefinitionen und Workflowkontrolldaten.

5.6.1.1 Szenario 1: *Connected Discrete*

In diesem Szenario ist es möglich, daß zwischen den beteiligten Systemen eine einzelne Instanz übertragen wird, die in der neuen Umgebung unabhängig von der alten Umgebung ausgeführt wird. Die Prozesse sind nur an einem bestimmten Punkt miteinander verbunden. Deshalb wird dieses Szenario von der WfMC als *Connected Discrete* oder *Chained* bezeichnet.

In Abbildung 11 ist der Fall dargestellt, daß die Verbindung zwischen dem Endpunkt eines Prozesses A und dem Startpunkt eines Prozesses B besteht. Verbindungen zwischen beliebigen anderen Punkten der Prozesse sind aber auch möglich.

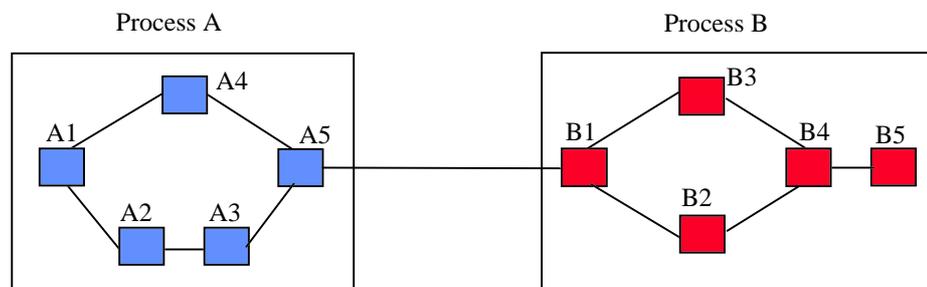


Abbildung 11: Chained Services Model

³⁸ [WfMC Reference Model], Abschnitt 3.7

5.6.1.2 Szenario 2: *Hierarchical*

Auf dieser Stufe wird die Ausführung von Subprozessen in anderen Workflow-Engines möglich. Zwischen den Prozessen besteht eine hierarchische Beziehung: Prozeß B ist ein Subprozeß von Prozeß A. Das Szenario trägt deshalb die Bezeichnung *Hierarchical* oder *Nested Subprocesses*. In Abbildung 12 enthält Prozeß A eine Aktivität, die die Ausführung des Subprozesses B in einer anderen Workflow-Engine bewirkt.

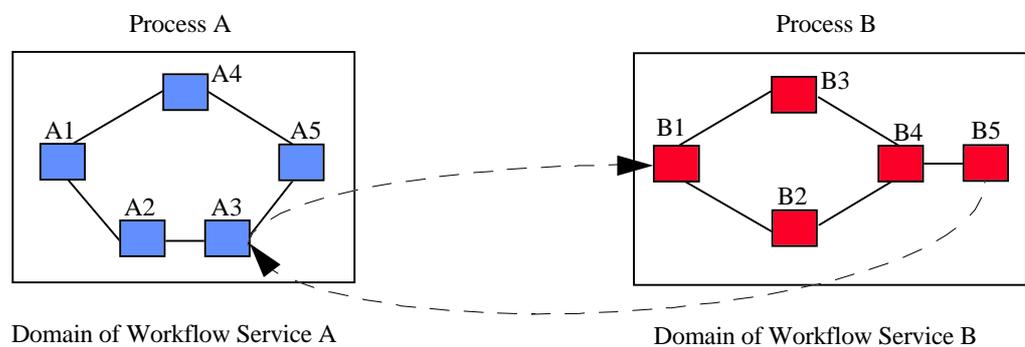
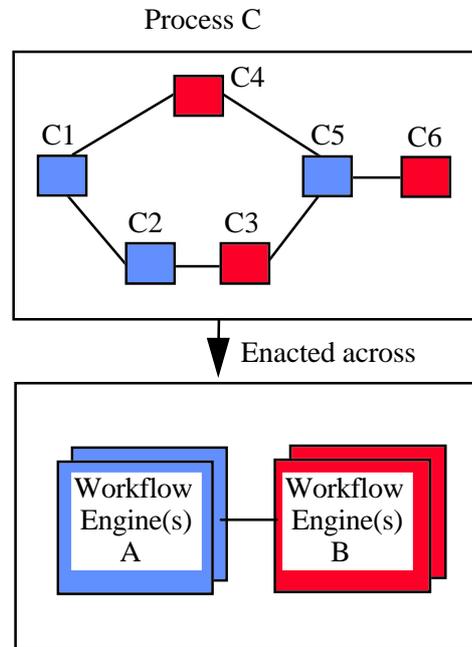


Abbildung 12: Nested Subprocesses Model

5.6.1.3 Szenario 3: *Connected Indiscrete*

In diesem Szenario bestehen Prozesse aus Aktivitäten, die in verschiedenen Workflow-Engines ausgeführt werden. Die Interaktionen zwischen den Workflow-Engines erfordern keine besonderen Handlungen durch Administratoren oder Nutzer. Sie erfolgen vielmehr über eine Schnittstelle, die von beiden Workflow-Engines unterstützt werden muß. Die WfMC bezeichnet diese Form der systemübergreifenden Ausführung von Prozessen als *Connected Indiscrete* oder *Peer-to-Peer-Model*. Abbildung 13 zeigt einen Prozeß C, dessen Aktivitäten in verschiedenen Umgebungen ausgeführt werden.



Shared Domain of Workflow Services A & B

Abbildung 13: Peer-to-peer Model

5.6.1.4 Szenario 4: *Parallel Synchronized*

In diesem Szenario werden in verschiedenen Systemen parallel ablaufende Prozesse miteinander synchronisiert. Deshalb wird es als *Parallel Synchronized* bezeichnet. Es erfordert einen Event-Mechanismus, der von verschiedenen Workflow-Engines genutzt werden kann. Dieses Szenario wird von den derzeitigen Aktivitäten der WfMC nicht erfaßt.

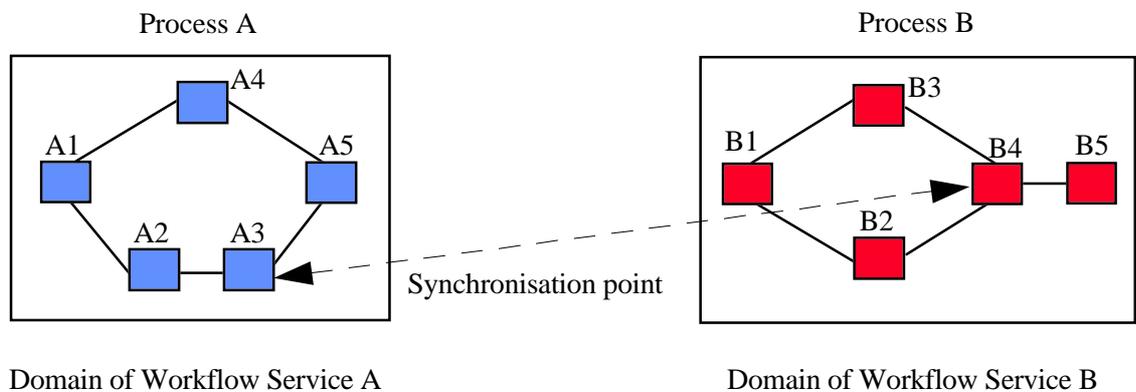


Abbildung 14: Parallel Synchronized Model

5.6.2 Stufen der Interoperabilität von Workflow-Management-Systemen

Der Zweck von Interface 4 des Referenzmodells der WfMC wurde in Abschnitt 5.5.4 bereits genannt. Die grundlegenden Unterschiede zwischen den Produkten der verschiedenen Hersteller lassen eine schnelle Umsetzung aller Funktionen des Interfaces unrealistisch erscheinen. Deshalb unterscheidet die WfMC acht Stufen der Interoperabilität³⁹, die von den unterschiedlichen Systemen umgesetzt werden können. Die Stufen bauen aufeinander auf, d.h. ein System einer bestimmten Stufe muß auch die Anforderungen der niedrigeren Stufen erfüllen.

- **Level 1 - No Interoperability:** Es gibt keine Möglichkeit der Interaktion mit anderen Produkten und somit auch keinerlei Interoperabilität.
- **Level 2 - Coexistence:** Auf dieser Stufe gibt es keinen standardisierten Ansatz für die Interoperabilität von Workflowprodukten. Verschiedene Workflow-Management-Systeme teilen sich dieselbe Umgebung bezüglich Hardware, Betriebssystem und Netzwerk. Es gibt aber keine direkte Interaktion zwischen den Systemen.
- **Level 3 - Unique Gateways:** Mittels Gateways ist es den Systemen möglich, Daten untereinander auszutauschen. Sind mehrere Systeme beteiligt, übernimmt das Gateway auch Routing-Funktionen. Erfolgt der Austausch über standardisierte Gateways, wird Level 3a (Common Gateway API) erreicht.
- **Level 4 - Limited Common API Subset:** Die Systeme haben eine standardisierte Schnittstelle, die eine direkte Interaktion zwischen ihnen ermöglicht. Die noch festzulegende Schnittstelle beinhaltet grundlegende Funktionen, die von möglichst allen Workflowprodukten unterstützt werden sollten.
- **Level 5 - Complete Workflow API:** Die Systeme haben eine einzige standardisierte Schnittstelle, die einen vollen Zugriff auf alle Operationen ermöglicht, ausgenommen solche, die der Ausrichtung eines Systems auf ein Spezialgebiet zuzuordnen sind.
- **Level 6 - Shared Definition Formats:** Die Systeme benutzen ein gemeinsames Format für Prozeßdefinitionen, das Routing-Aufgaben, Zugriffsrechte für Nutzer und den Umgang mit Ressourcen beinhaltet. Die WfMC will eine Prozeßdefinitionssprache (WPDL) definieren, die die allen Systemen gemeinsame Funktionalität bestimmt. Weiterhin sollen die Workflowprodukte entsprechend ihrer Ausrichtung in Klassen eingeteilt werden, für die die Funktionalität jeweils entsprechend erweitert wird.

³⁹ [WfMC-TC-1012], Abschnitt 3.2.3

- **Level 7 – Protocol Compatibility:** Sämtliche Interaktionen zwischen den Systemen einschließlich der Übertragung von Prozeßdefinitionen, Transaktionen und Recovery-Mechanismen sind standardisiert.
- **Level 8 – Common Look and Feel Utilities:** Zu den Forderungen von Level 7 kommt der Einsatz einer einheitlichen Benutzerschnittstelle hinzu. Die WfMC selbst ist der Ansicht, daß diese Stufe aus kommerziellen und praktischen Gründen wohl nicht erreicht wird.

5.7 Stand der Arbeit der WfMC

Zur Zeit ist ohne größeren Aufwand keine Zusammenarbeit von Workflow-Management-Systemen verschiedener Hersteller möglich. Das wird an der in Kapitel 8 dieser Arbeit vorgestellten Beispielimplementation deutlich, die auf der Grundlage der aktuellen Spezifikationen der WfMC⁴⁰ und der OMG⁴¹ eine Interoperabilitätsschnittstelle für ein bestimmtes Workflowprodukt realisiert.

Die WfMC definiert in ihrem Referenzmodell Schnittstellen, deren Standardisierung die Interoperabilität solcher Systeme ermöglichen soll. Damit werden bestimmte Module eines Workflow-Management-Systems festgelegt. Der innere Aufbau dieser Module bleibt dem Hersteller überlassen. Dadurch soll es möglich werden, daß einzelne Module eines Workflow-Management-Systems durch Module anderer Systeme und Hersteller ersetzt werden können.

Der aktuelle Stand der Arbeit der WfMC findet sich im WWW unter <http://www.aiim.org/wfmc/>. Einen Überblick über die derzeit⁴² verfügbaren Dokumente gibt Tabelle 3. Die Spezifikation der Schnittstellen ist aber keinesfalls abgeschlossen, die einzelnen Dokumente werden weiter bearbeitet. Zu Interface 3 liegen keine Dokumente vor.

Wie schon in Abschnitt 5.5 erwähnt, sind die fünf Schnittstellen nicht voneinander unabhängig. Kritisch ist die Definition des Interface 1, weil durch das dort festzulegende Workflow-Metaschema die grundlegende Sichtweise auf Workflows bestimmt wird, eine notwendige Voraussetzung für den Austausch von Prozeßdefinitionen zwischen unterschiedlichen Systemen. Das hat Einfluß auf alle anderen Schnittstellen. Der Entwurf einer „Workflow Process Definition Language“ (WPDL) ist aber noch nicht allgemein verfügbar. Die Standardisierung des Interface 1 muß in diesem Punkt möglichst schnell vorankommen.⁴³

⁴⁰ [WfMC-TC-1012], [WfMC-TC-1013]

⁴¹ [CORBA specification]

⁴² Stand : Juli 1998

⁴³ [Jablonski et al. 97], Seite 243 ff

Dokument	Nummer	Datum	Version
The Workflow Reference Model	WFMC-TC-1003	29-Nov-94	1.1
Terminology & Glossary	WFMC-TC-1011	June-1996	2.0
Interface 1 – Workflow Process Definition Read/Write Interface: Request For Comment	WFMC-WG01-1000	17-Feb-95	-
Interface 2 – Workflow Client Application: WAPI Specification	WFMC-TC-1009	Oct-97	2.0e (Beta)
Interface 2 – Workflow Client Application: WAPI Naming Conventions	WFMC-TC-1013	15-May-96	1.1
Interface 4 – Interoperability: Abstract Specification	WFMC-TC-1012	20-Oct-96	1.0
Interface 4 – Interoperability: Internet e-mail MIME Binding	WFMC-TC-1018	20-Oct-96	1.0
Interface 5 - Audit Data Specification	WFMC-TC-1015	1-Nov-96	1.0

Tabelle 3: Verfügbare Dokumente der WfMC

Bei allen Schnittstellen wird zunächst eine minimale Funktionalität standardisiert. Das ist angesichts der Unsicherheit über die weitere Entwicklung auch verständlich. Da die Hersteller aber trotzdem leistungsfähige Systeme anbieten wollen, ist zu befürchten, daß viele herstelllerspezifische Erweiterungen entstehen.

Seit der Gründung der WfMC sind mehr als vier Jahre vergangen. Der eigentlich vielversprechende Ansatz der WfMC, die wichtigsten Schnittstellen von Workflow-Management-Systemen herauszuarbeiten und zu standardisieren, ist noch längst nicht abgeschlossen. Möglicherweise liegt das an der Vielzahl ihrer Mitglieder, die alle ihre eigenen Interessen haben und die schnelle Entwicklung eines brauchbaren Standards diesen Interessen unterordnen. Es wird also noch einige Zeit dauern, bis der 7Standardisierungsprozeß abgeschlossen ist.

6 Gegenwärtiger Stand der Entwicklung

In diesem Kapitel soll anhand einiger konkreter Beispiele der gegenwärtige Entwicklungsstand von Workflowprodukten beschrieben werden. Eine vollständige Analyse der Produkte ist kein zentrales Anliegen dieser Arbeit. Herausgestellt werden die Eigenschaften der Produkte, die für die Interoperabilität mit anderen (Workflow-) Produkten von Bedeutung sind.

Eine umfassende Klassifikation für Workflow-Management-Systeme existiert derzeit nicht. Die Ansichten über die wesentlichen Eigenschaften, die als Grundlage dafür dienen können, sind sehr unterschiedlich. Deshalb werden zum Ende des Kapitels verschiedene Klassifikationsansätze vorgestellt, die auf ganz unterschiedliche Art eine Einordnung der Produkte vornehmen.

6.1 Beispiele für Workflow-Management-Systeme

Die hier vorgestellten Systeme stellen keine repräsentative Auswahl von Workflow-Management-Systemen dar. Im Allgemeinen ist es schwierig, an Informationen über ein Workflow-Management-System zu gelangen, die über Marketingaussagen hinausgehen. Die Produkte InterOffice und AdminFlow konnte ich im Rahmen meines Praktikums am Fraunhofer Institut für Arbeitswirtschaft und Organisation in Stuttgart untersuchen. Bei FlowMark handelt es sich um ein System, das vielen Forschungseinrichtungen zu Testzwecken zur Verfügung steht. Die Verschiedenheit dieser drei Produkte verdeutlicht recht gut die aktuelle Situation auf dem Markt der Workflow-Management-Systeme.

6.1.1 Oracle InterOffice 4.1

InterOffice⁴⁴ ist eine Groupware mit Nachrichtendienst, Verzeichnisdienst, Terminplaner und Dokumentenverwaltung, die auf einer Oracle7-Datenbank basiert. Ab der im Herbst 1997 erschienenen Version 4.1 beinhaltet das Produkt auch eine Workflowkomponente. Der InterOffice Server ist für die Plattformen HP-UX und Windows NT verfügbar. Als Benutzerschnittstelle stehen ein Windows95-Client und ein Webclient zur Auswahl.

⁴⁴ [InterOffice]

Die dem Produkt zugrundeliegende Datenbank enthält nicht nur die beim Einsatz des Produktes anfallenden Daten wie Nachrichtenobjekte, Dokumente und Workflowdefinitionen, das Produkt selbst ist zum Teil in der Datenbank enthalten. So besteht die Workflow-Engine aus *PL/SQL-Stored Procedures*, die im Package *WF_ENGINE* zusammengefaßt sind. InterOffice unterstützt vorwiegend Ad-Hoc-Workflows, wie sie typischerweise bei der Arbeit mit Groupwareprodukten entstehen, beispielsweise das Routen von Dokumenten zwischen verschiedenen Teilnehmern. Durch den Einsatz der Workflowkomponente werden auch stärker strukturierte Prozesse unterstützt. Abänderungen zur Laufzeit sind dann aber nicht möglich.

Die Prozeßdefinitionen werden mit dem *Oracle Workflow Builder* als Graphen modelliert. Die Schnittstelle zwischen dem Definitionstool und der Workflow-Engine ist proprietär. Die Knoten der Graphen stellen Aktivitäten oder Subprozesse dar. Es gibt zwei Arten von Aktivitäten: Notifikationsaktivitäten verschicken Nachrichten an Workflowteilnehmer, Funktionsaktivitäten führen eine PL/SQL-Prozedur in der Datenbank aus. Verzweigungsentscheidungen sind Bestandteile der Knoten. Die Aktivitäten oder Subprozesse liefern einen Ergebniswert, der die Fortsetzung des Prozesses im entsprechenden Ausgangspfad bewirkt. Diese Einschränkung hinsichtlich des Kontrollflusses läßt sich durch die Erstellung eigener PL/SQL-Prozeduren aufheben.⁴⁵ Ein mit dem Workflow Builder erstelltes Modell eines Beispielprozesses befindet sich in Anhang 3.

Das Produkt InterOffice ist für den Abschnitt 8 dieser Arbeit von besonderer Bedeutung. Deshalb werden hier einige Details ausführlich beschrieben, die für das Verständnis der dort beschriebenen Implementation nötig sind.

Die für die Unterstützung von Ad-Hoc-Workflows benötigten Aktivitäten, z. B. für das Versenden von Dokumenten, sind bereits im Produkt implementiert. Stärker strukturierte Vorgänge erfordern im Allgemeinen die Erstellung neuer Aktivitäten in der Form von *PL/SQL-Stored Procedures*. Dieses Verfahren ist aus verschiedenen Gründen bedenklich. Zum einen behindert das Fehlen selbst grundlegender Aktivitätstypen, wie zum Beispiel der in Abschnitt 8.1.2 beschriebenen Verzweigungsaktivität, die schnelle Einführung des Produkts in ein Unternehmen. Weiterhin ist der Inhalt der selbstdefinierten Aktivitäten nicht Bestandteil der Prozeßdefinition, obwohl die Aktivitäten in vollem Umfang auf die Workflow-Engine zugreifen können. Deshalb ist es im Allgemeinen nicht möglich, Prozeßdefinitionen auch nur zwischen verschiedenen Instanzen von InterOffice auszutauschen. Unter diesen Umständen ist die Implementation einer Schnittstelle zum Austausch von Prozeßdefinitionen mit anderen Produkten unmöglich.

⁴⁵ s. Abschnitt 8.1.2

Zu den vordefinierten Funktionen gehört das Versenden von HTML-basierten Formularen. Diese in einer Groupwarekomponente zweifellos sehr nützliche Funktion weist jedoch eine Einschränkung auf, die sie für den praktischen Einsatz ungeeignet erscheinen läßt: Es gibt nur eine Sicht auf ein Formular, d.h. alle Workflowteilnehmer haben die gleichen Zugriffsrechte auf alle Felder des Formulars. Umgehen läßt sich das nur, wenn man statt statischer HTML-Formulare CGI-Scripts verwendet.

Das ist wiederum nur durch Programmierung möglich, und zwar am besten durch die Verwendung des *Oracle Web Agent*, der in Verbindung mit dem *Oracle Webserver* die Ausführung von PL/SQL-Scripts ermöglicht, die HTML-Code erzeugen. Das Zusammenspiel der dabei beteiligten Komponenten ist in Abbildung 22 dargestellt. Der Client bekommt dadurch ein dynamisch erzeugtes Formular, das auf die Aktivitätsinstanz abgestimmt ist und unter anderem auch Zugriffsrechte auf die einzelnen Felder des Formulars ermöglicht.

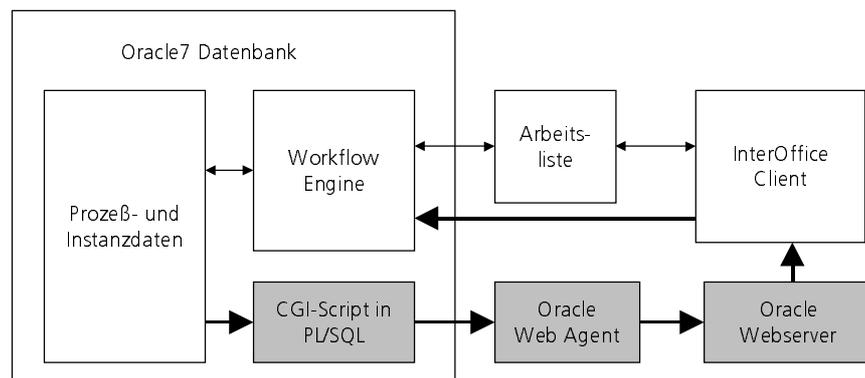


Abbildung 15: Dynamische Formulare mit InterOffice

Für die Lösung dieses doch recht komplexen Problems findet sich in der Dokumentation lediglich folgender Hinweis:

When you import an e-form template, you can import the form from a URL. If you select URL, you can specify the URL for a Web server program (such as a CGI script) that dynamically generates an e-form.⁴⁶

⁴⁶ InterOffice Client Documentation; Using Oracle InterOffice; Appendix B - Designing Workflows for InterOffice; Using Dynamic E-Forms

InterOffice bietet also keinerlei Unterstützung für die Erstellung der Formulare, so daß jeder Anwender seine eigene Vorstellung eines Formularmodells implementiert. Folglich wird der den Datenfluß beschreibende Informationsaspekt teilweise aus den Prozeßdefinitionen entfernt, was wiederum der Interoperabilität des Produkts bezüglich der Prozeßdefinitionen im Wege steht.

Die Workflowkomponente existierte zunächst nur als eigenständiges Produkt und wurde in InterOffice integriert. Beide Produkte bauen auf einer Oracle7-Datenbank auf. Der Grad der Integration ist aber nicht so hoch, wie man aufgrund dieser Tatsache erwarten könnte. So ist es beispielsweise Sache des Anwenders, die Mengen der Groupware-Nutzer und der Workflowteilnehmer aufeinander abzustimmen. Ein Rollenkonzept ist zwar vorgesehen, aber zumindest in der Windows NT-Version noch nicht umgesetzt. Insgesamt kann das Produkt, vor allem was die Workflow-Funktionalität betrifft, deshalb keinesfalls als ausgereift bezeichnet werden.

6.1.2 HP AdminFlow

Das Produkt AdminFlow⁴⁷ der Firma Hewlett Packard ist ein Workflow-Management-System, das auf den Betriebssystemen Windows NT und HP-UX läuft. Das Zusammenspiel der wichtigsten Systemkomponenten ist in Abbildung 16 dargestellt. Die Serverkomponenten können auf verschiedene Rechner verteilt werden. Die Workflow Engine ermittelt mit Hilfe der Prozeßdefinitionen die zu bearbeitenden Aktivitäten. Der Resource Executive bestimmt, welcher Workflowteilnehmer oder welche Ressource für die Ausführung der Aktivität zuständig ist. Dafür stehen ihm verschiedene Agenten zur Verfügung, die auf die entsprechenden Datenbestände zugreifen. Die Konfigurations- und Prozeßinformationen der Engine werden in einer Datenbank gespeichert. Zum Einsatz kommt dabei entweder eine Oracle7 Datenbank oder andere Produkte, die über eine ODBC-Schnittstelle verfügen.

Als Modellierungstool kommt das Produkt ProcessWise Workbench der Firma International Computers Ltd. zum Einsatz. Die Prozeßdefinitionen werden in einer Process Definition Language (PDL) abgespeichert und können anschließend von der Workflow Engine importiert werden. Die Prozesse werden als Graphen mit Knoten (Nodes) und Kanten (Arcs) modelliert. Die Arcs bestimmen den Kontrollfluß zwischen den Nodes. Es gibt zwei Typen von Nodes. Work Nodes enthalten Aktivitäten und besitzen immer genau je einen Eingangs- und einen Ausgangspfad. Route Nodes sind Verzweigungs- oder Synchronisationspunkte.

⁴⁷ [AdminFlow]

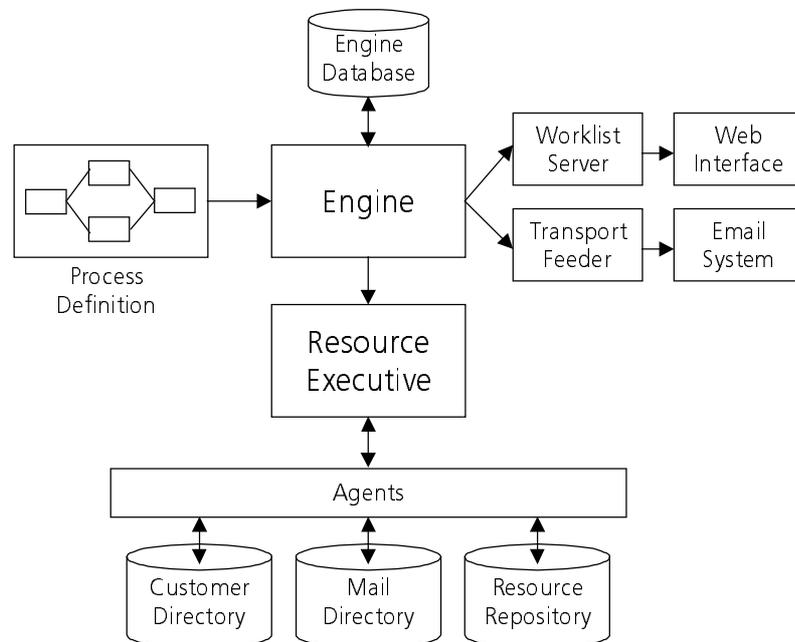


Abbildung 16. Komponenten von AdminFlow

Die zu einer Prozeßinstanz gehörenden workflowrelevanten Daten werden in einem Case Packet zusammengefaßt. Bei der Definition des Prozesses wird für jede Aktivität festgelegt, welche Daten bei der Bearbeitung gelesen und geschrieben werden können. Der Zugriff auf die Arbeitslisten erfolgt entweder über einen Browser oder über ein E-Mail-System. Die Interaktion der Workflowteilnehmer mit dem System erfolgt mit Hilfe von Formularen, die vom System aus den Daten des Case Packet erstellt werden. Änderungen am Ablauf von Prozessen zur Laufzeit sind nicht möglich.

Der Organisationsaspekt ist unabhängig von den Prozeßdefinitionen. Die Aktivitäten enthalten lediglich Verweise, die durch den Resource Executive mit Hilfe der Agenten interpretiert werden. Wenn im Unternehmen bereits ein Organisationsmodell vorliegt, kann es durch die Verwendung eines geeigneten Agenten genutzt werden. Änderungen am Organisationsmodell können so zur Laufzeit des Systems stattfinden und bleiben ohne Einfluß auf die Prozeßdefinitionen.

Die Interaktion mit externen Anwendungen erfolgt mittels Agenten. Die mitgelieferten Agenten erlauben lediglich Interaktionen mit den Benutzern. Das Produkt stellt damit eine Informationsinsel dar, die nicht mit den übrigen Ressourcen des Unternehmens verbunden ist. Dieser Mangel kann durch die

Erstellung zusätzlicher Agenten behoben werden. Dafür steht ein sogenanntes Resource Agent Software Development Kit zur Verfügung, eine C++ - basierte Programmierschnittstelle. Der Aufwand für die Implementierung von Schnittstellen ist dadurch sehr hoch.

6.1.3 IBM FlowMark

Das Produkt FlowMark⁴⁸ der Firma IBM ist ein verteiltes, objektorientiertes Client-Server-System, daß die Plattformen OS/2, Windows und AIX unterstützt. Persistente Informationen werden in einer objektorientierten Datenbank gespeichert. Die *Staff Definition Facility* beinhaltet eine hierarchische Organisationsstruktur von Personen und ihren Beziehungen untereinander. Das Produkt enthält eine *build time* Komponente, die die Modellierung von Prozessen unterstützt, sowie eine *run time* Komponente, die für die Ausführung von Prozeßinstanzen verantwortlich ist.

Der Client bietet eine graphische Benutzerschnittstelle, ruft lokale Programme auf und ermöglicht das Testen und Debuggen der Modelle. Der Server enthält die Datenbank mit den Prozeßinstanzen, steuert deren Ablauf und ordnet den Aktivitäten die Ressourcen zu.

FlowMark-Prozesse bestehen aus Aktivitäten, die weiter untergliedert sein können. In diesem Fall handelt es sich um Prozeßaktivitäten, anderenfalls um Programmaktivitäten. Programmaktivitäten verweisen mittels logischer Namen auf ausführbare Programme. Zu jeder Aktivität gehören unter anderem die folgenden Informationen:

- Startbedingungen
- Endbedingungen
- Zeitbeschränkung für die Ausführung der Aktivität
- benutztes Programm / benutzter Prozeß
- Eingangs- und Ausgangsdaten

Die Aktivitäten eines Workflows sind durch Kontroll- und Datenflußkonnektoren verbunden. Dadurch werden Kontrollfluß und Datenfluß getrennt modelliert. Aktivitäten können in Blöcken zusammengefaßt werden. Zu jedem Block kann eine Ausgangsbedingung angegeben werden. Die Aktivitäten des Blocks werden dann so oft ausgeführt, bis die Bedingung erfüllt ist.

⁴⁸ [FlowMark]

6.1.4 Weitere Workflowsysteme

Das System Workparty der Firma Siemens Nixdorf Informationssysteme AG unterstützt Prozesse auf der Grundlage des Prinzips der Umlaufmappe. Die Datenbank wird nur zum Protokollieren von Ereignissen verwendet.

Das System DOPAS⁴⁹ hebt sich von den anderen dadurch ab, daß neben dem Vorgangsmodell auch die Vorgangsbeschreibungen auf Schemaebene in der Datenbank gespeichert werden.

Für das Workflow-Management-System MOBILE⁵⁰ wurde die Workflow-Sprache MSL (Mobile Script Language) erstellt. Das Metaschema von MSL ist in die aus Abschnitt 4.1.1 bekannten Aspekte unterteilt. Die Sprachausdrücke sind entsprechend der Aspekte getrennt. Die Workflowschemata sind so aufgebaut, daß jeder Aspekt in einem eigenen Block des Schemas definiert wird. Schemadefinition und -verwendung sind dabei strikt getrennt, so daß einmal definierte Schemata beliebig wiederverwendet werden können.

Das Produkt Coordinator⁵¹ von Winograd und Flores ermöglicht es den Anwendern, Nachrichten bestimmter Typen zu versenden und dadurch mit anderen Anwendern zu kommunizieren. Reaktionen auf eingehende Nachrichten müssen mit dieser in einem sinnvollen Zusammenhang stehen. Das Produkt basiert auf der Sprechakttheorie von Austin⁵².

6.2 Workflow-Interoperabilität der vorgestellten Produkte

Eine standardisierte Schnittstelle im Sinne des Interface 4 der WfMC existiert bei keinem der vorgestellten Produkte. Deshalb sind Interaktionen mit anderen Workflow-Management-Systemen nicht ohne weiteres möglich. Die Produkte unterstützen aber den Operationsaspekt, der die Einbindung externer Anwendungen ermöglicht. Diese Mechanismen können grundsätzlich dazu verwendet werden, mit anderen Workflow-Management-Systemen zu interagieren. Dabei werden diese wie beliebige Anwendungen behandelt.

Der Aufwand der Entwicklung einer Interoperabilitätsschnittstelle ist vom jeweiligen System abhängig. In Kapitel 8 wird beispielhaft gezeigt, welche Schritte für die Implementation einer solchen Schnittstelle in das Produkt InterOffice nötig sind. Dabei legen spezielle Aktivitäten Informationen in der

⁴⁹ [Schulze/Böhm/Meyer-Wegener 94]

⁵⁰ [Jablonski/Bußler 96]; [MOBILE]

⁵¹ [Flores et al. 88]

⁵² [Austin 62]

Datenbank ab, auf die von den Routinen der Schnittstelle zugegriffen wird. Für zwei oder drei gegebene Systeme läßt sich dieser Ansatz noch weiter ausbauen. Das ist jedoch im Allgemeinen keine zufriedenstellende Lösung, da sie nicht standardisiert ist und für jede Kombination von Systemen angepaßt werden muß.

6.3 Klassifikationsansätze von Workflow-Management-Systemen

Die Erstellung einer umfassenden Klassifikation für Workflow-Management-Systeme ist schwer zu realisieren. Das liegt an den ganz unterschiedlichen Auffassungen darüber, was ein Workflow-Management-System ist und welche Eigenschaften eines solchen Systems als wichtig betrachtet werden. Im folgenden Abschnitt werden deshalb einige Ansätze vorgestellt, die jeweils auf der Grundlage einiger ausgesuchter Merkmale eine Klassifikation von Workflow-Management-Systemen vornehmen. Die Einordnung der drei in Abschnitt 6.1 vorgestellten Systeme wird vorgenommen, soweit das aufgrund der vorliegenden Informationen möglich ist. Oracle InterOffice ist kein reines Workflow-Management-System sondern eine Groupware, die unter anderem Workflow-Funktionalität bietet, und das auch erst ab der seit Ende 1997 verfügbaren Version 4.1. Trotzdem wurde der Versuch unternommen, das Produkt in die Schemata einzuordnen.

6.3.1 Klassifikation anhand des Workflow-Metaschema-Modells

Die folgenden wesentlichen Eigenschaften des Workflow-Metaschema-Modells können als Grundlage für eine Klassifikation von Workflow-Management-Systemen dienen⁵³:

- Anwendungsgebietsabhängigkeit: Sie ist vorhanden, wenn bestimmte Konzepte aus der Anwendungswelt vorgegeben sind, wie z.B. das Konzept der Umlaufmappe.
- Funktionalität: Die komplette Funktionalität sollte im Workflow-Metaschema enthalten sein. Sind einzelne Funktionen fest programmiert, können sie bei der Modellierung nicht geändert werden.
- Vollständigkeit: Ein Workflow-Metaschema sollte nicht eine vorgegebene Menge von Konstrukten festlegen, sondern dem Modellierer die Erstellung der benötigten Konstrukte ermöglichen.
- Wiederverwendung: Eine Workflow-Metaschema sollte die weitestmögliche Wiederverwendbarkeit aller Konstrukte ermöglichen. Beispielsweise sollten Prozeßdefinitionen sowohl als Top-Level-Prozeß als auch als Subprozeß verwendbar sein.

⁵³ [Jablonski et al. 97], Seite 124 ff

Die folgende Klassifikation zeigt für einige bekannte Workflow-Management-Systeme, wie sie bezüglich der angeführten Kriterien einzuordnen sind.

System	Anwendungs- gebiets- abhängigkeit	Funktionalität	Vollständigkeit	Wieder- verwendung
FlowMark	In der Organisationsbeschreibung wird z.B. das Konstrukt <i>employee</i> festgelegt, das eindeutig menschliche Benutzer beschreibt.	Teilweise fest vorgegeben. Beispielsweise wird der Prozeß-administrator bei Zeitüberschreitung benachrichtigt.	Konstrukte zur Beschreibung von Organisationen sind fest vorgegeben und nicht veränderbar.	Es ist beispielsweise nicht möglich, Kontrollflußkonstrukte zu definieren und wiederzuverwenden.
Work-Party	Das Konstrukt der Mappe bildet die wesentliche Grundlage.	Pro Akte gibt es unterschiedliche Operationen, die aber je Aktentyp festgelegt sind.	Kontrollflußkonstrukte sind fest vorgegeben.	Es ist beispielsweise nicht möglich, Kontrollflußkonstrukte zu definieren und wiederzuverwenden.
MOBILE	Es sind keine anwendungsspezifischen Festlegungen getroffen.	Das gesamte Verhalten eines Workflows muß in seinem Workflow-Schema festgelegt werden. Es sind keine Automatismen vorgesehen.	Alle Konstrukte müssen vor ihrer Verwendung definiert werden.	Alle Konstrukte, die in einem Workflow-Schema verwendet werden, sind vorher als eigene Typen bzw. Schemata in das System einzuführen.
Inter Office	Das Konstrukt <i>Nachricht</i> ist z.B. fest vorgegeben.	Das Verhalten der Aktivitäten wird im Programmcode festgelegt.	Der Datenfluß wird nicht explizit modelliert. Der Kontrollfluß ist fester Bestandteil der Aktivitäten.	Keine Wiederverwendung von Daten- und Kontrollflußkonstrukten.

Tabelle 4: Vergleich von Workflow-Metaschema-Modellen⁵⁴

⁵⁴ nach : [Jablonski et al. 97], Seite 126

6.3.2 Klassifikation bezüglich der Art der Datenbankverwendung⁵⁵

Dieser Ansatz geht von der Voraussetzung aus, daß Workflow-Management-Systeme Datenbanksysteme für die Datenhaltung einsetzen. Es mag Systeme geben, bei denen das nicht der Fall ist. Dennoch ist der Einsatz von Datenbanktechniken in Workflow-Management-Systemen naheliegend, was in Abschnitt 4.1.3.2 verdeutlicht wurde.

Als Kriterium wird der Zweck des Datenbankeinsatzes herangezogen. Für Vorgangsmodell und Vorgangstypbeschreibungen stellt sich jeweils die Frage, ob sie auf Objekt- oder Schemaebene repräsentiert werden. Die Systeme im 2. Quadranten sind auf eine begrenzte Anzahl von Vorgangstypen ausgerichtet. Das Vorgangsmodell befindet sich im Programmcode. Die Systeme des 3. Quadranten nutzen die Datenbank hauptsächlich als Protokollkomponente. Die Vorgangstypbeschreibungen befinden sich außerhalb der Datenbank. Die Systeme des 4. Quadranten besitzen ein Vorgangsmodell auf Schemaebene und können damit Vorgangstypbeschreibungen ebenfalls in der Datenbank speichern. Das im 1. Quadranten befindliche System DOPAS verwaltet auch Vorgangstypbeschreibungen auf Schemaebene. Der erhoffte Nutzen ist eine vorgangstypabhängige Integritätssicherung durch das Datenbanksystem und eine bessere Abstimmung von Speicherstrukturen auf vorgangstypspezifische Anforderungen.

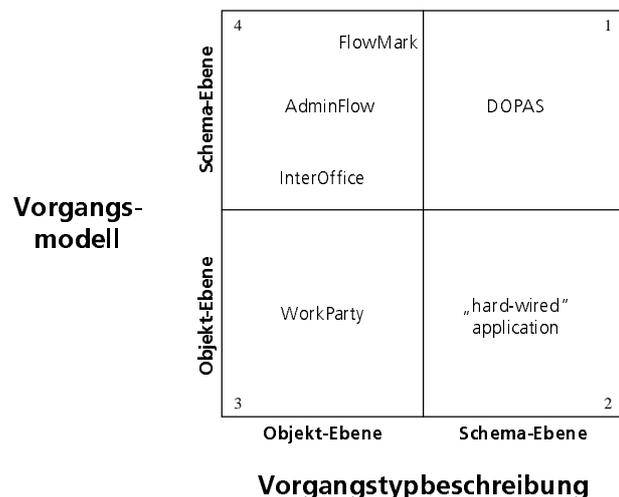


Abbildung 17: Workflow-Management-Systeme mit Datenbanken⁵⁶

⁵⁵ [Schulze/Böhm]

6.3.3 Einordnung nach Art der Vorgabebestimmung und Bezugsobjekt

Ein anderer Ansatz ordnet Koordinationssysteme anhand des sogenannten „Prinzips der Vorgabebestimmung“, also des Prinzips, nach dem die Vorgaben für die zu koordinierenden Abläufe bestimmt werden, sowie des Bezugsobjekts der Ablaufbeschreibung ein.⁵⁷ Zu den Koordinationssystemen gehören neben den Workflow-Management-Systemen auch andere Systeme, die Koordination nicht auf der Grundlage von Prozeßmodellen ermöglichen. So werden beispielsweise bei konversationsorientierten Systemen nicht die Vorgänge modelliert, sondern die Konversationsabläufe.

Bezüglich der Vorgabebestimmung werden standardisierungsorientierte, planungsorientierte und konversationsorientierte Ansätze unterschieden. Beim standardisierungsorientierten Ansatz werden die Vorgänge nach vereinheitlichten Regeln gesteuert. Dazu werden Vorgangstypen mittels formaler Beschreibungsverfahren wie z.B. Petrinetzen festgelegt. Der planungsorientierte Ansatz benutzt Methoden der künstlichen Intelligenz, um auf der Grundlage einer Wissensbasis über die Ablaufstrukturen des jeweiligen Anwendungsbereichs die Vorgänge einzelfallbezogen zu planen. Beim konversationsorientierten Ansatz werden die durchzuführenden Aufgaben anhand standardisierter Kommunikationsprotokolle direkt zwischen den beteiligten Stellen vereinbart. Die Methoden zur Beschreibung der Konversationen stammen aus der Sprechakttheorie.

Bezugsobjekte der Ablaufbeschreibung sind Prozesse, Stellen oder zu bearbeitende Objekte. Bei der prozeßbezogenen Ablaufbeschreibung werden Vorgänge explizit modelliert und koordiniert. Dazu werden sie in Teilvorgänge zerlegt, die in einem Bearbeitungsschritt erledigt werden können. Die stellenbezogene Ablaufbeschreibung gliedert die Abläufe anhand der Bearbeitungsstellen. Die Ablaufzusammenhänge werden dabei in der Regel durch Folgebeziehungen zwischen den Aufgaben angegeben. Die objekt- oder formularbezogene Ablaufbeschreibung versieht die Objekte mit Regeln für ihre Bearbeitung. Die Regeln geben meistens die Reihenfolge der Bearbeitungsschritte an. Für serielle Prozesse überschneiden sich formular- und prozeßbezogene Ablaufbeschreibung.

In der folgenden Tabelle findet sich die Einordnung einiger Systeme bezüglich der beiden ausgewählten Merkmale. Die Zuordnung ist nicht immer eindeutig; einem System können mehrere Ausprägungen desselben Merkmals zugeordnet sein.

⁵⁶ nach : [Schulze/Böhm], Seite 292

⁵⁷ [Wersch 95], Seite 104 ff

Koordinations- systeme	Prinzip der Vorgabebestimmung			Bezug der Ablaufbeschreibung		
	Standardi- sierungs- orientiert	Konversa- tions- orientiert	Planungs- orientiert	Prozeß- bezogen	Stellen- bezogen	Formular- bezogen
WorkParty	X			X		X
Coordinator		X			X	
InterOffice	X			X		
AdminFlow	X			X		
FlowMark	X			X		

Tabelle 5: Klassifikation von Workflow-Management-Systemen⁵⁸

6.4 Fazit

Dieses Kapitel gibt keinen umfassenden Überblick über die auf dem Markt befindlichen Workflow-Management-Systeme. Es zeigt aber an drei Beispielen die Heterogenität der Klasse der Workflowprodukte, die einen Vergleich der Produkte und damit eine Klassifikation erschwert. Die Systeme besitzen Schnittstellen, die Interoperabilität mit anderen Produkten ermöglichen. Diese sind aber nicht standardisiert, so daß in der Regel Anpassungen erforderlich sind.

⁵⁸ nach : [Wersch 95], Seite 106

7 Interoperabilität von Workflow-Management-Systemen

In diesem Kapitel werden zunächst verschiedene Interoperabilitätsstrategien von Software sowie eine Typologie zwischenbetrieblicher Anwendungen vorgestellt. Anschließend werden derzeit zur Verfügung stehende Technologien beschrieben, die Interoperabilität zwischen verschiedenen Workflow-Management-Systemen ermöglichen. Abschnitt 7.3 zeigt, wie das Internet von Workflow-Management-Systemen genutzt werden kann. Die Möglichkeiten, die sich durch den Einsatz von Middleware ergeben, werden in Abschnitt 7.4 am Beispiel von CORBA verdeutlicht.

7.1 Interoperabilitätsstrategien von Software

Der Wert einer Software richtet sich nicht nur nach ihrer Funktionalität. Es gibt eine Reihe anderer Faktoren, die dabei eine entscheidende Rolle spielen. Einer dieser Faktoren ist die Fähigkeit, mit anderer Software zu interagieren. Es gibt verschiedene Strategien, die solche Interaktionen ermöglichen.

Die verschiedenen Strategien haben unterschiedliche Stärken und Schwächen. Die **direkte Interaktion** bietet zwar prinzipiell die Möglichkeit, Daten effizient auszutauschen. Die Interaktion ist jedoch stark von den beteiligten Produkten abhängig, so daß keine Standardisierung erfolgen kann. Beim **Message Passing** besitzen beide Programme eine Schnittstelle, mit der sie Informationen senden und empfangen können. Beim **Bridging** kommunizieren beide Programme über ihre eigenen Protokolle mit einer Anwendung (Gateway), die in beiden Richtungen die notwendigen Anpassungen an das jeweils andere Protokoll vornimmt. Die **Nutzung gemeinsamen Speichers** unterscheidet sich von den anderen Ansätzen dadurch, daß die Interaktion nicht explizit erfolgt.

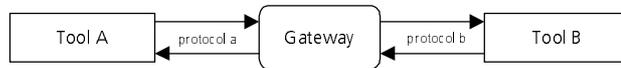
Direct interaction



Message passing



Bridging



Shared data store

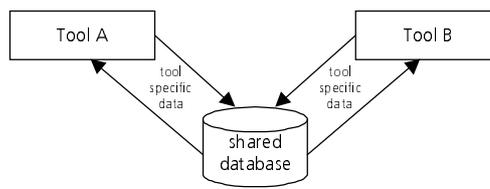


Abbildung 18: Interoperabilitätsstrategien von Software⁵⁹

7.2 Eine Typologie zwischenbetrieblicher Anwendungen

Einer an der zwischenbetrieblichen Geschäftstätigkeit orientierten Klassifikation verteilter Anwendungen⁶⁰ liegen die Kriterien

- Grad der Interaktivität
- Reichweite
- Automatisierungsgrad

zugrunde. Abbildung 19 zeigt die entstehenden Kategorien.

⁵⁹ [WfMC-TC-1012], Seite 7 f

⁶⁰ [Kurbel 97], Seite 25 ff

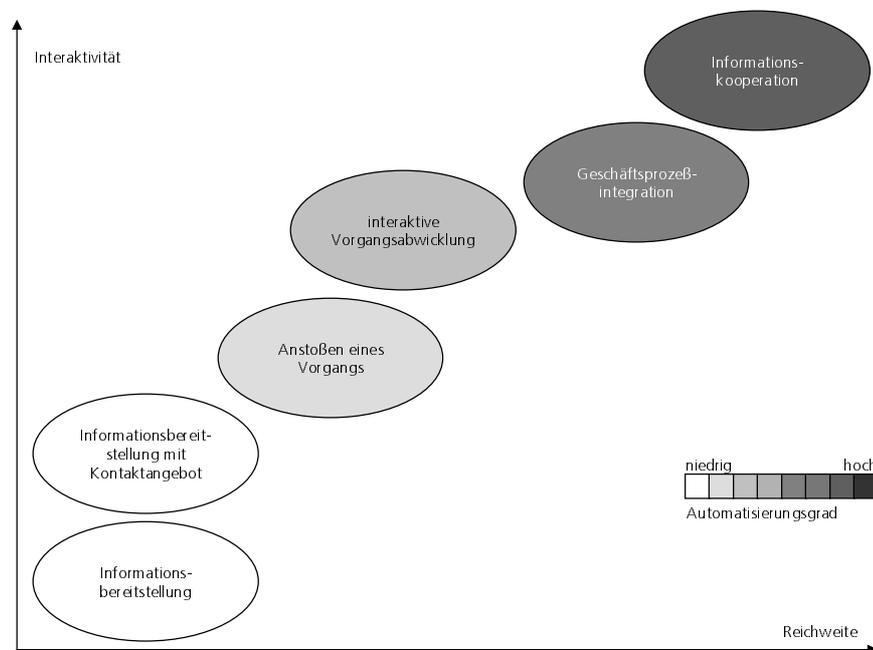


Abbildung 19: Typologie zwischenbetrieblicher Anwendungen⁶¹

Bei Anwendungen der Informationsbereitstellung bietet ein Unternehmen Informationen für beliebige Benutzer an. Es findet entweder keinerlei Informationsrückfluß statt, oder es gibt ein Kontaktangebot, so daß der Benutzer mit dem Anbieter über das System in Kontakt treten kann.

Das Anstoßen eines Vorgangs durch den Nutzer in dem Unternehmen, das die Informationen bereitstellt, kann entweder manuell oder automatisch geschehen, je nachdem wie die empfangenen Informationen verarbeitet werden.

Bei der interaktiven Vorgangsbewicklung werden die zwischenbetrieblichen Teilschritte des Geschäftsprozesses über das System abgewickelt. Geschäftsprozeßintegration bedeutet, daß auch innerbetriebliche Schritte, im Extremfall der ganze Geschäftsprozeß, unterstützt werden.

Bei der Informationskooperation schließlich tauschen mehrere Unternehmen geschäftsprozeßbezogene Informationen aus, um gemeinsame Ziele zu

⁶¹ [Kurbel 97], Seite 26

erreichen. Auch hier kann die Automatisierung unterschiedlich stark erfolgen. Unter Umständen sind an einem Kommunikationsprozeß mehrere Partner mit verschiedenen Automatisierungsgraden beteiligt.

Die Kategorien Anstoßen eines Vorgangs, interaktive Vorgangsabwicklung und Geschäftsprozeßintegration beziehen sich vorrangig auf die Unterstützung von Kommunikationsprozessen entlang einzelner Geschäftsprozesse. Anwendungen der Kategorie Informationskooperation unterstützen auch weitergehende Kooperationsformen, wie sie z.B. in virtuellen Unternehmen auftreten.

7.3 Nutzung von Internet-Technologien

Eine wesentliche Forderung an Workflow-Management-Systeme ist die Integration in bestehende Umgebungen durch die Nutzung der vorhandenen Kommunikationsinfrastruktur. Mit dem Internet steht seit einigen Jahren eine einheitliche Plattform zur Verfügung, die Workflow-Management-Systemen weltweit Verbindungen zu Nutzern und zu anderen Systemen ermöglicht.

Mit der Hypertext Markup Language existiert eine Sprache für die Informationsdarstellung, mit der unterschiedliche Medientypen in einem Dokument dargestellt werden können. Auch das Ausfüllen und Versenden von Formularen ist möglich. Zur Anzeige von HTML-Dokumenten dienen sogenannte Browser. Die Benutzung von Browsern für die Interaktion der Nutzer mit Workflow-Management-Systemen bringt folgende Vorteile:

- keine Eingewöhnung bezüglich der Oberfläche durch den Nutzer
- Plattformunabhängigkeit
- keine Administration der Clients
- hohe Erreichbarkeit der Nutzer

Der Einsatz von Browsern ist dabei für Administrations- und Monitoringaufgaben sowie als Client-Applikation möglich. Die Workflowteilnehmer können Prozesse starten und Aktivitätsinstanzen bearbeiten, indem sie das betreffende HTML-Formular ausfüllen und versenden.

Es gibt jedoch auch Nachteile dieses Ansatzes. So hat ein Nutzer, der mit verschiedenen Workflow-Management-Systemen interagiert, genau so viele verschiedene Arbeitslisten. Weiterhin werden die Interaktionen immer vom Nutzer initiiert, was nicht in allen Situationen ausreichend ist.

Als Alternative bietet sich das Versenden von E-Mails, die Formulare enthalten, an. Als Arbeitsliste dient dann der Posteingang, so daß zu einem Nutzer immer nur eine Arbeitsliste gehört. Sowohl der Nutzer als auch das System sind in der Lage, von sich aus Interaktionen auszuführen. Dieser Ansatz findet

beispielsweise im Produkt InterOffice (s. Abschnitt 6.1.1) Verwendung. Nachteilig wirkt sich dabei aus, daß sich eine Posteingangsliste nicht wie die Arbeitsliste eines Workflow-Management-Systems verhält. Abgearbeitete und ungültige Aktivitäten bleiben in der Liste, bis sie der Nutzer entfernt. Dies macht sich zum Beispiel dann nachteilig bemerkbar, wenn mehrere Workflowteilnehmer aufgrund ihrer Rollen dieselbe Aufgabe angeboten bekommen.

Bisher wurden Interaktionen der Nutzer mit den Systemen betrachtet. Aber auch die Interaktion zwischen verschiedenen Workflow-Management-Systemen ist mittels Formularen möglich. Ein Beispiel dafür ist das System Panta Rhei.⁶² In den Prozeßdefinitionen wird festgelegt, welche Aktivitäten welche Formulare versenden und wie die Prozesse auf den Empfang bestimmter Formulare reagieren. Die Schnittstelle 4 der Referenzarchitektur der WfMC wird also dadurch beschrieben, welche HTML-Formulare ein System empfangen kann, welche Prozesse dadurch gestartet werden und welche Formulare als Antwort zurückgeschickt werden. Dadurch ist es möglich, systemübergreifend Prozesse durchzuführen. Die Interaktionen zwischen beiden Systemen beschränken sich aber auf den Formularinhalt, der Austausch weiterer Informationen, z.B. bezüglich der Steuerung von Workflows, ist nicht standardisiert.

Das Internet stellt Technologien zur Verfügung, die Interaktionen zwischen Nutzern und Systemen grundsätzlich ermöglichen. Die semantische Heterogenität der Produkte setzt den Interaktionsmöglichkeiten jedoch enge Grenzen. Abhilfe kann hier der Einsatz von Middleware schaffen, die auf dieser Basis die benötigten Mechanismen zur Überwindung der Grenzen bereitstellt.

7.4 Workflow-Management und Middleware

Die in Abschnitt 4 vorgestellte allgemeine Architektur für Workflow-Management-Systeme zeigt, daß der Einsatz von Middleware für den Aufbau leistungsfähiger Systeme erforderlich ist. In diesem Abschnitt wird am Beispiel der Object Management Architecture (OMA) der OMG gezeigt, wie Interoperabilität von Workflow-Management-Systemen durch den Einsatz standardisierter Middleware erreicht werden kann. Von besonderem Interesse ist dabei die Workflow Management Facility, die der Workflowtechnologie einen festen Platz innerhalb der OMA zuweisen soll. Dabei ergeben sich interessante Zusammenhänge mit der Arbeit der Workflow Management Coalition, die sich am Standardisierungsprozeß der Workflow Management Facility beteiligt.

⁶² [Eder et al.]

7.4.1 Die Object Management Group

Die Object Management Group (OMG) ist eine internationale Organisation mit über 800 Mitgliedern, darunter Systemanbieter, Softwareentwickler und Anwender. Sie wurde 1989 mit dem Ziel der Verbreitung der objektorientierten Technologie gegründet. Die OMG will dadurch die Wiederverwendung, Portierbarkeit und Interoperabilität objektbasierter Software in verteilten, heterogenen Umgebungen fördern. Zu diesem Zweck wurde die Object Management Architecture (OMA) geschaffen, die als Referenzarchitektur für verteilte Anwendungen dient. Die OMG will damit einen umfassenden Architekturrahmen mit standardisierten Schnittstellen für fast alle Bereiche der EDV schaffen.

7.4.2 CORBA

Der **Object Request Broker** (ORB) und seine Schnittstellen bilden den zentralen Bestandteil der OMA, der als **Common Object Request Broker Architecture** (CORBA) bezeichnet wird.

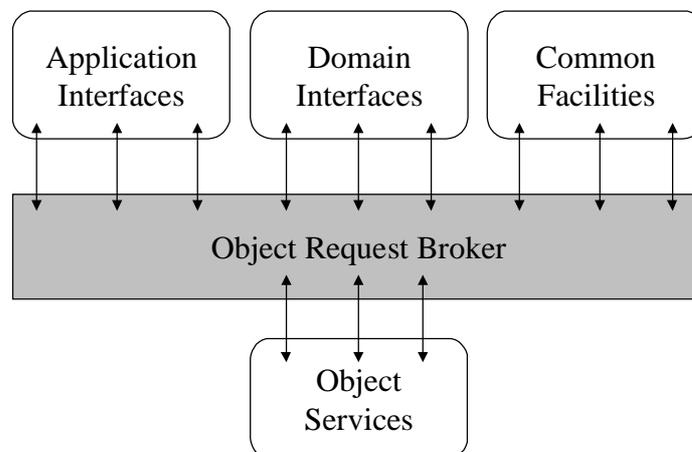


Abbildung 20: Die Referenzarchitektur OMA

Der ORB stellt Mechanismen für die Interaktion zwischen Objekten bereit und sorgt dadurch für Transparenz bezüglich der Verteilung der Objekte. Die **Object Services** stellen grundlegende Funktionen bezüglich des Einsatzes von Objekten zur Verfügung. Die **Common Facilities** bieten domänenunabhängige Dienste, die von vielen Anwendungen benötigt werden. Die **Domain Interfaces** dagegen fassen domänenspezifische Dienste zusammen. Sie werden in älteren Dokumenten der WfMC als Vertical Market Facilities

bezeichnet. **Application Interfaces** werden von der OMG nicht standardisiert, weil sie sich auf konkrete Anwendungen beziehen.

Die vollständige Beschreibung der Architektur findet sich unter <http://www.omg.org>. In den folgenden Abschnitten werden die für das Verständnis der Arbeit wichtigen Begriffe und Konzepte kurz vorgestellt.

7.4.2.1 Der Object Request Broker

Ein ORB vermittelt Methodenaufrufe zwischen Objekten. Das aufrufende Objekt wird als **Client** bezeichnet, das aufgerufene Objekt als **Server**. Welche Operationen des Servers der Client mit Hilfe des ORBs ausführen kann, wird durch eine Schnittstellendefinition bestimmt. Diese Definitionen werden in der **Interface Definition Language (IDL)** beschrieben. Ein IDL-Compiler übersetzt die IDL-Definitionen in die entsprechenden Konstrukte der von Client bzw. Server benutzten Programmiersprache. Die Implementierung kann in nahezu jeder Programmiersprache, auch in einer nicht-objektorientierten, erfolgen. Dabei können Client und Server verschiedene Sprachen benutzen. Bei der in Abbildung 21 dargestellten einfachsten Form der Interaktion werden Client-Aufrufe durch **IDL Stubs** an den ORB und von dort durch **IDL Skeletons** an den Server vermittelt. Das bleibt aber für den Client transparent: Er unterscheidet nicht zwischen lokalen Aufrufen und Aufrufen an IDL Stubs.

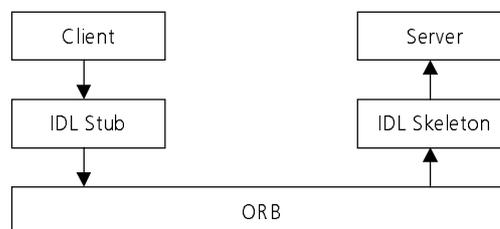


Abbildung 21: Interaktion zwischen Objekten mittels ORB

Eine CORBA-basierte Umgebung ist nicht auf die Verwendung eines einzelnen ORBs beschränkt. Die Objekte können über verschiedene ORBs miteinander in Verbindung stehen. Das abstrakte **General Inter-ORB Protocol (GIOP)** standardisiert die Interaktionen zwischen den ORBs. Es kann auf verschiedenen Transportmechanismen implementiert werden. Das **Internet Inter-ORB Protocol (IIOP)** spezifiziert den Austausch von GIOP-Botschaften über TCP/IP-Verbindungen. Es ist das Standardprotokoll von CORBA-Implementationen für das Internet.

7.4.2.2 Object Services

Die Object Services sind eine Sammlung von Schnittstellen und Objekten, die grundlegende Funktionen für die Benutzung und Implementation von Objekten bieten. Sie werden für die Erstellung jeglicher verteilter Anwendungen benötigt und sind unabhängig von Anwendungsgebieten. Die Services sind bezüglich ihrer Funktionalität orthogonal und können miteinander kombiniert werden. Zu folgenden Object Services existieren derzeit Spezifikationen, die in den Dokumenten der OMG enthalten sind:⁶³

Naming Service	bietet die Möglichkeit, ein Objekt an einen Namen zu binden
Event Service	unterstützt asynchrone Operationen
Persistent Object Service	bietet Mechanismen für die Verwaltung des persistenten Status von Objekten
Life Cycle Service	enthält Dienste zum Erzeugen, Löschen, Kopieren und Verschieben von Objekten
Concurrency Control Service	ermöglicht die Koordination des Zugriffs auf geteilte Ressourcen
Externalization Service	definiert Operationen zur Umwandlung von Objekten in Datenströme und zurück
Relationship Service	ermöglicht die explizite Repräsentation von Einheiten und Relationen
Transaction Service	unterstützt verschiedene Transaktionsmodelle
Query Service	ermöglicht Anfragen an Objektmengen
Licensing Service	bietet Mechanismen für die Kontrolle der Verwendung von Objekten
Property Service	erlaubt die dynamische Zuordnung von Eigenschaften zu Objekten

⁶³ [CORBAservices]

Time Service	ermöglicht zeitbasierte Operationen in verteilten Umgebungen
Security Service	definiert verschiedene Sicherheitsmechanismen
Trading Service	ermöglicht das Auffinden von Objekten, die bestimmte Dienste anbieten
Collections Service	bietet Mechanismen zur Erzeugung und Manipulation von Kollektionen von Objekten

Tabelle 6: CORBA Services

7.4.2.3 Common Facilities und Domain Interfaces

Common Facilities und Domain Interfaces füllen die Lücke zwischen der durch CORBA und die Object Services definierten Technologie und den anwendungsspezifischen (und damit unstandardisierten) Objekten. Die OMG hat einige Bereiche identifiziert, deren Standardisierung als notwendig angesehen wird. Die Spezifikation der entsprechenden Facilities und Domain Interfaces liegt aber in den meisten Fällen noch nicht vor.

Die Common Facilities werden von vielen Anwendungen benötigt, unabhängig von ihrer Domäne. Die OMG ordnet sie in vier Bereiche ein:

User Interface	
Rendering Management	ermöglicht die Präsentation von Objekten
Compound Presentation Management	unterstützt die Präsentation von Objekten in Verbunddokumenten
User Support Facilities	bietet Mechanismen für die Speicherung und Anzeige von Hilfsinformationen
Desktop Management	enthält Dienste für den Desktop der Nutzer
Scripting	unterstützt die interaktive Erzeugung von Scripts

Task Management	
Workflow Management	ermöglichte Management und Koordination von Objekten, die Teil von Arbeitsabläufen sind
Agent	unterstützt statische und dynamische Agenten
Rule Management	unterstützt den Wissenserwerb und die Ausführung regelbasierter Methoden
Automation	ermöglicht die objektübergreifende Nutzung bestimmter Funktionen

Information Management	
Information Modeling	unterstützt die Erzeugung von Informationsmodellen
Information Storage and Retrieval	unterstützt die Speicherung und das Wiederauffinden von Informationen
Compound Interchange	unterstützt den Austausch von Daten in Verbunddokumenten
Data Interchange	unterstützt den allgemeinen Datenaustausch
Information Exchange	unterstützt den Informationsaustausch
Data Encoding and Representation	unterstützt das Kodieren und Übersetzen von Daten
Time Operations	unterstützt die Manipulation von Kalender- und Zeitangaben

System Management	
Management Tools	unterstützt die Interoperabilität von Managementtools
Collection Management	unterstützt die Integration von Kollektionen
Control	unterstützt die Steuerung von Systemressourcen

Tabelle 7: CORBA Facilities

Domain Interfaces, in älteren OMG-Dokumenten⁶⁴ auch als Vertical Market Facilities bezeichnet, sollen bestimmte Anwendungsgebiete unterstützen. Die Standardisierung der Domain Interfaces steht erst am Anfang. Es bestehen verschiedene Arbeitsgruppen (Task Forces), die sich jeweils mit einem bestimmten Bereich beschäftigen:

Business Object Domain Task Force	Identifikation und Beschreibung von Common Business Objects
Electronic Commerce Domain Task Force	Verkauf und Copyright elektronischer Inhalte, Bezahlung mit virtueller Währung
Finance Domain Task Force	Abwicklung von Finanztransaktionen
CORBAmed Domain Task Force	Identifikation und Beschreibung von medizinischen Objekten
Manufacturing Domain Task Force	Anforderungen an Software in Produktionsumgebungen
Telecommunications Domain Task Force	Schnittstellen für Systeme im Telekommunikationsbereich

Tabelle 8: Domain Interface Task Forces

⁶⁴ [CORBAfacilities]

7.4.3 Die Workflow Management Facility

Unter den Common Facilities befindet sich eine Workflow Management Facility, mit der die OMG Unterstützung für Workflow-Management-Anwendungen in ihre Architektur einbringen will. Die Spezifikation der Workflow Management Facility liegt noch nicht vor. Im Mai 1997 erschien ein Request For Proposals (RFP), der die Softwareindustrie aufforderte, ihre Vorschläge für die Gestaltung der Workflow Management Facility einzureichen. Die obligatorischen und optionalen Anforderungen sowie die Diskussionspunkte des RFP sind in diesem Abschnitt aufgeführt.

7.4.3.1 Obligatorische Anforderungen

Die folgenden Anforderungen an die Workflow Management Facility werden von der OMG als grundlegend betrachtet und müssen in jedem Vorschlag enthalten sein.

- Das **Workflow-Metamodell** wird in dieser Arbeit als Workflow-Metaschema bezeichnet und ist in Abschnitt 3.1.1 definiert. Der RFP fordert ausdrücklich eine vollständige Semantik des Metaschemas.
- Schnittstellen für das **Workflow Enactment**, also die Manipulation von Workflowobjekten, insbesondere von Workflowinstanzen.
- Schnittstellen für das **Workflow Monitoring**, die Abfragen über den Status von Instanzen ermöglichen.
- **Workflow Audit Trail** – Schnittstellen sollen die Erstellung einer Historie ermöglichen.
- **Nesting of Workflows** – Schnittstellen sollen den Zugriff auf verschachtelte Workflows ermöglichen.

7.4.3.2 Optionale Anforderungen

Optionale Anforderungen sind seitens der OMG gewünscht und werden in den Evaluationsprozeß einbezogen, müssen aber nicht unbedingt in einem Vorschlag enthalten sein.

- Schnittstellen für die Erzeugung, das Wiederauffinden und die Wartung von Workflowschemata.
- Die Wiederverwendung von Prozeßdefinitionen soll unterstützt werden.
- Die Unterstützung von Ad-Hoc-Workflows soll durch die Modifikation des Verhaltens aktiver Instanzen unterstützt werden.
- Es werden Techniken benötigt, die Serviceparameter für die Ausführung von Workflows ermöglichen, so daß z.B. Ressourcen optimal genutzt werden können.

7.4.3.3 Diskussionspunkte

Wie jede komplexe Technologie kann Workflow-Management nicht in einem Schritt umfassend standardisiert werden. Die folgenden Punkte müssen deshalb nicht in den Vorschlägen enthalten sein, ihre spätere Integration in die Workflow Management Facility soll jedoch angedacht werden.

- Beziehungen zum und Abhängigkeiten vom Transaction Service
- Sicherstellung der Integrität von Workflows durch Recovery- und Rollback-Mechanismen
- Archivierung von Workflowinstanzen
- Auswahl von Ressourcen
- Skalierbarkeit und Ressourcenmanagement
- Sicherheitsmechanismen
- Organisationsübergreifende Ausführung von Workflows

7.4.4 Workflow-Management und die OMG

Die Workflow Management Facility ist eine optionale Komponente der OMA. Sie nimmt also keine herausragende Rolle ein, sondern unterliegt denselben Bedingungen wie alle anderen Facilities. Das bedeutet unter anderem, daß die Facility auf die zentralen Aspekte des Workflow-Managements beschränkt wird und mit anderen Facilities und Object Services in Zusammenhang steht. Ein Beispiel dafür ist der Transaction Service, der ein erweiterbares Transaktionskonzept vorsieht, das unter anderem verschachtelte Transaktionen unterstützt. Das ermöglicht einer CORBA-Implementation Modifikationen des klassischen Datenbank-Transaktionskonzepts, die den in Abschnitt 4.1.3.2 genannten Eigenschaften von Workflows Rechnung tragen. Ein weiteres Beispiel ist die Business Object Facility (BOF), deren Spezifikation allerdings noch nicht vorliegt. Abhängigkeiten zwischen Workflows und Geschäftsobjekten bestehen in beiden Richtungen. Deshalb ist die Kopplung zwischen beiden Facilities sehr eng. Es bestehen weitere Zusammenhänge zu anderen Object Services und Facilities, die hier aus Platzgründen nicht erwähnt werden können.⁶⁵

Die Workflow Management Facility der OMG bietet der WfMC die Möglichkeit, ihren Standard in die allgemeinere Architektur OMA einzubringen. Allerdings erfüllt die Architektur der WfMC wesentliche Anforderungen des betreffenden RFP nicht, da sie sich auf Aspekte der Ausführung von Workflows konzentriert und kein Workflow-Metaschema vorzuweisen hat. Das Metaschema gehört

⁶⁵ [Jablonski et al. 97], Seite 284 ff

jedoch zu den obligatorischen Anforderungen an die Workflow Management Facility.

Der OMG liegen drei Vorschläge für die Spezifikation der Workflow Management Facility vor. Der „jFlow“ - Vorschlag der WfMC wurde gemeinsam von 19 Firmen eingereicht, da die WfMC nach dem Reglement der OMG keine Vorschläge einreichen darf. Die anderen beiden Vorschläge stammen von der Firma NORTEL bzw. der Firma EDS. Keiner der drei Vorschläge allein ist für die Spezifikation der Workflow Management Facility ausreichend.⁶⁶ Wünschenswert wäre ein überarbeiteter Vorschlag, der die besten Elemente in sich vereinigt. Als Ergebnis ist eine Spezifikation zu erwarten, die einen Ausgangspunkt für die weitere Standardisierung des Bereichs Workflow-Management auf der Basis der OMA darstellt. Möglicherweise ergeben sich Rückschlüsse für die Architektur der WfMC, so daß sich beide Standardisierungsprozesse aneinander annähern.

⁶⁶ [Schulze/Bussler/Meyer-Wegener 98]

8 Workflow-Interoperabilität – ein Beispiel

In einem Praktikum am Fraunhofer Institut für Arbeitswirtschaft und Organisation in Stuttgart wurde untersucht, ob sich das Produkt InterOffice für den Einsatz in verteilten Umgebungen eignet. Dazu wurde ein Beispielprozeß erstellt. Weiterhin wurde untersucht, inwieweit die Implementation einer Interoperabilitätsschnittstelle für InterOffice möglich ist. Abschnitt 8.1 beschreibt das Projekt und den Beispielprozeß. In Abschnitt 8.2 wird der Aufbau der Interoperabilitätsschnittstelle erklärt. In Abschnitt 8.3 wird das Beispielprojekt bezüglich der erreichten Interoperabilität eingeordnet. Abschnitt 9 faßt die Ergebnisse zusammen.

8.1 Ein Beispielszenario: Das Projekt PoliFlow

PoliFlow ist eines von vier vom BMBF geförderten Projekte der Förderinitiative POLIKOM. Im Zentrum der Forschungsarbeiten von POLIKOM stehen Fragen zum Organisationsaspekt von Verwaltungen. Eine wichtige Aufgabe des Projekts PoliFlow ist die Integration von Workflow-, Telekooperations- und Multimediatechniken. Neben einer leistungsfähigen Steuerung der Arbeitsabläufe soll durch diese Integration die kooperative Bearbeitung standort- und organisationsübergreifender Aufgaben unterstützt werden. Nähere Informationen zum Projekt PoliFlow finden sich unter <http://poliflow/hewlett-packard.de/index.html>.

Im Rahmen des Projektes PoliFlow bestand die Aufgabe, am Fraunhofer Institut für Arbeitswirtschaft und Organisation in Stuttgart den Vorgang „Auftragsvergabe“ mit dem Produkt Oracle InterOffice zu implementieren, um die Eignung des Produktes bezüglich des geplanten Einsatzes in Verwaltungen zu prüfen. Bisher läuft dieser Vorgang weitgehend ohne die Unterstützung elektronischer Systeme ab.

8.1.1 Der Prozeß „Auftragsvergabe“

Beim Ablauf des Vorgangs ‚Auftragsvergabe‘ wird eine zusammengehörende Menge von Dokumenten nacheinander von verschiedenen Mitarbeitern mit unterschiedlichen, genau festgelegten Aufgaben und Rechten bearbeitet. Ergebnis der Bearbeitung sind Einträge in einem zu den Dokumenten gehörenden Formular. Es handelt sich um einen Vorgang, dessen Aktivitäten einzeln nacheinander ausgeführt werden. Der Vorgang sollte so implementiert werden, daß möglichst keine Veränderungen gegenüber dem bisherigen

Ablauf auftreten. Eine grafische Darstellung des Prozeßmodells findet sich in Anhang 3.

An dem Vorgang sind zwei Behörden beteiligt: ein Bundesbauamt und die Oberfinanzdirektion (OFD) Berlin. Der Vorgang läuft weitestgehend im Bauamt ab. In einigen Fällen ist eine Unterlagenprüfung durch die Oberfinanzdirektion notwendig. Ob die OFD eine Unterlagenprüfung durchführen muß, wird durch den Bearbeiter der Nachrichtenaktivität mit der Bezeichnung ‚Start‘ entschieden, der in das Feld 1 des Formulars entweder ‚Ja‘ oder ‚Nein‘ einträgt. Die entsprechende Verzweigungsaktivität schickt die Unterlagen dann entweder zur Prüfung an die Abteilung V3 der OFD oder direkt zum Bearbeiter O1 der Aktivität ‚Mitzeichnen‘.

Die der OFD zugeordneten Aktivitäten stellen einen zusammenhängenden Abschnitt des gesamten Prozesses dar. Deshalb bietet es sich an, sie in einem Subprozeß zusammenzufassen. Grundsätzlich ist mit InterOffice die Modellierung von Subprozessen möglich. Das aufgrund der fehlenden Flexibilität der Standardformulare (s. Abschnitt 6.1.1) implementierte Formularmodell (s. Abschnitt 8.1.2) ist dafür aber nicht geeignet, so daß der gesamte Vorgang als ein Prozeß implementiert werden mußte.

Wie man Abschnitt 8.3.1 entnehmen kann, würde die Zusammenfassung der OFD-Aktivitäten in einem Subprozeß prinzipiell die Aufteilung des Vorgangs auf zwei Workflow-Management-Systeme ermöglichen. So hingegen müssen alle Aktivitäten auch innerhalb eines Systems ausgeführt werden. Da zwei verschiedene Organisationen (in diesem Fall Behörden) in der Regel nicht dieselbe Instanz eines Workflow-Management-Systems benutzen, ist die Demonstration organisationsübergreifender Prozesse am Beispiel der Zusammenarbeit von Bauamt und OFD hier nicht möglich.

8.1.2 Implementation der benötigten Aktivitätstypen

Da InterOffice keine geeigneten vordefinierten Aktivitätstypen besitzt, mußten diese bei der Implementierung des Beispielprozesses erstellt werden. Da ihre Funktionsweise für das Verständnis der Implementation notwendig ist, werden sie hier kurz beschrieben.

Der Beispielprozess besitzt eine Start- und eine Endaktivität, deren Funktionalität ausschließlich darin besteht, Start- und Endpunkt des Prozesses festzulegen. Die Aktivitäten vom Typ ‚Noop‘ dienen lediglich der besseren Übersicht des Ablaufdiagramms und haben für den Prozeß keine Bedeutung.

Die Aktivitäten vom Typ WF_HUBAU_N sind Nachrichtenaktivitäten. Wird eine solche Aktivität ausgeführt, erzeugt sie zunächst einen Eintrag in einer Tabelle der dem System zugrundeliegenden Datenbank und wartet anschließend auf

die Bearbeitung des Eintrags durch den entsprechenden Mitarbeiter. Jeder Mitarbeiter hat eine Arbeitsliste, in der die ihm zugewiesenen Einträge stehen. Nach der Bearbeitung eines Eintrags wird dieser aus der Tabelle entfernt und die zugehörige Aktivität beendet. Daraufhin wird der Prozeß fortgesetzt.

Die Aktivitäten vom Typ OR_SPLIT sind Verzweigungsaktivitäten, die ein zur Instanz gehörendes Attribut als Grundlage für die Verzweigungsentscheidung heranziehen. Dabei wird der Prozeß in genau einem von zwei möglichen Pfaden fortgesetzt. Für die Modellierung des Beispielprozesses ist diese Art der Verzweigung ausreichend, da zu einem gegebenen Zeitpunkt immer nur ein Pfad durchlaufen wird.

Die beiden genannten Aktivitätstypen benutzen Instanzattribute, die bei der Prozeßmodellierung angelegt wurden. Bei der Bearbeitung von Nachrichtenaktivitäten werden die Attribute ausgelesen, um dem Bearbeiter die zur Instanz gehörenden Informationen in einem Formular zu präsentieren und nach der Bearbeitung die Änderungen abzuspeichern. Die Verzweigungsaktivitäten lesen die Attribute nur aus und verzweigen entsprechend der aktuellen Werte. Der Grund für die Implementierung dieses Formularmodells kann in Abschnitt 6.1.1 nachgelesen werden.

Während der Ausführung einer Instanz können semantische Fehler auftreten. Dann wird zur Fehleraktivität des Prozesses verzweigt. Diese ist eine besondere Nachrichtenaktivität, die durch den Initiator des Prozesses bearbeitet wird. Einige Fehler sind im Prozeßmodell berücksichtigt, so daß eine Verzweigungsaktivität die Fehlerbehandlung einleitet. Außerdem hat jeder Bearbeiter einer Nachrichtenaktivität die Möglichkeit, unvorhergesehene Fehler zu melden, was ebenfalls zur Ausführung der Fehleraktivität führt.

8.2 Workflow-Interoperabilität mit InterOffice

Die Workflow-Engine von InterOffice verfügt über eine gut dokumentierte Schnittstelle. Deren Routinen werden von anderen Komponenten des Produkts InterOffice zur Steuerung der Engine benutzt. Es ist aber auch möglich, die Routinen durch beliebige andere Programme aufzurufen. Die Schnittstelle genügt jedoch nicht den Vorgaben der WfMC bezüglich Interface 4. Deshalb wurde eine CORBA-basierte, an den Vorgaben der WfMC orientierte Schnittstelle implementiert. Diese Implementation ermöglicht es anderen Anwendungen, über eine standardisierte Schnittstelle die Workflow-Engine von InterOffice anzusteuern.

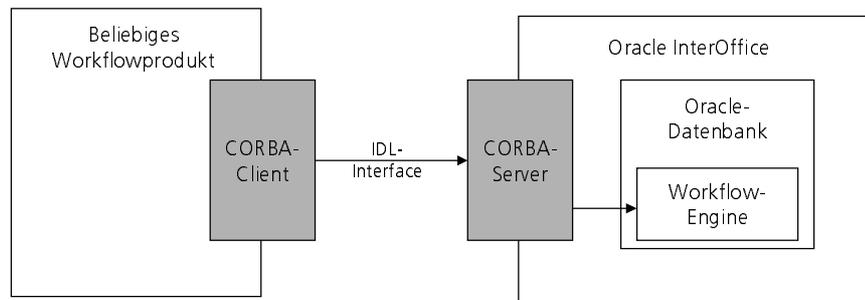


Abbildung 22: Workflow-Interoperabilität mit InterOffice

8.2.1 Das IDL-Interface

In den Unterlagen der WfMC finden sich Konventionen für Bezeichner, die bei der Entwicklung von WfMC-konformen Anwendungen eingehalten werden sollen⁶⁷, sowie die abstrakte Beschreibung einer Schnittstelle mit Routinen zur Prozeßsteuerung.⁶⁸ Diese Beschreibung bedient sich der zur Architektur CORBA gehörenden Interface Definition Language. Auf der Grundlage dieser Vorgaben wurde ein IDL-Interface entwickelt, das in Anhang 1 dieser Arbeit enthalten ist. Eine Übersicht über die Routinen des Interfaces gibt Tabelle 9.

Die Implementierung des IDL-Interfaces erfolgte mit dem Produkt Orbixweb. Dieses Produkt ist eine Implementation der CORBA-Spezifikation der OMG, die die Funktionalität von CORBA mittels der Programmiersprache Java nutzbar macht. Der zugehörige IDL-Compiler erzeugt alle für die Entwicklung von Client- und Serverprogrammen notwendigen Java-Klassen. Die verwendete Version 3.0 unterstützt dabei das standardisierte IDL-Java-Mapping der CORBA-Spezifikation 2.2, die im Februar 1998 veröffentlicht wurde.

⁶⁷ [WfMC-TC-1013]

⁶⁸ [WfMC-TC-1012]

WMRequestCreateProcessInstance	Erzeugen einer Prozeßinstanz
WMRequestGetProcessInstanceState	Ermitteln des Status einer Prozeßinstanz
WMRequestSetProcessInstanceAttributes	Setzen von Attributen einer Prozeßinstanz
WMRequestGetProcessInstanceAttributes	Lesen von Attributen einer Prozeßinstanz
WMRequestStartProcessInstance	Prozeßinstanz starten
WMNotifyProcessInstanceStarted	Nachricht: Prozeßinstanz gestartet
WMRequestAbortProcessInstance	Prozeßinstanz abbrechen
WMNotifyProcessInstanceAborted	Nachricht: Prozeßinstanz abgebrochen
WMRequestTerminateProcessInstance	Prozeßinstanz abnormal beenden
WMNotifyProcessInstanceTerminated	Nachricht: Prozeßinstanz abnormal beendet
WMRequestChangeProcessInstanceState	Status einer Prozeßinstanz ändern
WMNotifyProcessInstanceStateChanged	Nachricht: Status einer Prozeßinstanz geändert
WMNotifyProcessInstanceCompleted	Nachricht: Prozeßinstanz normal beendet
WMRequestListProcessInstances	Liste von Prozeßinstanzen anfordern
WMNotifyProcessAttributesChanged	Nachricht: Prozeßattribute geändert
WMRequestRelinquishProcessInstance	Desinteresse an einer Prozeßinstanz

Tabelle 9: Routinen des IDL-Interfaces zur Prozeßsteuerung

8.2.2 Implementation des CORBA-Servers

Der CORBA-Server ermöglicht den Aufruf von Routinen der Workflow-Engine von InterOffice, indem er die entsprechenden Statements in der Datenbank ausführt. Die meisten Routinen der IDL-Schnittstelle entsprechen einer Routine der Workflow-Engine. Ein Beispiel dafür ist die in Anhang 2 enthaltene Implementation der Routine WMRequestCreateProcessInstance, die die *PL/SQL-Stored Procedure* CreateProcess im Package WF_ENGINE aufruft.

8.2.3 Implementation des CORBA-Clients

Für die Beispielimplementation stand kein zweites Workflowprodukt zur Verfügung, mit dem die Möglichkeiten der implementierten Schnittstelle demonstriert werden konnten. Deshalb wurde ein Java-Client entworfen, der die Rolle dieses zweiten Systems übernimmt, obwohl er selbst über keinerlei Workflow-Funktionalität verfügt. Mit dem Client können Prozeßinstanzen in InterOffice erzeugt und verwaltet werden. Außerdem können zu den erzeugten Instanzen Statusinformationen abgefragt werden. Abbildung 23 zeigt die Oberfläche des Java-Clients.

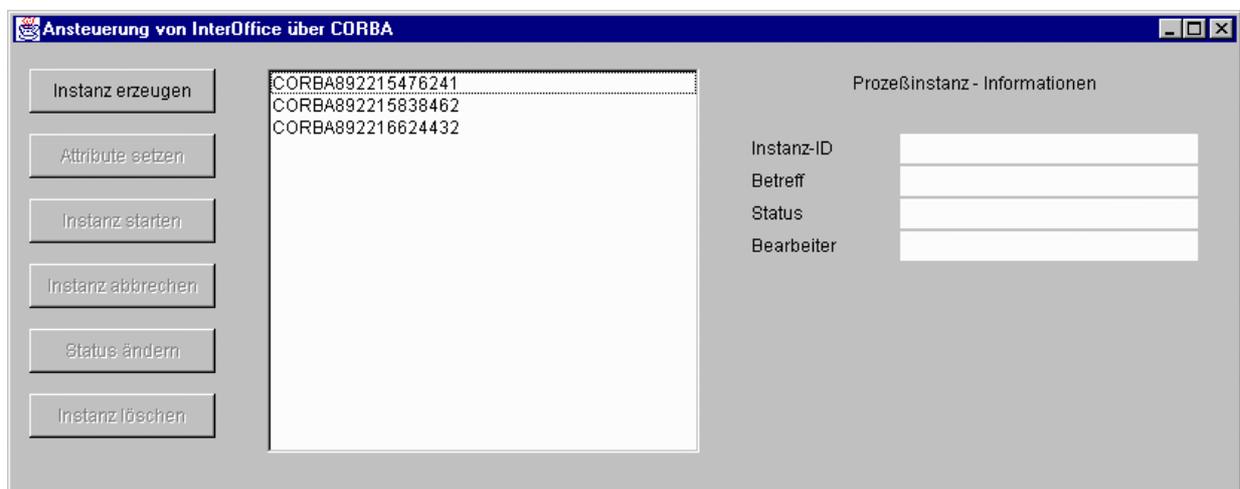


Abbildung 23: Oberfläche des CORBA-Clients

8.3 Einordnung des Beispielprojekts

Die hier vorgestellte Implementierung einer standardisierten Schnittstelle zur Prozeßsteuerung ermöglicht die Ausführung und Verwaltung von Prozeßinstanzen der Workflowkomponente von InterOffice. Die Schnittstelle läßt sich auch für andere Produkte implementieren, die über irgendeine Schnittstelle zur Prozeßsteuerung verfügen. Die damit erreichbare Workflow-Interoperabilität wird in diesem Abschnitt beschrieben.

8.3.1 Unterstützte Interoperabilitätsszenarien

In Abschnitt 5.6.1 wurden Szenarien aufgeführt, in denen verschiedene Workflow-Management-Systeme unterschiedlich stark miteinander gekoppelt sind. Die im ersten Szenario beschriebene Kopplung erfordert lediglich das Anstoßen von Vorgängen im anderen System. Das ist bereits dann möglich, wenn die oben beschriebene Schnittstelle nur in einer Richtung implementiert

ist, also das aufrufende System nur über einen CORBA-Client und das andere System nur über einen CORBA-Server verfügt. Verfügen hingegen beide Systeme über je einen CORBA-Client und einen CORBA-Server, wird auch Szenario 2 unterstützt. Die Realisierung der Szenarien 3 und 4 mittels standardisierter Schnittstellen erfordert den Zugriff auf den Inhalt von Prozeßdefinitionen über diese Schnittstellen. Das Interface 4 der WfMC bietet diese Möglichkeit zur Zeit nicht, was in Abschnitt 5.7 bereits erwähnt ist. Selbst wenn eine solche standardisierte Schnittstelle existieren würde, ist eine Implementation in InterOffice nicht ohne weiteres möglich. Die Gründe dafür können in Abschnitt 6.1.1 nachgelesen werden.

Das in Anhang 3 enthaltene Prozeßmodell beinhaltet Aktivitäten, die verschiedenen Organisationen zugeordnet sind. Trotzdem wurde der Vorgang als ein Prozeß modelliert. Der Grund dafür ist in Abschnitt 8.1.1 nachzulesen. Hier werden Überlegungen dazu angestellt, wie die Aufteilung des Vorgangs auf zwei verschiedene Systeme mittels der in Abschnitt 8.2 vorgestellten Schnittstelle realisiert werden kann. Dabei soll der Prozeß weiterhin mit InterOffice ausgeführt und ein Abschnitt des Prozesses als Subprozeß in einem anderen System implementiert werden.

Dazu sind zunächst einige Änderungen am Prozeßmodell erforderlich. Die der OFD zugeordneten Aktivitäten von *Bearbeiterauswahl (V3)* bis *Zustimmung erteilen (V3)* werden in einem Subprozeß zusammengefaßt, der im Workflow-Management-System der OFD implementiert wird. An der Stelle dieser Aktivitäten wird in das bisherige Prozeßmodell eine neue Aktivität eingefügt. Diese Schnittstellen-Aktivität soll die Ausführung des Subprozesses im anderen System bewirken und muß in InterOffice implementiert werden. Der fehlerlose Ablauf eines solchen Subprozesses ist in Tabelle 10 skizziert.

Während der systemübergreifenden Ausführung eines Subprozesses können in beiden Systemen Fehler auftreten. Mit Hilfe der Schnittstellenroutinen ist eine Fehlerbehandlung in Ansätzen möglich. Im Falle eines semantischen Fehlers im aufrufenden System, der zum Prozeßabbruch führt, steht die Routine *WMRequestAbortProcessInstance* zur Verfügung. Damit wird dem Zielsystem mitgeteilt, daß die Subprozeßinstanz abgebrochen werden kann. Ein semantischer Fehler im Zielsystem sollte den Aufruf der Routine *WMNotifyProcessInstanceAborted* bewirken. Daraufhin kann das aufrufende System eine entsprechende Fehlerbehandlung einleiten, die in InterOffice allerdings programmiert werden muß. Im Falle von Systemfehlern, die einen Neustart des Systems erfordern, hat jedes System seine eigenen Recovery-Mechanismen. Das aufrufende System muß sich nach erfolgreichem Neustart außerdem über den Zustand der Subprozeßinstanzen informieren. Dazu wird eine persistente Liste der Instanzen benötigt, die als Datenbanktabelle realisiert werden kann.

	Beschreibung
1	Die zur Schnittstellen-Aktivität gehörende Prozedur wird aufgerufen. Sie hat Zugriff auf alle Daten der Prozeßinstanz und der Aktivitätsinstanz.
2	Die Routine WMRequestCreateProcessInstance wird aufgerufen. Im Zielsystem wird eine Prozeßinstanz des Subprozesses erzeugt.
3	Mittels der Routine WMRequestSetProcessInstanceAttributes werden alle benötigten Daten ins Zielsystem übertragen. Im Beispiel sind das u.a. die zum Vorgang gehörenden Dokumente und die bisherigen Bearbeitungsergebnisse.
4	Mit dem Aufruf der Routine WMRequestStartProcessInstance wird die Subprozeßinstanz gestartet.
5	Das Zielsystem ruft die Routine WMNotifyProcessInstanceCompleted auf. Dadurch wird der Workflow-Engine von InterOffice mitgeteilt, daß der Subprozeß beendet ist.
6	Die Routine WMRequestGetProcessInstanceAttributes wird aufgerufen, um die Ergebnisse der Bearbeitung zu ermitteln und abzuspeichern.
7	Durch den Aufruf der Routine WMRequestRelinquishProcessInstance wird dem Zielsystem mitgeteilt, daß kein Interesse mehr an der Subprozeßinstanz besteht.
8	Die zur Schnittstellen-Aktivität gehörende Prozedur wird beendet.

Tabelle 10: Produktübergreifende Ausführung eines Subprozesses

8.3.2 Unterstützte Interoperabilitätsstufen

Die Workflow Management Coalition unterscheidet hinsichtlich der Zusammenarbeit verschiedener Workflow-Management-Systeme acht Interoperabilitätsstufen, die im Abschnitt 5.6.2 aufgeführt sind. Stufe 2 verlangt lediglich die Koexistenz verschiedener Produkte in derselben Umgebung, was durch das Produkt InterOffice gewährleistet ist (siehe Abschnitt 6.2). Auf Stufe 3 wird ein Gateway benötigt, daß zwischen Programmen mit nichtstandardisierten Schnittstellen vermittelt. Die Beispielimplementierung geht über diese Forderung hinaus, indem sie die von der WfMC vorgeschlagenen Routinen zur Prozeßsteuerung mittels einer CORBA-Implementierung unterstützt. Damit existiert eine standardisierte Schnittstelle mit grundlegenden Funktionen, wie sie von Stufe 4 gefordert wird. Da bisher nur eine abstrakte

Spezifikation des Interface 4 der WfMC vorliegt, bleibt die Auswahl der der Schnittstelle zugrundeliegenden Architektur implementationsspezifisch. Die Verwendung von CORBA bietet sich aus verschiedenen, in Abschnitt 7.4 aufgeführten Gründen an.

Stufe 5 verlangt eine einzige standardisierte Schnittstelle mit sämtlichen Operationen, die allen Workflowsystemen gemeinsam sind. Diese Forderung wird von den Routinen zur Prozeßsteuerung keinesfalls erfüllt, so daß Stufe 5 nicht erreicht wird. Steht die Menge der zu unterstützenden Funktionen fest, läßt sich die Schnittstelle in dem Maße erweitern, wie es der Funktionsumfang des jeweiligen Workflow-Management-Systems erlaubt. Alle höheren Stufen sind in absehbarer Zeit nicht erreichbar, da bereits Stufe 6 die Verwendung eines gemeinsamen Formats für Prozeßdefinitionen fordert, das noch nicht existiert.

8.3.3 Unterstützung zwischenbetrieblicher Anwendungen

In Abschnitt 7.2 findet sich eine Typologie zwischenbetrieblicher Anwendungen auf der Grundlage der Kriterien Grad der Interaktivität, Reichweite und Automatisierungsgrad. Im hier behandelten Fall besteht die zwischenbetriebliche Anwendung aus zwei oder mehr Workflow-Management-Systemen, die über ihre standardisierte CORBA-Schnittstelle und das Internet miteinander in Verbindung stehen. Da Workflow-Management-Systeme der Automatisierung von Arbeitsvorgängen dienen, ist der Automatisierungsgrad solcher Systeme von vornherein als hoch einzustufen. Die ersten beiden Kategorien ‚Informationsbereitstellung‘ und ‚Informationsbereitstellung mit Kontaktangebot‘ weisen einen niedrigen Automatisierungsgrad auf. Anwendungen dieser Kategorien verlangen keinerlei Workflow-Funktionalität und können mit einfacheren Mitteln erstellt werden.

Die Kategorie ‚Anstoßen eines Vorgangs‘ stellt den Minimalfall einer zwischenbetrieblichen Anwendung mit Workflow-Management-System dar. Die Realisierung einer solchen Anwendung mit Hilfe der implementierten CORBA-Schnittstelle ist denkbar einfach. Benötigt wird lediglich ein Programm, das wie der in Abschnitt 8.2.3 beschriebene Java-Client IDL-Routinen aufruft.

Die CORBA-Schnittstelle bietet die Möglichkeit, zwischenbetriebliche Teilschritte von Geschäftsprozessen abzuwickeln. Deshalb sind Anwendungen der Kategorie ‚interaktive Vorgangsabwicklung‘ möglich. Werden zusätzlich Workflow-Management-Systeme und andere Groupware durchgängig für die Durchführung der innerbetrieblichen Vorgänge eingesetzt, ist die gesamte Anwendung in die Kategorie ‚Geschäftsprozeßintegration‘ einzuordnen. Ein unternehmensübergreifender Geschäftsprozeß besteht dann aus mehreren Prozessen, die in den entsprechenden Systemen implementiert sind und über die Schnittstelle miteinander in Verbindung stehen.

Anwendungen der Kategorie ‚Informationskooperation‘ erfordern den Bezug auf unternehmensübergreifende Prozesse. Dies erfordert systemübergreifende Prozeßdefinitionen, die zum heutigen Zeitpunkt im Allgemeinen nicht möglich sind.

8.3.4 Grenzen

Bereits in Abschnitt 5.7 wurde festgehalten, daß das Fehlen eines Standards für Prozeßdefinitionen die weitere Entwicklung des gesamten Standards der WfMC behindert. In den vergangenen Abschnitten ist mehrmals deutlich geworden, daß dies bereits heute negative Auswirkungen für die Interoperabilität von Workflow-Management-Systemen hat. So können systemübergreifende Operationen nur auf Prozeßebene stattfinden, weil über den Inhalt von Prozeßdefinitionen keine Aussagen getroffen werden können.

Angesichts dieser Tatsache ist es nicht verwunderlich, wenn die Systeme keinerlei oder wenig Unterstützung hinsichtlich der Interoperabilität mit anderen Systemen bieten. InterOffice besitzt immerhin gut dokumentierte Schnittstellen, die eine Implementation bestimmter Operationen ermöglicht. Das war auch die Voraussetzung dafür, daß die Implementation der standardisierten Schnittstelle überhaupt möglich war. Das Produkt hat aber eine für heutige Workflowprodukte typische Schwäche. Die Entwickler haben ihre Vorstellung eines Workflow-Management-Systems umgesetzt und Erweiterungen des Systems nur ungenügend in Betracht gezogen. Deutlich wird das unter anderem, wenn man die Unterstützung flexibler Formulare betrachtet (s. Abschnitt 6.1.1). Dadurch wird die Verwendung derselben Formulare in verschiedenen Prozessen erschwert, was natürlich auch der systemübergreifenden Verkettung von Prozessen hinderlich ist. Auch wenn InterOffice eine Groupware und damit bezüglich der Herkunft kein Workflow-Management-System ist, stellt die mangelnde Interoperabilität der Workflowkomponente eine Einschränkung dar, deren Beseitigung die Einsatzmöglichkeiten des Produkts erheblich verbessern könnte.

9 Kurzzusammenfassung

Die Arbeit gibt einen Überblick über grundlegende Konzepte des Bereichs Workflow-Management, die von den Produkten auf unterschiedliche Art und Weise und in verschiedenem Umfang realisiert werden.

Die Beispielimplementation demonstriert die Möglichkeiten des Produkts InterOffice bezüglich der Interoperabilität mit anderen Workflow-Management-Systemen. Die volle Leistungsfähigkeit der Schnittstelle steht dann zur Verfügung, wenn sie von zwei oder mehr Workflow-Management-Systemen in beiden Richtungen genutzt wird. Das auch damit noch keine ausreichende Interoperabilität erreicht werden kann, ist in Abschnitt 8.3 deutlich geworden. Wünschenswerte Erweiterungen der Schnittstelle sind zur Zeit nicht möglich, da sich die entsprechenden Standards noch in der Entwicklung befinden.

Die eigentliche Herausforderung stellen heutzutage nicht die Unterschiede zwischen den verschiedenen Hard- und Softwareplattformen dar, die durch den Einsatz von Middleware überbrückt werden können, sondern die Heterogenität hinsichtlich der Workflow-Metaschemata, die unter anderem auf die unterschiedliche Herkunft der Workflowprodukte zurückzuführen ist. Einen vielversprechenden Ansatz zur Standardisierung stellt die Workflow Management Facility der OMG dar, an deren Spezifikation zur Zeit gearbeitet wird. Ein Workflow-Metaschema gehört hier unter anderem zu den obligatorischen Anforderungen. Die Diskussion von Mechanismen für die unternehmensübergreifende Ausführung von Workflows ist im zugehörigen Request for Proposals angeregt.

Mit der Weiterentwicklung der Standards der WfMC und der OMG werden sich auch höhere Stufen der Workflow-Interoperabilität erreichen lassen. Dann lohnt sich das Nachdenken über eine Erweiterung der Schnittstelle zur Prozeßsteuerung. Die Schnittstellen der in Abschnitt 6.1 vorgestellten Systeme sind dafür wenig geeignet, da sie nicht standardisiert, sondern am Einsatzzweck des jeweiligen Systems orientiert sind. Wenn der Standardisierungsprozeß den Weg in die Produkte findet, kann die Interoperabilität von Workflow-Management-Systemen, auch unternehmensübergreifend, Realität werden.

Literaturverzeichnis

- 1 [AdminFlow] <<http://www.hp.com/ghp/products.html>>
- 2 [Austin 62] J. Austin: How to do Things with Words, Oxford University Press, Oxford 1962
- 3 [Baeten/Weijland 90] J.C.M. Baeten, W.P. Weijland: Process Algebra, Cambridge University Press 1990
- 4 [Baumgarten 96] B. Baumgarten: Petri-Netze. Grundlagen und Anwendungen, Spektrum Akademischer Verlag 1996
- 5 [Böhm/Schulze 95] Markus Böhm / Wolfgang Schulze: Grundlagen von Workflow-Management-Systemen, erschienen in: Wissenschaftliche Beiträge zur Informatik, 8 (1995) Heft 2, TU Dresden, S. 50-65
- 6 [CORBAservices] CORBAservices: Common Object Services Specification, <<http://www.omg.org>>
- 7 [CORBAspecification] The Common Object Request Broker: Architecture and Specification, Revision 2.2, Februar 1998, <<http://www.omg.org>>
- 8 [CORBAfacilities] CORBAfacilities: Common Facilities Architecture, <<http://www.omg.org>>
- 9 [Eder et al.] J. Eder, H. Groiss, W. Liebhart: Workflow-Systeme im WWW, <<http://www.ifi.uni-klu.ac.at/~herb/Overview.html>>
- 10 [Flores et al. 88] F. Flores, M. Graves, B. Hartfield, T. Winograd: Computer Systems and the Design of Organizational Interaction, erschienen in: ACM Transactions on Office Information Systems, vol. 6 no. 2, April 1988, S. 153-172
- 11 [FlowMark] <<http://partnerinfo.software.ibm.com/ntbdk/HTML/FLWMRK/START.HTM>>
- 12 [Hennessy 88] M. Hennessy: Algebraic Theory of Processes, MIT Press 1988
- 13 [InterOffice] <<http://www.oracle.com.sg/products/interoffice/html/>>
- 14 [Jablonski/Bußler 96] Stefan Jablonski, Christoph Bußler: Workflow Management: Modeling Concepts, Architecture and Implementation, International Thomson Computer Press 1996
- 15 [Jablonski et al. 97] Stefan Jablonski / Markus Böhm / Wolfgang Schulze: Workflow-Management – Entwicklung von Anwendungen und Systemen, dpunkt – Verlag 1997

- 16 [Kurbel 97] Karl Kurbel: Internet-Nutzung im Business-to-Business-Bereich: Stand der Entwicklung, Typologie und Anwendungsbeispiele; aus: Herrmann Krallmann: Wirtschaftsinformatik '97: Internationale Geschäftstätigkeit auf der Basis flexibler Organisationsstrukturen und leistungsfähiger Informationssysteme, Physica-Verlag 1997
- 17 [Lockemann 93] Peter Lockemann: Datenbank-Handbuch, Springer 1993
- 18 [Milner 89] R. Milner: Communication and Concurrency, Prentice Hall 1989
- 19 [MOBILE] Webseiten der Universität Erlangen, <<http://www6.informatik.uni-erlangen.de/Research/wfm.html>>
- 20 [Oberweis 96] Andreas Oberweis: Modellierung und Ausführung von Workflows mit Petri-Netzen, Teubner 1996, Reihe: Teubner-Reihe Wirtschaftsinformatik
- 21 [Reuter/Schwenkreis] Andreas Reuter / Friedemann Schwenkreis: ConTracts – A low-level Mechanism for Building General-Purpose Workflow-Management-Systems, Universität Stuttgart, IPVR
- 22 [Schulze/Böhm] Wolfgang Schulze, Markus Böhm: Klassifikation von Vorgangsverwaltungssystemen, in: [Vossen, Becker], Seite 279 ff.
- 23 [Schulze/Böhm/Meyer-Wegener 94] Wolfgang Schulze, Markus Böhm, Klaus Meyer-Wegener: Vergleichende Bewertung objektorientierter Datenbanksysteme für die Speicherung von Vorgängen und Dokumenten, Technischer Bericht (3.Meilenstein) BERKOM-Projekt DOPAS, TU Dresden 1994
- 24 [Schulze/Bussler/Meyer-Wegener 98] Wolfgang Schulze, Christoph Bussler, Klaus Meyer-Wegener: Standardizing on Workflow-Management – The OMG Workflow Management Facility, in: ACM SIGGROUP Bulletin, Volume 19 (3), April 1998
- 25 [Teufel et al. 95] Stefanie Teufel / Christian Sauter / Thomas Mühlherr / Kurt Bauknecht: Computerunterstützung für die Gruppenarbeit, Addison-Wesley 1995
- 26 [Vossen/Becker] Gottfried Vossen, Jörg Becker: Geschäftsprozessmodellierung und Workflow-Management, Int. Thomson Publishing 1996
- 27 [Wersch 95] Markus Wersch: Workflow Management, DUV 1995
- 28 [Wodtke 97] Dirk Wodtke: Modellbildung und Architektur von verteilten Workflow-Management-Systemen, Infix 1997, Reihe: Dissertationen zu Datenbanken und Informationssystemen, 31
- 29 [WfMC Glossary] The Workflow Management Coalition Specification - Terminology & Glossary, Document Number WfMC-TC-1011, Juni 96, <http://www.aiai.ed.ac.uk:80/project/wfmc/DOCS/glossary/glossary.html>

- 30 [WfMC Reference Model] The Workflow Management Coalition Specification –The Workflow Reference Model, Document Number WFMC-TC00-1003, November 94
- 31 [WfMC-TC-1012] Workflow Standard – Interoperability Abstract Specification, Document Number WFMC-TC-1012, Oktober 1996
- 32 [WfMC-TC-1013] Workflow Application Programmer's Interface Naming Conventions, Document Number WFMC-TC-1013, Mai 1996

Anhang 1 IDL-Interface zur Prozeßkommunikation

```
/**
 * Jens Drawehn
 * 26.03.1998
 *
 * CORBA-Schnittstelle für die Kommunikation zwischen Workflow-Management-
Systemen
 * Versuch einer Umsetzung der WfMC Interoperability Abstract Specification
 *
 */

// Basisdatentypen
// Quelle : WfMC WAPI Naming Conventions, TC10-0013, Version 1.1, 15.03.1996
// entfernt wegen Problemen mit dem Mapping
/*
typedef char           WMTInt8;
typedef short         WMTInt16;
typedef long          WMTInt32;
typedef unsigned char WMTUInt8;
typedef unsigned short WMTUInt16;
typedef unsigned long WMTUInt32;
typedef string        WMTText;
typedef boolean       WMTBoolean;
*/

// Standardparameter
// Quelle : WfMC WAPI Naming Conventions, TC10-0013, Version 1.1, 15.03.1996
// Quelle : WfMC Interoperability Abstract Specification, WfMC-TC-1012,
Version 1.0, 20.10.1996

// Standard-Rückgabewerte von Routinen des Interfaces
enum WMTReturnCode {
    Failure,
    Operation_not_implemented,
    Operation_not_performed,
    Success
};

// zur Zeit nicht benutzt
struct WMTFilter {
    long         filter_type;
    long         filter_length;
    string       attribute_name;
};
```

```

        unsigned long comparison;
        string      filter_string;
    };

// Identifier für Workflow Engines
struct WMTEngineID {
    string      id;
};

// Standard-Identifier für Objekte
struct WMTObjectID {
    string      id;
};

// Liste von Objekt-Identifiern
typedef sequence<WMTObjectID> WMTObjectIDList;

// Identifier für Workflow-Aktivitäten
struct WMTActivityID {
    string      id;
};

// Identifier für Workflow-Ressourcen
struct WMTResourceID {
    string      id;
};

// Status von Workflow-Objekten
struct WMTObjectState {
    string      state;
};

// Workflow-Attribute
struct WMTAttribute {
    string      name;
    string      value;
};

// Liste von Workflow-Attributen
typedef sequence<WMTAttribute> WMTAttributeList;

// IDL-Definition des Interfaces mit den Operationen zur Prozeßsteuerung
// Quelle : WfMC Interoperability Abstract Specification, WfMC-TC-1012,
// Version 1.0, 20.10.1996
interface ProcessControl {

    // steht die Datenbankverbindung des CORBA-Servers ?

```

```

readonly attribute boolean GetConnectState;

// Initialisierung einer Prozeßinstanz (ohne Start der Instanz)
WMTReturnCode WMRequestCreateProcessInstance (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_definition_id,
    in   boolean           return_flag,
    in   WMTObjectID      parent_id,
    in   WMTActivityID     activity_id,
    out  WMTObjectID      sub_process_id,
    out  WMTResourceID     user_id,
    out  WMTResourceID     role_id
);

// Ermitteln des Status einer Prozeßinstanz
WMTReturnCode WMRequestGetProcessInstanceState (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_id,
    out  WMTObjectState   state
);

// Setzen von Attributwerten einer Prozeßinstanz
WMTReturnCode WMRequestSetProcessInstanceAttributes (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_id,
    in   WMTAttributeList attributes
);

// Ermitteln von Attributwerten einer Prozeßinstanz
WMTReturnCode WMRequestGetProcessInstanceAttributes (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_id,
    in   WMTAttributeList request_attributes,
    out  WMTAttributeList result_attributes
);

// Starten einer Prozeßinstanz (muß vorher initialisiert werden)
WMTReturnCode WMRequestStartProcessInstance (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_id
);

// Benachrichtigung über den erfolgten Start einer Prozeßinstanz
WMTReturnCode WMNotifyProcessInstanceStarted (
    in   WMTEngineID      engine_identifizier,
    in   WMTObjectID      process_id
);

// bricht die Ausführung einer Prozeßinstanz ab
WMTReturnCode WMRequestAbortProcessInstance (

```

```

        in  WMTEngineID      engine_identifizier,
        in  WMTOBJECTID      process_id
    );

// Benachrichtigung über den erfolgten Abbruch einer Prozeßinstanz
WMTReturnCode WMNotifyProcessInstanceAborted (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id
);

// Beenden einer Prozeßinstanz
WMTReturnCode WMRequestTerminateProcessInstance (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id
);

// Benachrichtigung über die erfolgte Beendigung einer Prozeßinstanz
WMTReturnCode WMNotifyProcessInstanceTerminated (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id
);

// Ändern des Status eines Prozesses
WMTReturnCode WMRequestChangeProcessInstanceState (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id,
    in  WMTOBJECTSTATE    state
);

// Benachrichtigung über eine Statusänderung
WMTReturnCode WMNotifyProcessInstanceStateChanged (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id,
    in  WMTOBJECTSTATE    new_state
);

// Benachrichtigung über die normale Beendigung einer Prozeßinstanz
WMTReturnCode WMNotifyProcessInstanceCompleted (
    in  WMTEngineID      engine_identifizier,
    in  WMTOBJECTID      process_id
);

// liefert eine Liste aller Prozeßinstanzen
WMTReturnCode WMRequestListProcessInstances (
    in  WMTEngineID      engine_identifizier,
    in  WMTFILTER         filter,
    out WMTOBJECTIDLIST   process_id_list
);

// Benachrichtigung über Änderungen von Attributen
WMTReturnCode WMNotifyProcessAttributesChanged (

```

```
        in  WMTEngineID      engine_identifizier,  
        in  WMTOBJECTID     process_id,  
        in  WMTAttributeList attributes  
);  
  
// Löschen aller Informationen bezüglich einer Prozeßdefinition  
WMTReturnCode WMRequestRelinquishProcessInstance (  
    in  WMTEngineID      engine_identifizier,  
    in  WMTOBJECTID     process_id  
);  
  
};
```

Anhang 2 Java-Beispielroutine des CORBA-Servers

```
// legt eine Prozessdefinition fest, die spaeter ausgefuehrt werden soll
public WMTReturnCode WMRequestCreateProcessInstance (
    WMTEngineID          engine_identifizier,
    WMTObjectID          process_definition_id,
    boolean              return_flag,
    WMTObjectID          parent_id,
    WMTActivityID        activity_id,
    WMTObjectIDHolder    sub_process_id,
    WMTResourceIDHolder  user_id,
    WMTResourceIDHolder  role_id )
{
    System.out.println( "Prozess erzeugen" );

    // neuen Prozess-Instanz-Schlüssel erzeugen
    String ItemKey = "CORBA" + ( new java.util.Date() ).getTime();

    // Prozessinstanz erzeugen
    try {
        CallableStatement myCallStatement = IOConnection.prepareCall(
            "{call WF_ENGINE.CreateProcess( ?, ?, ? )}" );
        myCallStatement.setString( 1, ItemType );
        myCallStatement.setString( 2, ItemKey );
        myCallStatement.setString( 3, process_definition_id.id );
        myCallStatement.executeUpdate();
        myCallStatement.close();
    } catch( SQLException ex ) {
        System.out.println( "SQLException : " + ex.toString() );
        return ( WMTReturnCode.Failure );
    }

    // Rueckgabewerte setzen
    sub_process_id.value.id = ItemKey;
    // Nutzer und Rolle bleiben dem aufrufenden System verborgen
    user_id.value.id = "UNKNOWN";
    role_id.value.id = "UNKNOWN";

    // Prozess in Prozessliste eintragen ( fuer spaetere Abfragen )
    try {
        PreparedStatement myStatement = IOConnection.prepareStatement(
            "INSERT INTO IO_CORBA_PROCESS_INSTANCES ( ITEM_TYPE, ITEM_KEY, PROCESS_NAME,
            PROCESS_ID ) VALUES ( ?, ?, ?, ? )" );
        myStatement.setString( 1, ItemType );
        myStatement.setString( 2, ItemKey );
        myStatement.setString( 3, process_definition_id.id );
        myStatement.setString( 4, ItemKey );
    }
}
```

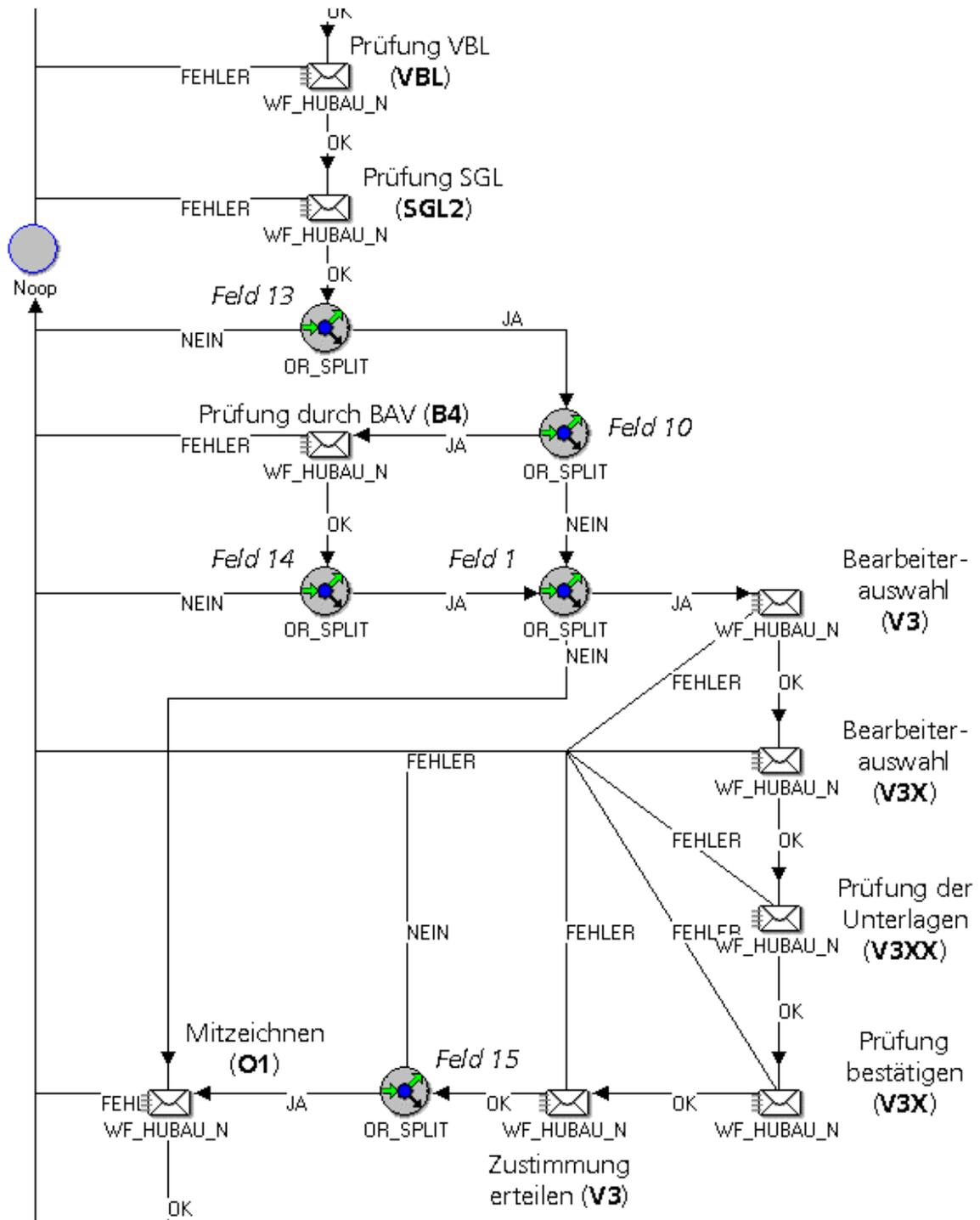
```

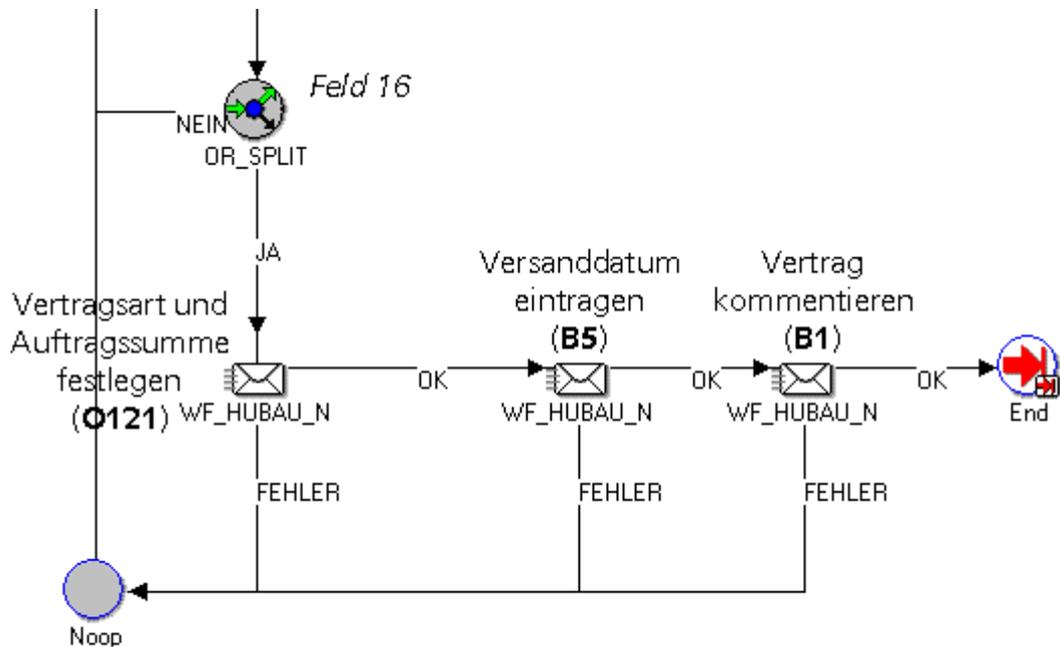
        myStatement.executeUpdate();
        myStatement.close();
    } catch( SQLException ex ) {
        System.out.println( "SQLException : " + ex.getMessage() );
        return ( WMTReturnCode.Failure );
    }

    // Attribut 'ADMIN' setzen ( dieser Nutzer ist fuer die Fehlerbehandlung
im Zielsystem zustaendig )
    WMTReturnCode myReturnCode = null;
    WMTAttribute[] myAttributes = new WMTAttribute[1];
    myAttributes[0] = new WMTAttribute( "ADMIN", "BAUAMT_B1" );
    try { myReturnCode = WMRequestSetProcessInstanceAttributes (
        engine_identifier, sub_process_id.value, myAttributes );
    } catch ( SystemException se ) {
        System.out.println ( "Call to object failed : " + se.toString () );
        return ( WMTReturnCode.Failure );
    }

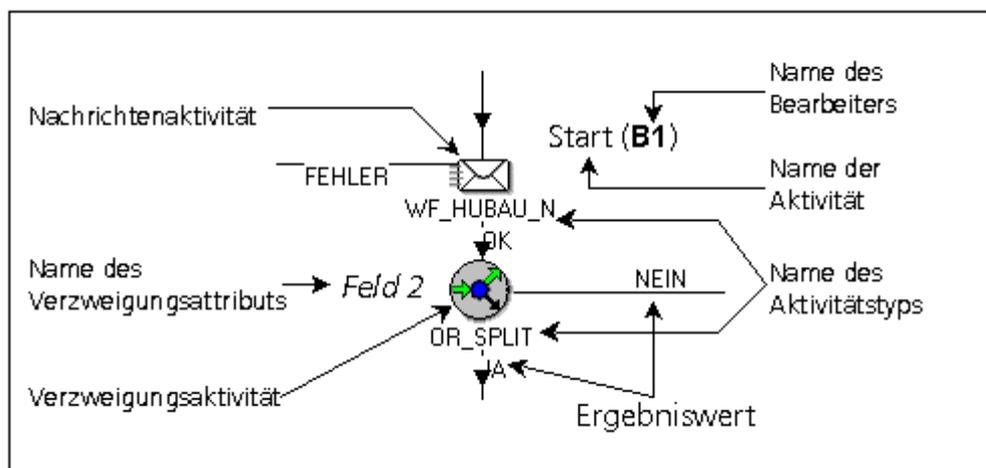
    System.out.println( "Prozess " + ItemKey + " erzeugt" );
    return ( WMTReturnCode.Success );
}

```



Die folgende Legende zeigt einen Ausschnitt aus dem Prozeßdiagramm, der eine Nachrichtenaktivität und eine Verzweigungsaktivität enthält.



Anhang 4 Inhalt der CD-ROM

Dateien	Beschreibung
DA.doc	die Diplomarbeit (MS Word 97)
poliflow\process diagram.ppt	das Diagramm des Beispielprozesses aus Anhang 3 (MS PowerPoint 97)
poliflow\sql*.sql	die Implementation der für den Interoffice-Beispielprozeß benötigten Aktivitäten (PL/SQL-Quellcode)
poliflow\corba\workflow.idl	IDL-Schnittstellendefinition für Interaktionen zwischen Workflow-Management-Systemen (Quellcode)
poliflow\corba\java_output\defhg\iao\poliflow\workflow*.java	Ausgabe des IDL to Java-Compilers (Java-Quellcode)
poliflow\corba\Sim.java	Implementation des CORBA-Clients (Java-Quellcode)
poliflow\corba\IOServer.java	Implementation des CORBA-Servers (Java-Quellcode)
wfmc*.doc wfmc*.pdf	Dokumente der WfMC (MS Word, Adobe Portable Document Format)
omg*.pdf	Dokumente der OMG (Adobe Portable Document Format)

Erklärung

Ich versichere, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig

August 1998

Ort

Datum

Unterschrift