



**Universität
Gesamthochschule
Paderborn**

Diplomarbeit

Graphische Visualisierung und Analyse
verteilten Workflow Managements

-

Konzeption und prototypische Implementierung des
Wide Area GroupFlow Analyzers

Prof. Dr. Ludwig Nastansky

Prof. Dr. Hans-Ulrich Heiß

1997

vorgelegt von:

Olaf Hein

Studiengang Informatik

Matr.-Nr.: 3968767

Langer Weg 2, 33100 Paderborn

E-Mail: OHein@aol.com

Danksagung

Mein Dank gilt allen, die zum Gelingen dieser Diplomarbeit beigetragen haben. Besonders bedanken möchte ich mich bei Prof. Dr. Ludwig Nastansky und Dipl. Wirtsch. Ing. Gerold Riempp für die gute Betreuung und bei meiner Frau Andrea sowie meiner Tochter Jana für ihre Geduld. Weiterhin möchte ich mich bei allen Mitgliedern des WAW-Teams für die freundliche Unterstützung bedanken.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Paderborn, den

(Datum) (Unterschrift)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Szenario	1
1.2	Aufgabenstellung.....	2
1.3	Aufbau der Arbeit.....	2
2	Grundlagen	4
2.1	Groupware und Workgroup Computing.....	4
2.2	Workflows und Workflow Management	5
2.3	Verteiltes Workflow Management	6
2.4	Wide Area GroupFlow System	11
2.5	Lotus Notes als Basis-Umgebung des Wide Area GroupFlow Systems.....	15
3	Motivation	18
3.1	Ziele einer Workflow Analyse.....	18
3.1.1	Workflow Lebenszyklus.....	18
3.1.2	Workflow Referenz Modell der Workflow Management Coalition	19
3.1.3	GroupFlow und der GroupFlow Analyzer	21
3.1.4	Weitere Ansätze in der Literatur	22
3.2	Anforderungen an einen Workflow Analyzer.....	22
3.2.1	Visualisierung von Abläufen und Prozeßstrukturen	22
3.2.2	Statistische Auswertung von Vorgangsdaten.....	23
3.2.3	Schnittstellen zum Workflow System	23
3.3	Inhaltliche Abgrenzung.....	24
4	Konzeption und Lösungsansätze für eine Analyse verteilter Vorgangsbearbeitung	25
4.1	Vorgänge, Aufgaben und Bearbeiter als Gegenstand der Analyse	27
4.2	Arten der Visualisierung von Vorgängen.....	30
4.2.1	Kommunikationsbeziehungen zwischen Bearbeitern, Aufgaben oder Arbeitsschritten.....	32
4.2.2	Ein Bearbeiter, eine Aufgabe oder ein Arbeitsschritt im Mittelpunkt der Betrachtung.....	36
4.2.3	Bearbeitungsschritte von Vorgängen.....	39
4.3	Selektion zu analysierender Vorgänge.....	44
4.3.1	Einlesen von Daten aus einer Datenbank	44
4.3.2	Gezielte Auswahl einzelner Vorgänge	45
4.3.3	Zeitraumbezogene Analyse durch Differenzworkflows	47

4.4	Statistische Auswertung von Vorgangsdaten.....	47
4.4.1	Informationen über die Anzahl der bearbeiteten Vorgänge.....	47
4.4.2	Bearbeitete Aufgaben der Bearbeiter.....	48
4.4.3	Analyse eines Bearbeiters oder einer Aufgabe	49
4.4.4	Kommunikationsbeziehungen zwischen Bearbeitern oder Aufgaben	49
4.5	Anforderungen an Eingabedaten	50
5	Implementierung des Wide Area GroupFlow Analyzer	54
5.1	Verwendete Werkzeuge.....	54
5.1.1	Visual Basic 5.0 als Programmiersprache	54
5.1.2	MS Excel für die Statistische Auswertung.....	54
5.2	Beschreibung der Schnittstelle zum Wide Area GroupFlow System.....	55
5.3	Datenstruktur für das Abspeichern eines Vorganges innerhalb des WAGS Analyzers	58
5.3.1	Allgemeine Informationen über Graphen	58
5.3.2	Adjazenzliste als Repräsentation eines Vorganges	59
5.3.3	Einlesen der Vorgangsdaten und Aufbau des Graphen.....	60
6	Grenzen und Risiken der Analyse	65
6.1	Rechtliche Aspekte	65
6.1.1	Datenschutz.....	65
6.1.2	Betriebsverfassungsgesetz.....	66
6.2	Technische Grenzen.....	67
6.3	Soziale Gesichtspunkte	68
7	Ausblick.....	69
7.1	Erweiterung der Funktionalität des WAGS Analyzers	69
7.2	Unterschiede zum GroupFlow Analyzer.....	69
7.3	Verbesserung des Laufzeitverhaltens und des Ressourcenbedarfs des WAGS Analyzers	70
8	Zusammenfassung.....	71
9	Anhang	a
9.1	Data Dictionary	a
9.2	Literaturangaben.....	b

Abbildungsverzeichnis

Abbildung 1: Schadensfallmeldung	6
Abbildung 2: Drei Dimensionen des Verteilten Workflow Managements	7
Abbildung 3: Message Objekt	9
Abbildung 4: Content Management	10
Abbildung 5: Aspekte der organisatorischen Nähe	11
Abbildung 6: Wide Area GroupFlow System - Architektur	12
Abbildung 7: Wide Area GroupFlow System	14
Abbildung 8: Lotus Notes Ansicht	16
Abbildung 9: Workflow Lebenszyklus	19
Abbildung 10: Workflow Referenz Modell	20
Abbildung 11: Modellierung eines Workflows	25
Abbildung 12: Instanzen der Schadensfallmeldung	26
Abbildung 13: Bearbeiternamen filtern	29
Abbildung 14: Beispielvorgänge einer Bestellung	31
Abbildung 15: Kommunikationsbeziehungen zwischen Arbeitsschritten	33
Abbildung 16: Kommunikationsbeziehungen zwischen Bearbeitern	34
Abbildung 17: Bearbeitete Aufgaben einer Organisation	35
Abbildung 18: Kommunikationsbeziehungen zwischen Aufgaben	36
Abbildung 19: Ein Arbeitsschritt im Mittelpunkt	37
Abbildung 20: Ein Arbeitsschritt im Mittelpunkt der Betrachtung	38
Abbildung 21: Eine Aufgabe im Mittelpunkt	39
Abbildung 22: Ein Bearbeiter im Mittelpunkt	39
Abbildung 23: Bearbeitungsschritte Arbeitsschritt	41
Abbildung 24: Bearbeitungsschritte Aufgaben	42
Abbildung 25: Bearbeitungsschritte Bearbeiter	43
Abbildung 26: Auswahl der Feldnamen	45
Abbildung 27: Selektion von Vorgängen	46
Abbildung 28: Bearbeiter auswählen	46
Abbildung 29: Bearbeiter Diagramm	48
Abbildung 30: Aufgaben Diagramm	49
Abbildung 31: Falsche Rekonstruktion eines Vorganges	52
Abbildung 32: Entity-Relationship-Modell eines Vorganges	56
Abbildung 33: Graph	58
Abbildung 34: Adjazenzliste	60

Tabellenverzeichnis

Tabelle 1: Namenskonvention	28
Tabelle 2: Felder im Trackingdokument	57

Abkürzungsverzeichnis

API.....	Aplication Programming Interface
BDSG.....	Bundesdatenschutzgesetz
BetrVG.....	Betriebsverfassungsgesetz
CSCW	Computer Supported Cooperative Work
DB.....	Datenbank
DLL.....	Dynamic Link Library
DWM	Distributed Workflow Management
EMail.....	Electronic Mail
engl.....	Englisch
ff.....	folgende (Seiten)
GRI	globale Routing Informationen
ID	Identifikationsnummer
IRI.....	interne Routing Informationen
LAN	Local Area Network
OCX.....	OLE Control Extension
OLE	Object Linking and Embedding
WAGS.....	Wide Area Groupflow System
WAN.....	Wide Area Network
WAPI	Workflow Aplication Programming Interface
WAW	Wide Area Workflow
WFMC	Workflow Management Coalition
WFMS.....	Workflow Management System
WMC	WorkFlow Management Cycle
WWW	World Wide Web
VBA.....	Visual Basic for Applications

1 Einleitung

1.1 Szenario

Workflow Management umfaßt sich schnell weiterentwickelnde Konzeptionen und Technologien. Es ist gekennzeichnet durch die computergestützte Automatisierung und Steuerung von kooperativer Büroarbeit. Diese kooperativen Arbeiten werden in der Literatur häufig als Prozeß, Workflow, Bürovorgang oder einfach Vorgang bezeichnet.

Durch den Einsatz von Workflow Management Systemen (WFMS) kann der herkömmliche Papier-Workflow effizienter gestaltet werden. Allein durch den Wegfall von Medienbrüchen entstehen Kosten- und Zeitersparnisse bei der Vorgangsbearbeitung. Weitere Vorteile ergeben sich unter anderem aus der Automatisierung wiederkehrender Abläufe, der gezielten Informationsbereitstellung durch das System oder der Unterstützung bei der Bearbeitung einzelner Aufgaben.

Kommerzielle Workflow Management Systeme sind beispielsweise IBM Flowmark, SNI WorkpartyPC, IABG ProMinanD, Filenet VisualWorkflow, Pavone GroupFlow¹, ARIS Workflow oder Onestone Prozeßware. Alle genannten Produkte sind a priori darauf ausgelegt, Vorgänge innerhalb einer Organisation zu unterstützen.

In vielen Fällen sind an Bürovorgängen aber nicht nur Mitarbeiter innerhalb einer Organisation beteiligt, sondern auch Lieferanten oder Kunden. Neue Organisationsformen wie virtuelle Unternehmen und Profit Center oder das Outsourcing von Teilen der Organisation erhöhen die Zahl der organisationsübergreifenden Vorgänge erheblich. Als logische Konsequenz sollten die daraus resultierenden Wide Area Workflows (WAW) auch durch ein WFMS unterstützt werden.

WAWs stellen neue Anforderungen an ein WFMS. Diese ergeben sich zum Beispiel aus der räumlichen Distanz zwischen den Organisationen oder den Sicherheitsbedürfnissen der Beteiligten. Daher ist es im allgemeinen nicht möglich, ein vorhandenes WFMS ohne weiteres für WAW zu verwenden.

¹ Vgl. [Hilpert, 1994]

An der Universität-GH Paderborn wird an der Lehr- und Forschungseinheit Wirtschaftsinformatik 2 (Prof. Dr. Nastansky) das Wide Area GroupFlow System (WAGS) entwickelt. Dies ist ein WFMS zur Unterstützung von Vorgängen über die Grenzen von rechtlich oder geographisch verteilten Organisationen hinweg. Ein Teil dieses Systems ist ein graphisches Werkzeug zur Analyse bearbeiteter Vorgänge. Die Konzepte und Lösungsansätze dieses Werkzeuges sind Gegenstand dieser Arbeit.

1.2 Aufgabenstellung

Das WAGS ist eine Weiterentwicklung von GroupFlow. Für GroupFlow existiert bereits ein Analysewerkzeug mit dem Namen GroupFlow Analyzer². Mit ihm ist es möglich, Daten von bearbeiteten Vorgängen einzulesen und auf verschiedene Arten zu visualisieren. Die gewonnenen Informationen können dann zur Definition neuer oder zur Verbesserung vorhandener Workflowtypen verwendet werden.

Analog zum GroupFlow Analyzer für einzelne Datenbanken soll im Kontext des Wide Area GroupFlow System-Projektes ein Analyse-Werkzeug entwickelt werden, mit dem abgelaufene WAW-Prozesse graphisch aufbereitet und ausgewertet werden können.

Insbesondere ist auch ein „Zeitscheiben-Konzept“ zum Vergleich verschiedener Entwicklungsschritte von Interesse.

1.3 Aufbau der Arbeit

Das folgende Kapitel befaßt sich mit Grundlagen, die für den Rest der Arbeit von Bedeutung sind. Zunächst werden wichtige Begriffe aus den Bereichen Groupware, Workgroup Computing und Workflow Management definiert und erläutert. Danach wird auf die besonderen Anforderungen an verteiltes Workflowmanagement eingegangen. Abschließend wird das Wide Area GroupFlow System und die zugrunde liegende Basis-Umgebung Lotus Notes vorgestellt.

Im dritten Kapitel werden die Ziele einer Workflow Analyse und die Anforderungen an ein Analysewerkzeug herausgearbeitet. Dazu werden auch Überlegungen anderer Autoren aufgegriffen.

² Vgl. [Piesk, Pötter, 1994]

Das vierte Kapitel beschreibt die Konzepte und Lösungsansätze des WAGS Analyzers. Den Anfang macht eine Beschreibung der grundlegenden Konzepte der Analyse. Danach werden die wichtigsten Funktionen des WAGS Analyzers vorgestellt. Dabei wird insbesondere darauf eingegangen, warum die einzelnen Funktionen implementiert wurden und wie sie genutzt werden können, um die gesetzten Ziele zu erreichen. Am Ende des Kapitels wird dargestellt, wie die Eingabedaten beschaffen sein müssen, um diese Analyse zu ermöglichen.

Kapitel fünf beschäftigt sich mit Implementierungsdetails des WAGS Analyzers. Dies beginnt mit der Beschreibung der verwendeten Werkzeuge und den Gründen für ihren Einsatz. Danach wird die Schnittstelle zum Rest des Wide Area GroupFlow Systems spezifiziert. Das letzte Drittel des Kapitels befaßt sich mit der Beschreibung der Datenstruktur für einen Vorgang und einigen eingesetzten Algorithmen.

Das sechste Kapitel zeigt Grenzen und Risiken einer Workflowanalyse auf. Hier wird auf rechtliche, soziale und technische Aspekte eingegangen.

Im letzten Kapitel wird ein Blick in die Zukunft gewagt, und es werden Möglichkeiten für die Weiterentwicklung des Analyzers aufgezeigt.

2 Grundlagen

2.1 Groupware und Workgroup Computing

Groupware und Workgroup Computing sind Begriffe, die seit Beginn der 90er Jahre immer häufiger im Zusammenhang mit computerunterstützter Büroarbeit verwendet werden. Sie können in den Bereich des CSCW (Computer Supported Cooperative Work) eingeordnet werden, einem interdisziplinären Forschungsgebiet, das sich mit Theorie und Konzepten computerunterstützter kooperativer Arbeitsprozesse befaßt. Bisher ist es noch nicht gelungen, eine klare Definition für die Begriffe Groupware und Workgroup Computing zu finden, und häufig werden sie synonym verwendet. Es ist schwierig, genau einzugrenzen und mit wenigen Worten zu beschreiben, was darunter zu verstehen ist. Dies liegt vor allem an der Komplexität der Systeme, die sie beschreiben. Daher soll hier auch keine klare Definition gegeben werden, sondern an Beispielen aus der Literatur aufgezeigt werden, was andere Autoren darunter verstehen. So schreibt Nastanski: „Der Versuch einer klaren Definition ergibt z.B.: Groupware stellt computergestützte Konzepte für die Teamarbeit bereit. Insbesondere müssen dabei natürlich, der Arbeitsfluß und das Vorgangsmanagement in den vielfältigen Kommunikations- und Arbeitsinteraktionen zwischen den Mitarbeiterinnen und Mitarbeitern im Office-Bereich bzw. in Projektteams unterstützt werden.“³, oder Lichtner: „Als Workgroup-Computing wird die Technik verstanden, die unter Ausnutzung einer vernetzten Arbeitsumgebung Informationen kontrolliert und verteilt zur Verfügung stellt. Der Schwerpunkt hierbei liegt in der Koordination von unstrukturierten Aufgaben und dem schnellen Treffen von Gruppenentscheidungen. Darüber hinaus sind die orts- und zeitungebundene Kommunikation sowie die gemeinsame Bearbeitung eines Objektes wichtige Aspekte.“⁴ Beide Autoren sehen die computergestützte Teamarbeit als eine wichtige Funktion von Groupware bzw. Workgroup-Computing an. Weiterhin muß das System die Anwender bei ihren Aufgaben unterstützen können. Wie diese Unterstützung genau aussehen soll oder welche Aufgaben zu unterstützen sind wird nicht gesagt.

³ Vgl. [Nastanski et. al. , 1995, S.274]

2.2 Workflows und Workflow Management

Der Begriff Workflow steht für die Automatisierung von Vorgängen, in denen Dokumente, Informationen oder Aufgaben nach festen Regeln zwischen den beteiligten Bearbeitern weitergegeben werden. Unter einem Vorgang wird in dieser Arbeit ein konkreter Bürovorgang verstanden. Dieser besteht aus einer oder mehreren Aufgaben, deren Erledigung in einer vorgegebenen Reihenfolge einem gemeinsamen Ziel dient. Aufgaben sind in diesem Kontext Tätigkeiten, die bei der Büroarbeit anfallen. Diese können sehr einfach aber auch sehr komplex sein und aus mehreren Teilaufgaben bestehen. Häufig wird ein Vorgang auch als Geschäftsprozeß (engl. Business process) bezeichnet. Hollingsworth definiert in einem „paper“ der Workflow Management Coalition (WFMC) Workflow wie folgt: „The computerised facilitation or automation of a business process, in whole or part.“⁵

Im folgenden wird der Begriff Workflow für einen oder mehrere konkrete Vorgänge verwendet. Ein einzelner Vorgang wird auch als Workflowinstanz bezeichnet. Ein Workflowtyp oder auch Vorgangstyp bzw. Prozeß bezeichnet eine bestimmte Art von ähnlichen Vorgängen. Marketingplan erstellen, Jahresabschluß anfertigen oder Schadensfallmeldung sind Beispiele für Workflowtypen und „Jahresabschluß 1996“ oder „Autoradio geklaut“ für Workflowinstanzen. Eine Schadensfallmeldung bei einer Versicherung könnte aus den Aufgaben Meldung erfassen, Prüfung des Schadensfalls, Anzahlungsanweisung und Abschlußmeldung bestehen.

Den einzelnen Aufgaben innerhalb eines Vorganges wird im allgemeinen ein Bearbeiter zugewiesen. Weiterhin wird eine Bearbeitungsposition definiert, d.h. es wird die Ausführungsreihenfolge der Arbeiten festgelegt. Die Kombination von Bearbeiter, Aufgabe und Bearbeitungsposition innerhalb eines Vorganges wird im folgenden als Arbeitsschritt bezeichnet. Vorgänge können als Graph dargestellt werden. Dabei entsprechen die Knoten den Arbeitsschritten, und die Kanten repräsentieren den Informationsfluß und die Ausführungsreihenfolge. Für das Beispiel der Schadensfallmeldung könnte der Graph dann wie in Abbildung 1 aussehen.

⁴ Vgl. [Lichtner, 1997, S.36]

⁵ Vgl. [Hollingsworth, 1996, S. 6]

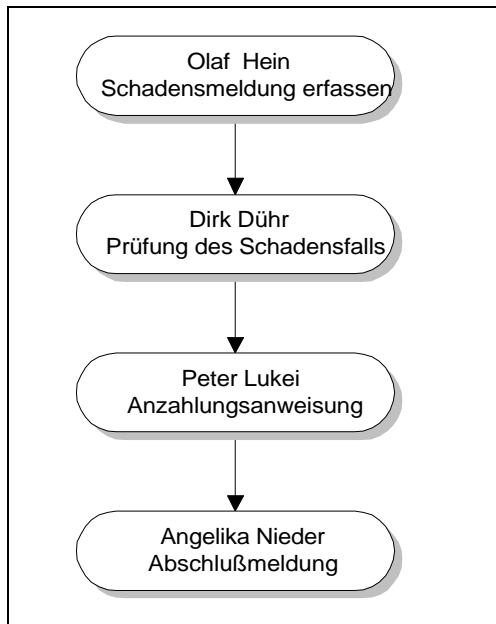


Abbildung 1: Schadensfallmeldung

Ein Workflow Management System (Workflowsystem) ist ein System, das die Automatisierung von sich wiederholenden Geschäftsprozessen ermöglicht. Es bietet die Möglichkeit, Workflowtypen zu modellieren. Aus diesen werden dann konkrete Vorgänge instanziiert. Deren Ausführung wird vom Workflowsystem unterstützt und überwacht. Dabei werden auch ad-hoc Ausnahmen, bei denen ein Bearbeiter vom vorgesehenen Bearbeitungspfad abweichen kann, erlaubt. Hierdurch kann flexibel auf unerwartete Situationen reagiert werden.

Hollingsworth definiert ein Workflowsystem wie folgt: „A system that completely defines, manages and executes ‘workflows’ through the execution of software whose order of execution is driven by a computer representation of the workflow logic.“⁶

2.3 Verteiltes Workflow Management

Herkömmliche Workflowsysteme erlauben meist nur das zentrale Workflow Management innerhalb einer Organisation. Dies ist dadurch gekennzeichnet, daß die beteiligten Personen, Abläufe und Datenbestände bekannt sind. Weiterhin steht das System unter der Kontrolle einer rechtlichen Organisation. Diese hat auch Einfluß auf die verwendete Hard- und Software. Wenn von Verteiltem Workflow Management oder

⁶ Vgl. [Hollingsworth, 1996, S. 6]

auch Distributed Workflow Management (DWM) gesprochen wird, sind diese Voraussetzungen nicht immer gegeben.

Die zusätzlichen Anforderungen können zu drei orthogonalen Dimensionen des Verteilten Workflowmanagements zusammengefaßt werden.⁷

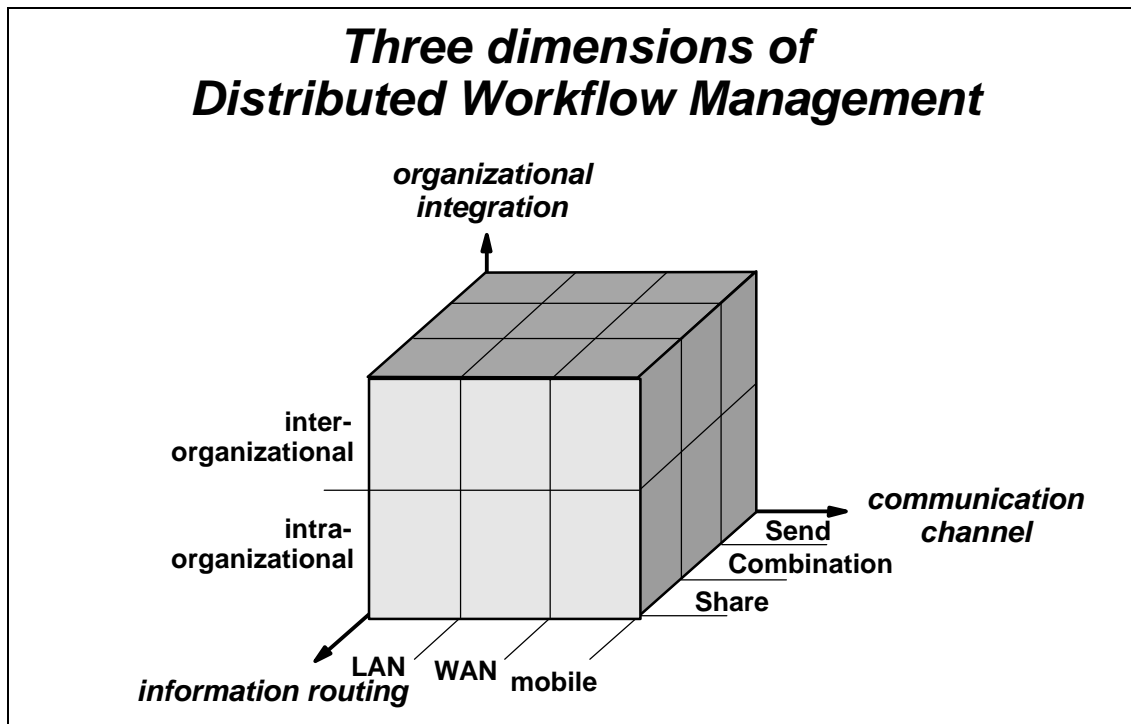


Abbildung 2: Drei Dimensionen des Verteilten Workflow Managements

Information Routing gibt dabei an, wie die Informationen transportiert werden. Es wird grundsätzlich zwischen dem Send und dem Share Modell unterschieden. Beim Send Modell werden Informationen aktiv vom Sender verteilt. Dies ist mit EMail oder der herkömmlichen Post zu vergleichen. Der Empfänger braucht nichts weiter zu tun als regelmäßig in seinen Briefkasten, seine Mailbox oder auf seinen Schreibtisch zu schauen. Das Send Modell kann für einfache Workflows eingesetzt werden. Es ist einfach zu bedienen und zu warten.

Beim Share Modell legt der Sender die Informationen an einem geeigneten Ort so ab, daß die potentiellen Empfänger darauf zugreifen können. Dies ist mit Schwarzen Brettern, an denen Informationen veröffentlicht werden, zu vergleichen. Aber auch das World Wide Web (WWW) oder gemeinsam genutzte Datenbanken arbeiten nach dem

⁷ Vgl. [Riemp; Nastansky, 1996]

Share Prinzip. Hierbei muß sich der Empfänger aktiv die Informationen beschaffen. Dazu schaut er zum Beispiel regelmäßig ans Schwarze Brett oder sieht sich die für ihn wichtigen Seiten im WWW an.

Für DWM kann das Share Modell verwendet werden, indem mehrere Kopien einer gemeinsam genutzten Datenbank erstellt werden. Diese Kopien können dann an verschiedenen Orten eingesetzt werden. In regelmäßigen Abständen werden die Daten der einzelnen Kopien untereinander abgeglichen. Dies erfordert aber ein hohes Vertrauensverhältnis unter den Benutzern der Kopien. Weiterhin wird bei allen das gleiche System vorausgesetzt.

Durch die Kombination beider Systeme kann ein leistungsfähiges verteiltes Workflowmanagement System geschaffen werden. Dabei werden verteilte Datenbanken verwendet, um die Informationen zu speichern und zu verwalten. Die Kommunikation zwischen den Datenbanken wird mit Hilfe von Message Objekten nach dem Send Prinzip realisiert. Diese Message Objekte bestehen aus der zu übermittelnden Information, Ausführungsintelligenz, dem Layout und einer Wegbeschreibung. Dabei können sowohl strukturierte Informationen als auch unstrukturierte Daten wie zum Beispiel formatierter Text und Diagramme übertragen werden. Das Layout sorgt dafür, daß die Informationen am Ziel genau so dargestellt werden, wie es der Absender vorgesehen hat. Diese Message Objekte können mit Compound Documents⁸, wie sie Notes zur Verfügung stellt, erzeugt werden.

⁸ Semi Strukturierte Dokumente, die aus strukturierten Feldern, RichText, Code und dem Layout bestehen

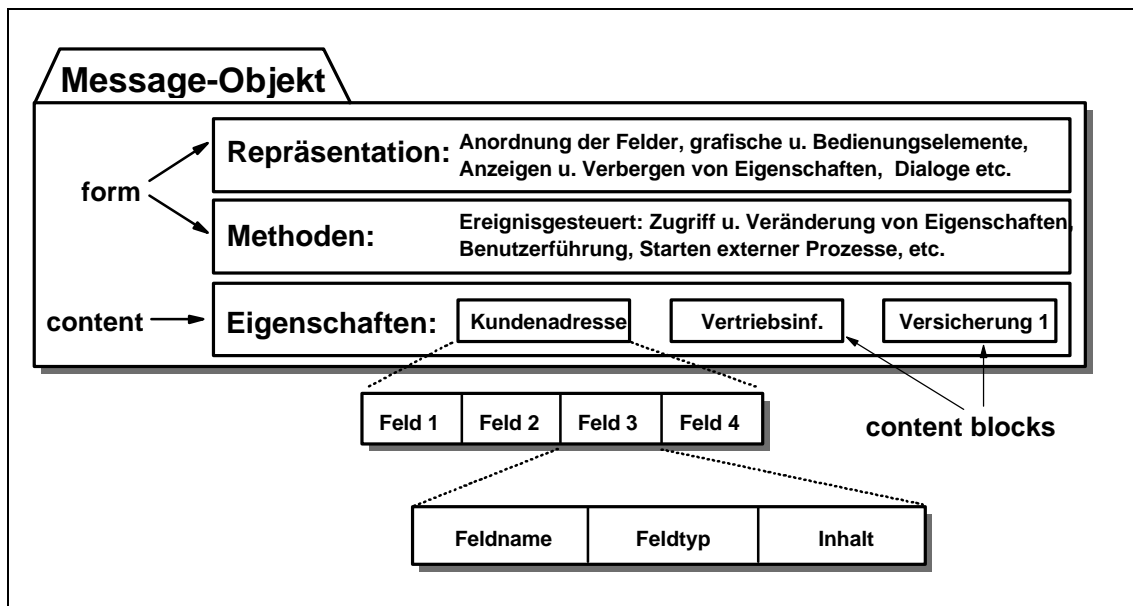


Abbildung 3: Message Objekt

Als Kommunikationskanal für DWM kommt ein Wide Area Network (WAN) wie zum Beispiel das Internet oder das Telefonnetz in Betracht. Diese Netze sind im Vergleich zu einem Local Area Network (LAN) sehr langsam. Da sie im allgemeinen nicht dem Benutzer sondern einem Netzbetreiber gehören, muß für deren Nutzung bezahlt werden. Daher muß darauf geachtet werden, daß die Datenmengen möglichst klein bleiben, um die Übertragungszeiten und Kosten gering zu halten. Da der Benutzer keinerlei Kontrolle über das Netz hat, ist es auch wichtig, die Daten vor Mißbrauch und Spionage zu schützen. Aus diesen Gründen ist ein Content Management bei DWM sinnvoll. Das Content Management sorgt dafür, daß nur die Informationen übermittelt werden, die für die Ausführung einer Aufgabe auch benötigt werden. Weder vertrauliche noch überflüssige Informationen werden übertragen. Das Content Management erstellt aus den internen Vorgangsdokumenten Message Objekte. Diese werden dann an den Empfänger geschickt.

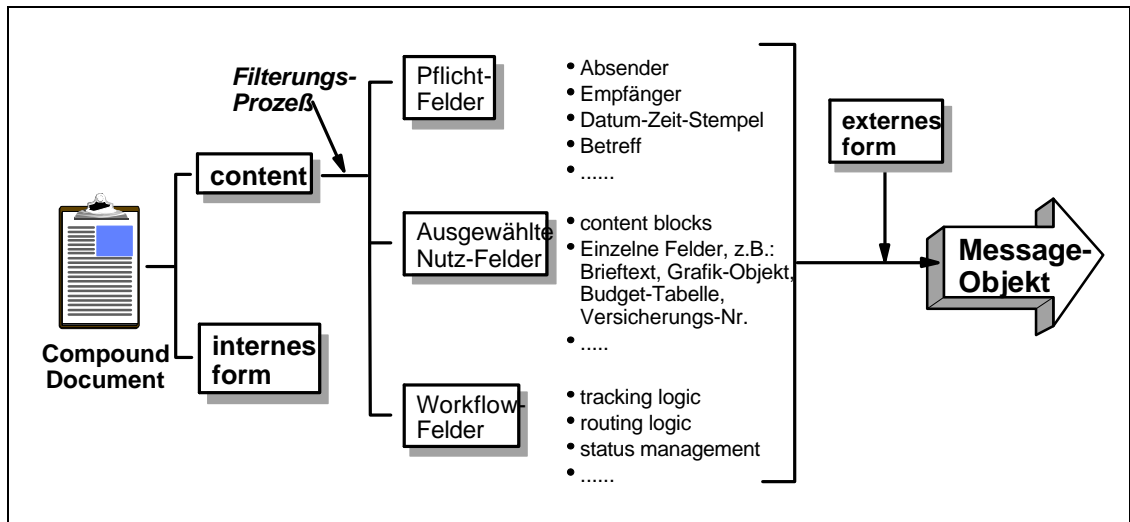


Abbildung 4: Content Management

Beim DWM kann zwischen internen und externen Vorgängen unterschieden werden. Interne Vorgänge werden vollständig innerhalb einer Organisation bearbeitet. Sie werden auch „intraorganizational“ genannt. Wenn Vorgänge von mindestens zwei rechtlich getrennten Organisationen bearbeitet werden, nennt man sie extern oder auch „interorganizational“. Je nach Vorgang werden verschiedene Anforderungen an das WFMS gestellt. Diese betreffen hauptsächlich das Sicherheitsbedürfnis der beteiligten Organisationen. Je größer das Vertrauen der Unternehmen untereinander ist, desto geringer sind diese Bedürfnisse. Bei einer Kommunikation zwischen Kunde und Lieferant wird das Sicherheitsbedürfnis im allgemeinen sehr hoch sein. Bei Profit Centern oder virtuellen Organisationen nicht ganz so hoch und zwischen einzelnen Niederlassungen einer Organisation eher gering.

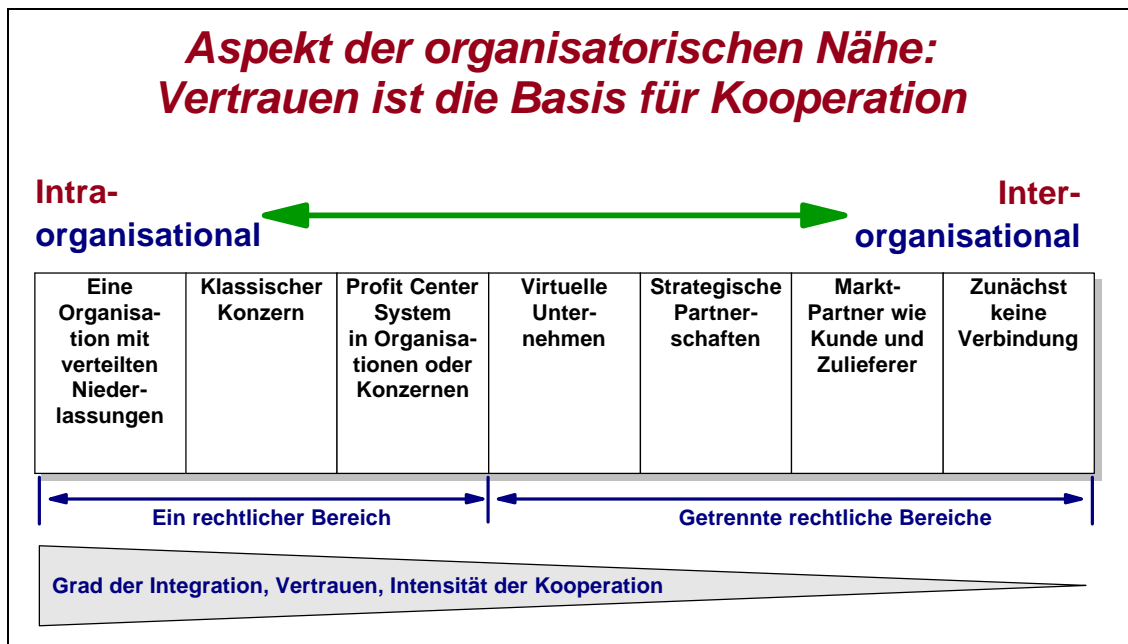


Abbildung 5: Aspekte der organisatorischen Nähe

Abbildung 5 zeigt mögliche Organisationsformen. Von links nach rechts steigt dabei das Sicherheitsbedürfnis bei der Kommunikation untereinander, und der Grad der Integration, das gegenseitige Vertrauen und die Intensität der Kooperation sinken.

2.4 Wide Area GroupFlow System

Ziel des Wide Area GroupFlow Systems ist es, ein organisationsübergreifendes Workflowmanagement zu ermöglichen. Dabei werden die Anforderungen an verteiltes Workflowmanagement, wie sie im vorigen Kapitel beschrieben wurden, erfüllt. Das WAGS Architekturkonzept soll hier kurz vorgestellt werden.

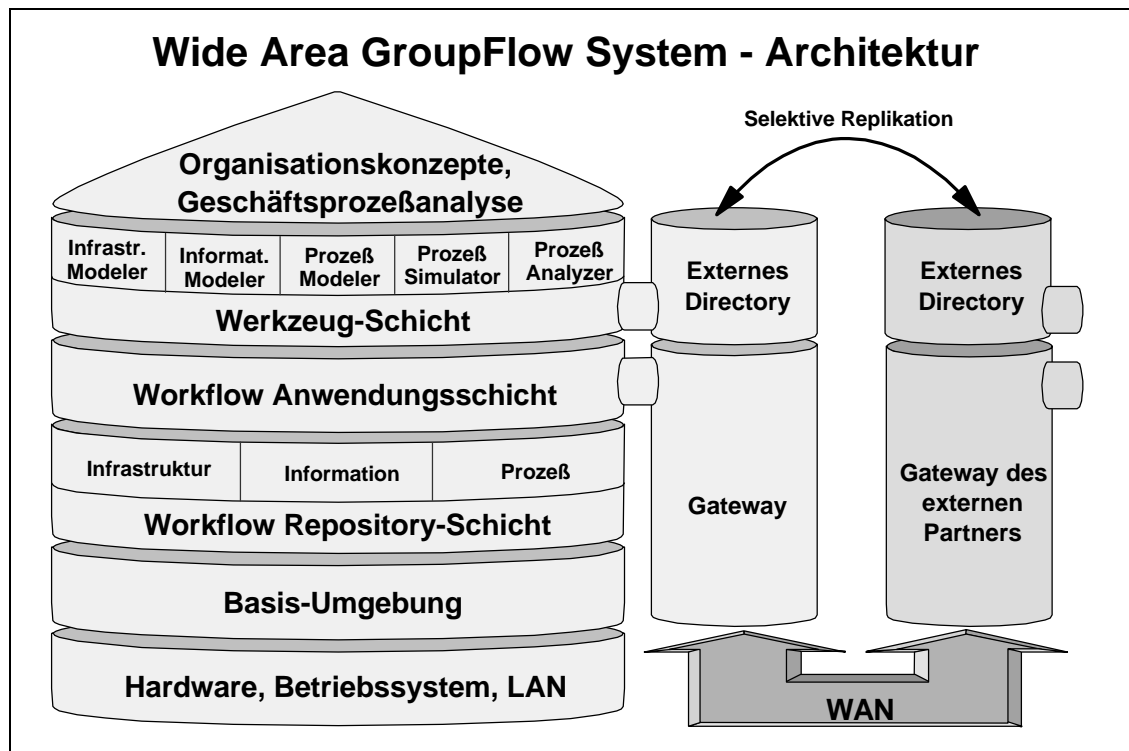


Abbildung 6: Wide Area GroupFlow System - Architektur

Das ganze System besteht aus mehreren, aufeinander aufbauenden Schichten. Als Basis-Umgebung wurde Lotus Notes gewählt. Lotus Notes ist für die verschiedensten Betriebssysteme auf unterschiedlichen Hardwareplattformen verfügbar und unterstützt die gängigsten Netzwerkprotokolle. Dadurch wird die unterste Schicht, die aus der Rechnerhardware, dem Betriebssystem und dem Netzwerk besteht, verborgen und das WAGS kann in heterogenen Umgebungen betrieben werden. Darauf baut die Workflow Repository-Schicht auf. Sie besteht aus den Daten der Aufbauorganisation (Infrastruktur), den operativen Daten (Information) und der Prozeßstruktur (Prozeß). Darauf baut die Workflow Anwendungsschicht auf. Sie initiiert und kontrolliert Workflowinstanzen. Dazu greift sie auf die Informationen in der Workflow Repository-Schicht zurück. Die Werkzeug-Schicht stellt Hilfsmittel für die Konfiguration der darunter liegenden Schichten zur Verfügung. Diese Hilfsmittel haben eine graphische Benutzeroberfläche und müssen zum Teil aufwendige Berechnungen durchführen. Daher sind sie nicht als Notes Datenbanken realisiert. Es handelt sich um C, C++ oder Visual Basic Programme. Diese Programme wurden und werden für PCs mit dem Betriebssystem Windows 3.1x, Windows 95 oder Windows NT entwickelt. Dadurch sind sie nicht mehr plattformunabhängig. Da diese Werkzeuge aber nur zur Administration

und nicht zum Betrieb benötigt werden, ist dieser Nachteil nicht so gravierend und wird durch die erweiterten Möglichkeiten ausgeglichen.

Mit dem OrgaModeler kann die Organisationsstruktur bearbeitet werden. Der WAGS Modeler dient zur Definition von Workflowtypen. Für die ex ante Analyse steht der Prozeß Simulator und für die ex post Analyse der WAGS Analyzer zur Verfügung.

Die oberste Schicht wird von den Organisationskonzepten und der Geschäftsprozeßanalyse gebildet. Sie bestimmen die Gestaltung des gesamten Workflow Management Systems.

Daneben existieren ein Gateway und ein External Directory. Sie dienen als Schnittstelle zwischen Organisationen. Die Gateway Datenbanken der verschiedenen Organisationen werden über ein WAN verbunden. Sie wickeln den organisationsübergreifenden Datenaustausch ab. Wichtige Aufgaben dabei sind das Routing, Content Management und Tracking⁹ von E-Mails oder Message Objekten. Das External Directory ist eine Lotus Datenbank, die als Adreßbuch verwendet wird. Es enthält globale Routing Informationen (GRI). Das sind abstrakte Adressen und Beschreibungen von Prozessen. Diese Prozesse können von anderen Organisationen in Workflows verwendet werden. Weiterhin enthält das External Directory zu den GRIs interne Routing Informationen (IRI). Diese Daten werden benötigt, um eingehende Informationsobjekte wieder in den Workflow einzufügen. Wenn ein eingehendes Informationsobjekt als Adresse eine GRI enthält, dann wird diese von der Gateway Datenbank durch die IRI ersetzt. Aus der IRI geht die Aufgabe und der Bearbeiter hervor, so daß der Workflow fortgesetzt werden kann.

⁹ Aufzeichnen von Informationen der Vorgangsbearbeitung zur späteren Kontrolle und Analyse. Vgl. [Wolke, 1994]

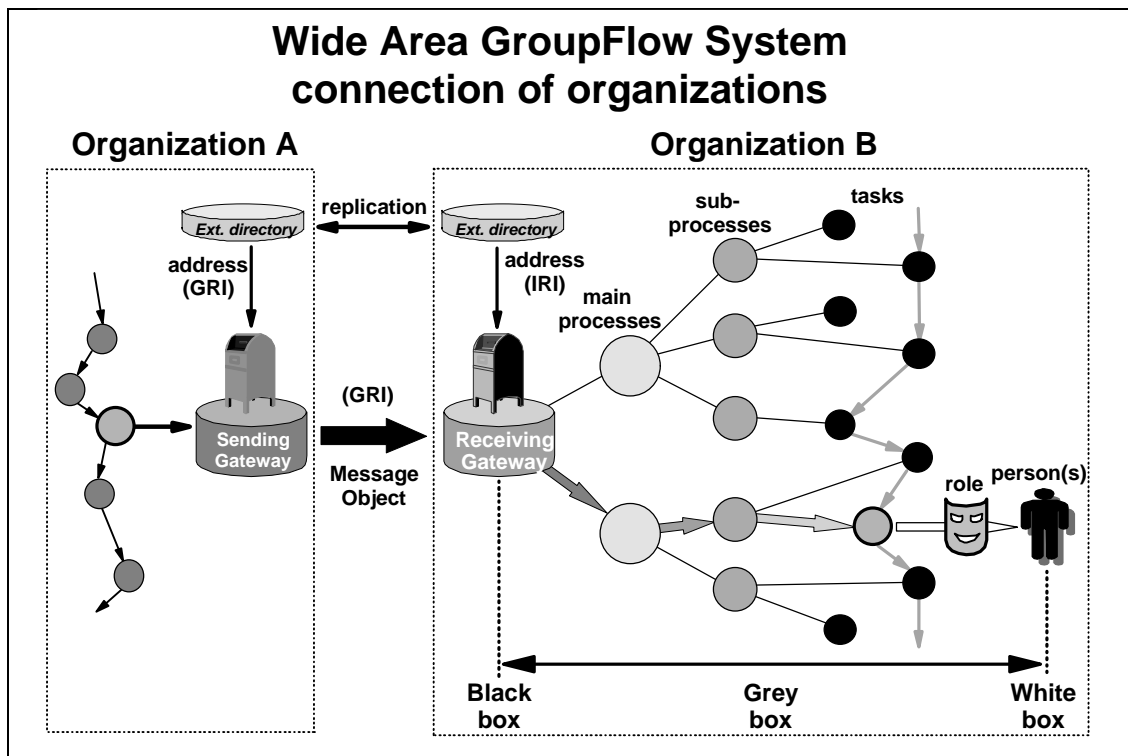


Abbildung 7: Wide Area GroupFlow System

Eine Erweiterung des Beispiels einer Schadensfallmeldung soll die Funktionsweise verdeutlichen. Sei Organisation A eine Autovermietung und Organisation B ein Autoversicherer. Bei der Autovermietung kommt es immer wieder zu Schadensfällen. Um diese effizient abwickeln zu können, wurde mit der Versicherung ein Verteiltes WFMS installiert. Die Versicherung veröffentlicht dazu im External Directory die Adresse für die Aufgabe Schadensfallmeldung. Kommt es bei der Autovermietung zu einem Schadensfall wird dort eine Workflowinstanz erzeugt. Zuerst sind einige Aufgaben innerhalb der Autovermietung zu erledigen. Dazu gehört zum Beispiel das Ausfüllen eines Unfallberichtes und das Prüfen der abgeschlossenen Versicherung. An einem Punkt im Workflow wird dann die Schadensfallmeldung der Versicherung angestoßen. Es wird aus dem Workflow heraus ein Message Objekt erzeugt, das den empfangenen Prozeß bei der Versicherung adressiert. Hierzu wird die GRI aus dem External Directory verwendet. Im Sending Gateway wird die Adresse des Receiving Gateway aus dem External Directory gelesen und ins Message Objekt eingetragen. Weiterhin wird eine inhaltliche Filterung durchgeführt, und es werden Trackinginformationen gespeichert. Dann wird das Objekt abgeschickt. Bei der Versicherung wird die GRI durch die IRI aus dem External Directory ersetzt und der Prozeß gestartet. Hier werden dann die Aufgaben

Meldung erfassen, Prüfung des Schadensfalls, Anzahlungsanweisung und Abschlußmeldung bearbeitet. Weiterhin wird eine Meldung an die Autovermietung erzeugt und zurückgeschickt. Ist diese angekommen wird auch dort der Workflow fortgesetzt.

2.5 Lotus Notes als Basis-Umgebung des Wide Area GroupFlow Systems

Lotus Notes (Notes) ist ein dokumentenorientiertes Datenbanksystem. Es ist nicht relational. Die Stärken liegen nicht in der Verwaltung von strukturierten Informationen. Vielmehr ist Notes sehr gut geeignet, unstrukturierte Informationen, wie Texte, Grafiken oder Diagramme zu verwalten. Im allgemeinen besteht eine Notes Installation aus einem oder mehreren Servern, von denen die Datenbanken verwaltet werden und mehreren Arbeitsplatzrechnern (Clients), von denen aus die Anwender auf die Datenbanken auf den Servern zugreifen. Eine Stärke des Systems besteht in der Replikation von Datenbanken. Dabei wird von einer Datenbank eine spezielle Kopie (Replik) erzeugt. Werden an den einzelnen Repliken einer Datenbank Änderungen durchgeführt, dann merkt sich Notes diese Änderungen. Bei einem späteren Datenabgleich zwischen zwei Repliken (Replikation) werden dann beide Datenbanken aktualisiert. Dadurch ist es möglich, auch ohne die Verbindung zu einem zentralen Server mit den Datenbanken zu arbeiten. Daraus ergeben sich vielfältige Möglichkeiten. So können auch Clients, die nicht ständig mit dem Server verbunden sind, mit den Datenbanken arbeiten. Datenbanken können auch auf Server an verschiedenen Standorten verteilt werden. Teure Standleitungen zwischen den Standorten sind dabei nicht nötig.

Datum	Status	Typ	Arb. mögl.	Wiederholbar	Spezialist
15.12.95	Benutzerservice	Problem PC	Unbekannt	Unbekannt	
15.12.94	Intern	Problem	Eingeschränkt möglich	Unregelmäßig	Jörg Hain (Host)
15.12.94	Erledigt neue Lösung	Info	Eingeschränkt möglich	Nicht wiederholbar	
15.12.94	Erledigt gelöst	Problem	Möglich	Wiederholbar	
15.12.94	Intern	Problem	Nicht möglich	Wiederholbar	
15.12.94	Extern	Problem	Möglich	Wiederholbar	
15.12.94	Intern	Problem	Eingeschränkt möglich	Wiederholbar	
14.12.94	Intern	Problem	Eingeschränkt möglich	Unregelmäßig	
13.12.94	Erledigt neue Lösung	Problem	Eingeschränkt möglich	Wiederholbar	
13.12.94	Erledigt gelöst	Problem	Eingeschränkt möglich	Wiederholbar	Hans Schultze-Neumann (Anwen)
13.12.94	Erledigt neue Lösung	Problem	Nicht möglich	Wiederholbar	Rudi Bümmel (Notes)
13.12.94	Erledigt neue Lösung	Problem	Nicht möglich	Wiederholbar	Jörg Hain (Host)
13.12.94	Erledigt neue Lösung	Problem	Nicht möglich	Wiederholbar	Marion Baumann (Netze)

Abbildung 8: Lotus Notes Ansicht

Eine Datenbank in Notes ist eine Aufnahme-, Speicher- und Verwaltungseinheit für Dokumente (Note). Die Dokumente einer Datenbank werden in einer oder mehreren Ansichten (Views) dargestellt. Dies sind Listen, in denen einige Informationen der Dokumente dargestellt werden können. Über die Einträge der Liste kann dann auf das eigentliche Dokument zugegriffen werden. Die Dokumente enthalten die eigentlichen Informationen. Innerhalb eines Dokuments werden die Informationen in Feldern abgelegt. Die Dokumente entsprechen den Datensätzen einer relationalen Datenbank und die Felder den Attributen. Im Unterschied zu relationalen Datenbanken können Notes Dokumente aber auch Rich Text¹⁰ Felder enthalten. Diese können umfangreiche formatierte Texte, Tabellen, Grafiken, OLE-Objekte¹¹, Dateianhänge etc. enthalten. Dargestellt werden diese Informationen mit Hilfe von Masken. Eine Maske bestimmt,

¹⁰ Textformat für formatierbaren Text

¹¹ OLE (Object Linking and Embedding) ist eine von Microsoft entwickelte objekt-orientierte Technologie zum Datenaustausch zwischen Programmen. Dadurch können unter anderem Objekte (z.B. Tabellen und Grafiken) in andere Programme eingefügt werden. vgl. [Thomas Lauer, 1993]

welche Felder wie angezeigt werden. Weiterhin können Masken ebenso wie Felder Verarbeitungsintelligenz besitzen.

Über eine Makrosprache oder Lotus Scrip¹² können Notes Datenbanken programmiert werden. Dadurch können zum Beispiel Feldinhalte automatisch berechnet oder Eingaben überprüft werden. Es besteht aber auch die Möglichkeit so komplexe Anwendungen, wie ein Workflowsystem zu entwickeln.

Notes stellt umfangreiche Möglichkeiten bereit, um die Daten in den Datenbanken vor unbefugtem Zugriff zu schützen. So benötigt jeder Benutzer, der mit Notes Datenbanken arbeiten will, eine ID Datei. In dieser sind relevante Benutzerdaten gespeichert. Dies sind unter anderem der Name des Benutzers, die Server, auf denen er zugelassen ist, sein öffentlicher und sein privater Schlüssel. Die Benutzerdatei wird durch ein Kennwort geschützt. Nur wenn ein Benutzer auf einem Server zugelassen ist, darf er auf diesen zugreifen. Der Zugriff auf einzelne Datenbanken wird über deren Zugriffskontrollliste kontrolliert. Diese enthält für die verschiedenen Zugriffsebenen Listen der Benutzer oder Benutzergruppen, denen der Zugriff erlaubt ist. Die Zugriffsebenen bestimmen, welche Aktionen ein Benutzer ausführen darf. Mögliche Aktionen sind das Erstellen oder Löschen von Dokumenten, das Bearbeiten von eigenen oder fremden Dokumenten oder das Vergeben von Zugriffsrechten. Auch innerhalb einer Datenbank kann der Zugriff auf einzelne Dokumente beschränkt werden.

Weiterhin wird von Notes ein Leistungsfähiges Mail System zur Verfügung gestellt. Jeder Benutzer erhält eine persönliche Mail Datenbank. Diese ist eine Notes Datenbank, die um Funktionen für das Senden und Empfangen von EMail erweitert wurde.

¹² Von Lotus entwickelter Basic Dialekt, der auch ein Klassenkonzept mit Vererbung beinhaltet.

3 Motivation

3.1 Ziele einer Workflow Analyse

Bevor ein Analysewerkzeug entworfen werden kann, sollte zuerst erarbeitet werden, welche Ziele mit der Analyse erreicht werden sollen. Als erster Anhaltspunkt dient dabei der Workflow Lebenszyklus, der im folgenden kurz erläutert wird.

3.1.1 Workflow Lebenszyklus

Der Workflow Lebenszyklus beschreibt das gesamte Vorgehen bei der Geschäftsprozeßmodellierung. Er basiert auf dem Konzept der kontinuierlichen Verbesserung. Am Anfang steht dabei die Analyse des Prozesses, der modelliert werden soll. Die Prozeßstruktur muß erkannt werden. Dabei kann es sich sowohl um einen papierbasierten als auch elektronisch unterstützten Prozeß handeln. Dazu müssen die einzelnen Aufgaben des Prozesses und ihre Abhängigkeiten untereinander identifiziert werden. Mit Hilfe der gewonnenen Informationen wird dann ein Workflowtyp modelliert. Der so modellierte Workflow wird anschließend simuliert. Mit Hilfe der Simulationsergebnisse kann die Modellierung anschließend wiederholt und verbessert werden. Dieser sich wiederholende Prozeß wird durchgeführt, bis die Simulation befriedigende Ergebnisse liefert. Danach können Instanzen des Workflows erzeugt und bearbeitet werden. Diese speichern während der Ausführung Informationen zur späteren Analyse ab. Der tatsächliche Ablauf wird anschließend analysiert, um mit den Ergebnissen die Modellierung weiter zu verbessern.

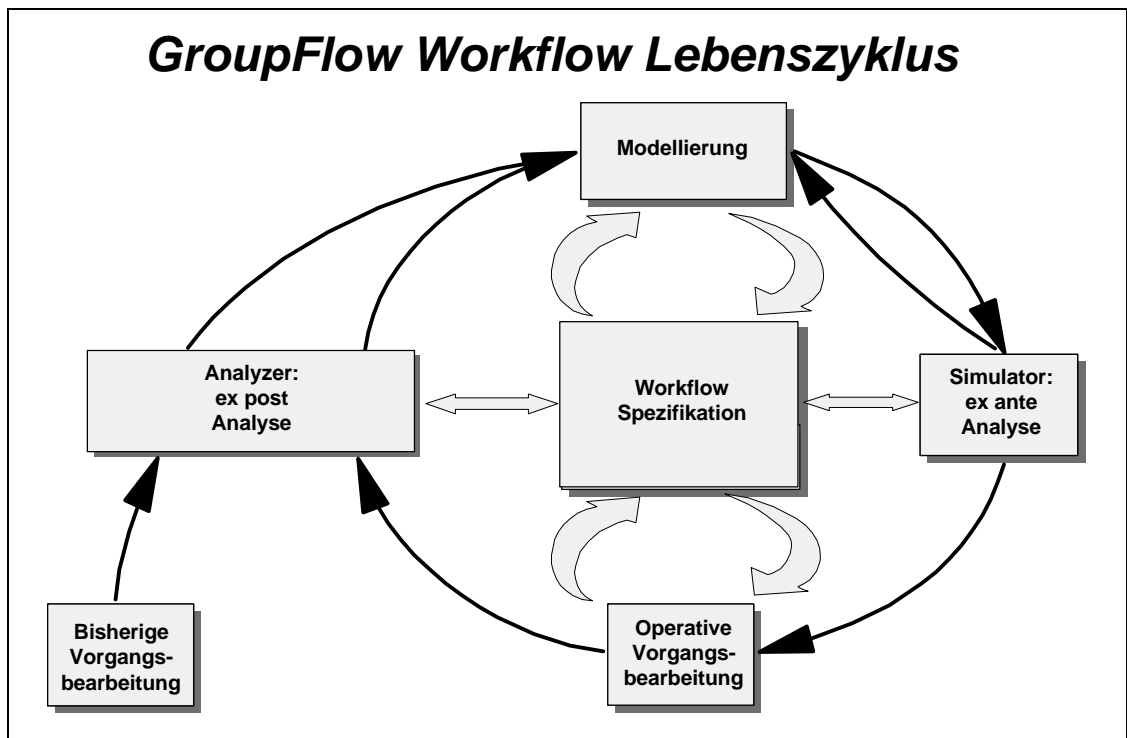


Abbildung 9: Workflow Lebenszyklus

Für ein Analysewerkzeug bedeutet dies, daß es mit zwei unterschiedlichen Arten von Eingabedaten arbeiten muß. Für die erste Analyse von Prozessen müssen Daten über die bisherige Vorgangsbearbeitung ausgewertet werden. Dies ist in der Regel nur möglich, wenn Daten über die bisherige Vorgangsbearbeitung in elektronischer Form vorliegen. Bei der Kontrolle bearbeiteter Workflowinstanzen sind die Daten aus dem Workflow System auszuwerten. Im ersten Fall sollten möglichst wenig Annahmen über die zugrunde liegende Datenstruktur gemacht werden, um ein möglichst flexibles Werkzeug zu erhalten. Aus den vorhandenen Daten sollten dann natürlich entstandene Abläufe zu erkennen sein. Ziel sollte es sein, die vorhandene Prozeßstruktur zu analysieren um einen neuen Workflowtyp definieren zu können. Im zweiten Fall sind die Datenstrukturen der Eingabedaten bekannt bzw. beeinflubar. Bei der Analyse wird der Soll- mit dem Istzustand des Systems verglichen. Abweichungen müssen erkennbar und identifizierbar sein, damit der Ursache auf den Grund gegangen werden kann. Ziel ist es, mit diesen Informationen den Workflow zu verbessern.

3.1.2 Workflow Referenz Modell der Workflow Management Coalition

Von der Workflow Management Coalition (WFMC) wurde ein Workflow Referenz Modell vorgeschlagen. Der Zweck dieses Referenzmodells ist es, einen Rahmen für die

Entwicklung von Workflow Management Systemen zu liefern. Das Modell beschreibt unter anderem Eigenschaften, Komponenten, Architektur und Begriffe von WFMS.

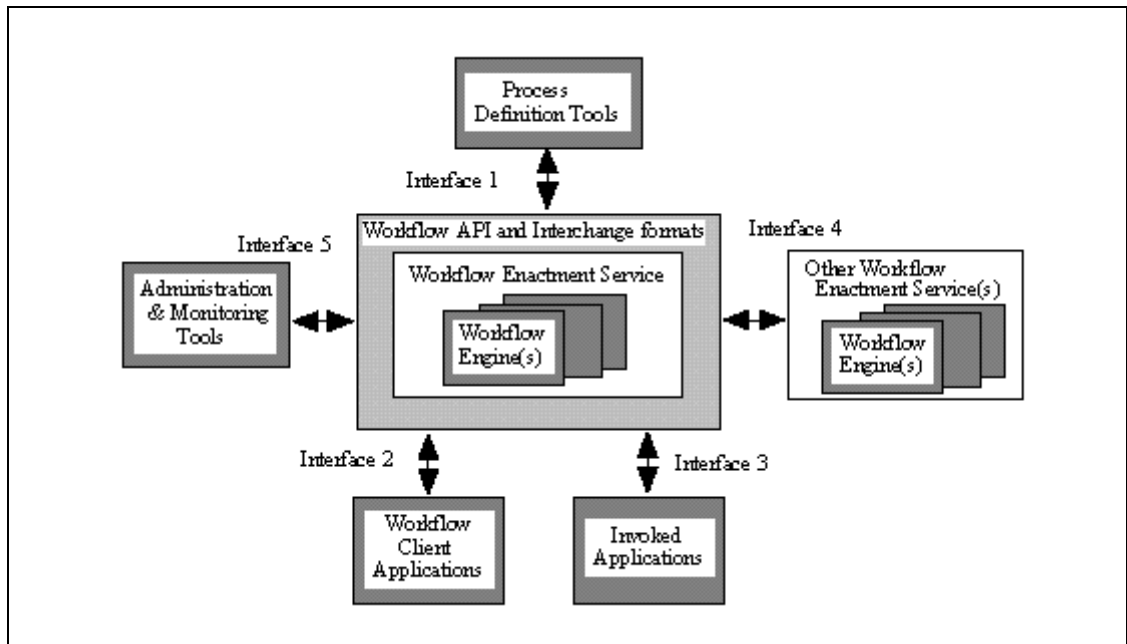


Abbildung 10: Workflow Referenz Modell

Nach diesem Modell besteht ein WFMS aus den in Abbildung 10 abgebildeten Komponenten. Diese kommunizieren über definierte Schnittstellen miteinander. Eine zentrale Rolle spielt dabei das Workflow API and Interchange formats (WAPI). Es stellt eine Menge von Funktionen zur Verfügung, mit denen auf das Workflowsystem zugegriffen werden kann. Weiterhin wird durch das WAPI die Kommunikation zwischen den einzelnen Systemkomponenten reguliert.

Für diese Arbeit sind insbesondere das Interface 5 und die Administration & Monitoring Tools von Interesse. Durch die Monitoring Werkzeuge soll der Status von Vorgängen überwacht werden können. Weiterhin haben sie die Aufgabe quantitative Informationen der Workflowinstanzen aufzubereiten und anzuzeigen. Mit den Administrationswerkzeugen kann über Funktionen, die das WAPI Interface 5 zur Verfügung stellt, das Workflowsystem konfiguriert und administriert werden. Es folgen einige der Funktionen die vom Interface 5 angeboten und von den Monitoring und Administrationswerkzeugen genutzt werden können:

- Benutzermanagement
 - Verwaltung von Benutzern und Gruppen mit deren Rechten

- Rollenmanagement
Definition von Rollen sowie die Verwaltung der Rollen-Benutzer bzw. Gruppen Zuordnung
- Ressourcenkontrolle
Überwachung der Auslastung von Ressourcen
- Statusfunktionen
Anzeige der Detailinformationen von Workflowinstanzen

3.1.3 GroupFlow und der GroupFlow Analyzer

GroupFlow ist ein an der Universität-Gesamthochschule Paderborn entwickeltes Workflow Management System. Es basiert auf der Groupwareplattform Lotus Notes und ermöglicht Workflow Management innerhalb einer Organisation. Im Rahmen dieses Projektes wurde bereits ein Analysewerkzeug entwickelt¹³. Zur Analyse einer Datenbank wird aus den Dokumenten einer Notes Ansicht das \$UpdatedBy Feld ausgelesen und ausgewertet. In diesem Feld speichert Notes automatisch die Namen der Bearbeiter eines Dokuments in einer chronologisch sortierten Liste. Zur Visualisierung werden die Bearbeiter und ihre Kommunikationsbeziehungen in den drei Ansichten Workflow-Steps, Workflow-Users und Workflow-Agent dargestellt. Dazu wird für jede dieser Ansichten ein gerichteter Graph erzeugt und angezeigt. Die Knoten des Graphen sind die Bearbeiter der Vorgänge und die Kanten die Kommunikationsbeziehungen zwischen den Bearbeitern. Die einzelnen Ansichten unterscheiden sich in der Auswahl der darzustellenden Bearbeiter. Bei der Workflow-Users Ansicht wird jeder Bearbeiter genau einmal dargestellt. In der Workflow-Steps Ansicht werden die Bearbeiter zusätzlich durch ihre Position in der Bearbeitungsfolge der Dokumente unterschieden. Dadurch werden für einen Bearbeiter, der ein Dokument als erster und ein anderes als zweiter bearbeitet hat, zwei Knoten im Graphen erzeugt. Für die Workflow-Agent Ansicht wird ein Bearbeiter ausgewählt und in den Mittelpunkt der Betrachtung gestellt. Es werden nur die Kommunikationsbeziehungen dieses Bearbeiters betrachtet. Für die weitere Auswertung stehen Diagramme und eine Tabelle zur Verfügung. Mit diesen Hilfsmitteln ist es möglich, natürlich entstandene Bearbeitungsfolgen innerhalb von Datenbanken zu

¹³ Vgl. [Piesk, Pötter, 1994]

erkennen. Die gewonnenen Informationen können zur Modellierung von Prozessen verwendet werden.

3.1.4 Weitere Ansätze in der Literatur

Gérard Derszteler beschreibt einen WorkFlow Management Cycle (WMC)¹⁴. Dieser ähnelt dem GroupFlow Workflow Lebenszyklus sehr. Er basiert auch auf dem Konzept der kontinuierlichen Verbesserung. Ein wesentlicher Teil dabei ist der WorkFlow Analyzer, der auf einem Kennzahlenmodell zur Bewertung von Prozeßstrukturen basiert. Der Analyzer sorgt für die Rückkopplung im Kreislauf. Während der Ausführung werden von den Workflowinstanzen Kenngrößen, wie z.B. die Durchlaufzeit protokolliert. Aus diesen Größen werden weitere Kennzahlen abgeleitet. Weiterhin werden die Werte verdichtet. Dazu werden über fest vorgegebene Zeiträume Statistische Werte wie der Durchschnitt oder Minimum und Maximum berechnet. Diese Kenngrößen können dann mit dem Analyzer visualisiert und mit den Solldaten verglichen werden.

3.2 Anforderungen an einen Workflow Analyzer

3.2.1 Visualisierung von Abläufen und Prozeßstrukturen

Prozesse lassen sich als gerichtete Graphen darstellen. Die Knoten entsprechen dabei den Arbeitsschritten. Sie repräsentieren einen Bearbeiter, eine Aufgabe und eine bestimmte Position in der Bearbeitungsfolge. Die Kanten bestimmen den oder die Nachfolger des Arbeitsschrittes. Während der Modellierung eines Workflowtyps wird solch ein Graph erzeugt. Es werden für die einzelnen Knoten Aufgaben festgelegt und Bearbeiter bestimmt. Die Bearbeiter sind meist keine Personen sondern Rollen oder Gruppen. Danach wird die Ausführungsreihenfolge der einzelnen Arbeitsschritte bestimmt. Diese wird durch gerichtete Kanten dargestellt. Für die einzelnen Kanten können dann noch Bedingung festgelegt werden. Diese Bedingungen müssen erfüllt sein, damit bei der Bearbeitung der Workflowinstanz diese Kante verwendet wird. Wird eine Instanz dieses Workflow Typs ausgeführt, dann muß sich nicht immer an diese Definition gehalten werden. Die einzelnen Arbeitsschritte werden von Personen bearbeitet. Weiterhin werden nicht immer alle möglichen Wege und Aufgaben verwendet. Zusätzlich kann es zu ad-

¹⁴ Vgl. [Derszteler, 1997]

hoc Ausnahmen kommen, bei denen völlig andere Bearbeitungswege eingeschlagen und andere Aufgaben ausgeführt werden als im Workflowtyp definiert wurden. Dieser tatsächliche Ablauf der Vorgänge soll visualisiert werden. Dabei ist darauf zu achten, daß nicht immer alle Informationen für die Konstruktion des Graphen eines Vorganges vorhanden sind. Es kann sein, daß keine Information über den tatsächlichen Bearbeiter oder die ausgeführte Aufgabe vorhanden ist. Weiterhin muß beachtet werden, daß speziell bei verteiltem Workflow nur ungenaue Informationen über den tatsächlichen Ablauf vorhanden sind. Andere Organisationen werden nur im Ausnahmefall bereit sein, genaue Angaben über den internen Ablauf ihrer Workflows preiszugeben. Es soll aber möglich sein, externe Arbeitsschritte zu erkennen. Spezielle Systemkomponenten, wie das Sending und Receiving Gateway sollen als Bearbeiter eines Vorganges darstellbar sein.

3.2.2 Statistische Auswertung von Vorgangsdaten

Es ist eine Statistische Auswertung der Daten vorzusehen. Diese sollte durch Tabellen und Diagramme erfolgen. Dabei sollen die Daten der erzeugten Graphen übersichtlich dargestellt werden. Von besonderem Interesse sind dabei Auswertungen, in denen Bearbeiter und bearbeitete Aufgaben in Verbindung gebracht werden. Die Informationen aus diesen Auswertungen sollen für die Modellierung neuer oder Verbesserung vorhandener Workflowtypen verwendet werden können. Weiterhin sollen Informationen über die Ressourcenauslastung ausgewertet werden können.

3.2.3 Schnittstellen zum Workflow System

Die Schnittstelle zum Workflow System sollte möglichst flexibel sein. Da sich WAGS noch in der Entwicklung befindet, kann es in Zukunft zu Designänderungen kommen. Die Schnittstelle sollte so gestaltet werden, daß bei Änderungen des restlichen Systems nicht zwangsläufig der Analyzer angepaßt werden muß. Weiterhin darf die Schnittstelle die Funktionalität des Systems nicht einschränken.

Vom Workflow-System werden nicht immer alle Informationen über einen Vorgang und die bearbeiteten Arbeitsschritte für eine spätere Analyse abgespeichert (getrackt). So ist es denkbar, daß nur der Name der Aufgabe aber nicht der Bearbeiter zur Verfügung steht. Insbesondere bei Aufgaben, die innerhalb anderer Organisationen ausgeführt

werden, kann nicht davon ausgegangen werden, daß ausführliche Informationen zur Verfügung stehen. Durch die verteilte Ausführung der Workflows können auch Trackinginformationen verloren gehen. Diese Situationen sind bei der Analyse zu berücksichtigen.

Die Auswertung des \$UpdatedBy Feldes, wie sie vom GroupFlow Modeler durchgeführt wurde, sollte weiterhin durchführbar sein. Dadurch können natürlich entstandene Workflows in Notes Datenbanken erkannt werden. So können auch Datenbanken analysiert werden, in denen normalerweise nur ad-hoc Workflows stattfinden.

3.3 Inhaltliche Abgrenzung

Der GroupFlow Analyzer dient für den WAGS Analyzer als Vorlage. Der Quellcode konnte aber nur für konzeptionelle Anregungen genutzt und nicht direkt weiter verwendet werden. Dies liegt an der benutzten Programmiersprache. Der GroupFlow Analyzer wurde in Visual Basic 3.0 implementiert. Der WAGS Analyzer wurde bzw. wird in Visual Basic 5.0¹⁵ entwickelt. Diese Version stellt im Gegensatz zur Version 3.0 Klassen¹⁶ und Polymorphismus¹⁷ zur Verfügung. Dadurch ist es möglich, objektbasiert zu programmieren. Eine Verwendung des alten Codes hätte zu umfangreichen Anpassungen geführt.

Der Schwerepunkt liegt nicht auf der Implementierung aller Details des GroupFlow Analyzers, sondern auf der Entwicklung neuer Funktionen. Insbesondere sollen die Bedürfnisse von verteilten Workflows berücksichtigt werden.

¹⁵ Vgl. Kapitel 5.1.1

¹⁶ „Eine Klasse ist eine Menge von Objekten, die eine gemeinsame Struktur und ein gemeinsames Verhalten aufweisen.“ Vgl. [Booch, 1994, S. 136]

¹⁷ Objekte verschiedener Klassen können auf eine gemeinsame Menge von Operationen reagieren. Vgl. [Booch, 1994, S.98]

4 Konzeption und Lösungsansätze für eine Analyse verteilter Vorgangsbearbeitung

Bei der Modellierung von Workflows wird der zu modellierende Prozeß in einzelne Aufgaben zerlegt. Einer Aufgabe kann zur Bearbeitung eine Abteilung, Rolle, Arbeitsgruppe, Person, Position oder ein Software-Agent zugewiesen werden. Die Kombination aus einer Aufgabe und einem Bearbeiter bildet einen Arbeitsschritt. Dieser wird durch ein Icon innerhalb des Modelers dargestellt. Für die einzelnen Arbeitsschritte wird dann die Ausführungsreihenfolge bestimmt. Die möglichen Bearbeitungswege werden durch Pfeile zwischen den Arbeitsschritten dargestellt. Für die Wege können Bedingungen angegeben werden. Nur wenn diese Bedingungen erfüllt sind wird der Bearbeitungsweg gewählt. Dadurch es ist möglich, je nach Zustand des Vorganges einen anderen Bearbeitungsweg einzuschlagen. Weiterhin ist auch eine Parallelbearbeitung von Vorgängen möglich.

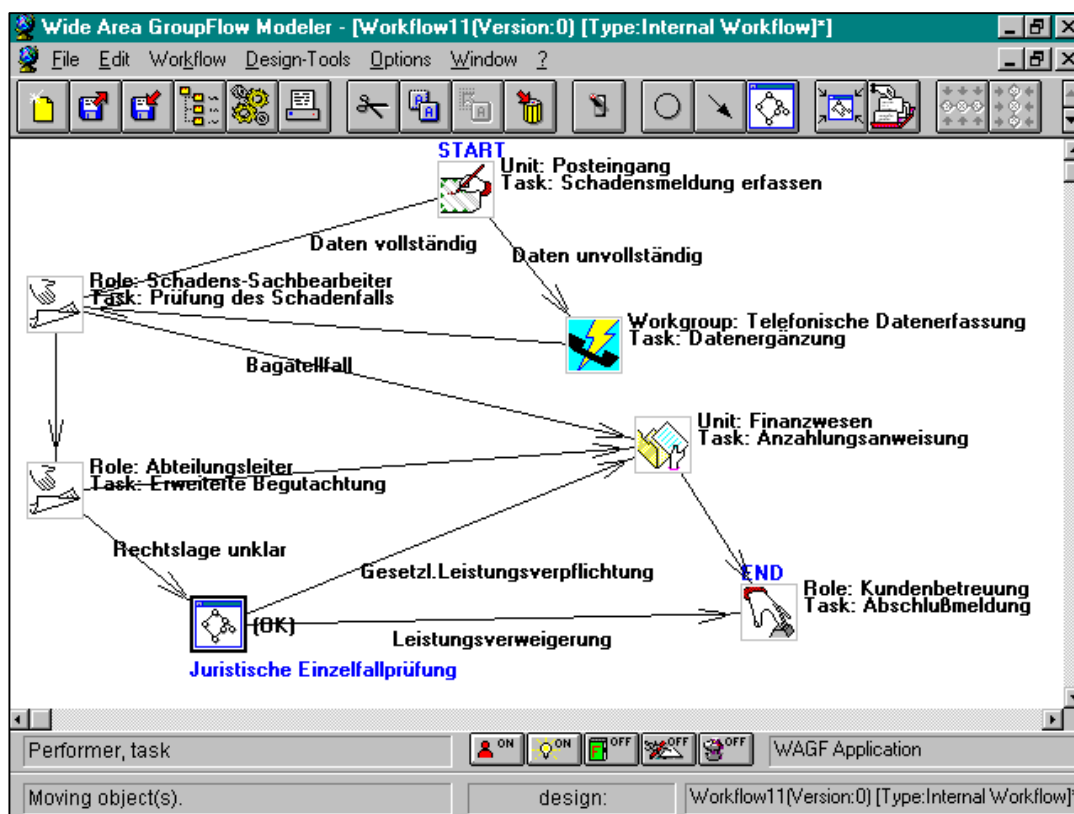


Abbildung 11: Modellierung eines Workflows

Das obige Beispiel zeigt einen Workflowtyp, bei dem mehrere Arbeitsschritte zwei Nachfolger haben. In Abhängigkeit des Ergebnisses der Bearbeitung eines

Arbeitsschrittes wird einer der beiden Nachfolger ausgeführt. Nach dem Erfassen der Schadensmeldung wird, wenn alle Daten vollständig sind, die Prüfung des Schadensfalls durchgeführt. Sind die Daten unvollständig, dann wird vorher noch die Aufgabe Datenergänzung bearbeitet.

Grundlage der Analyse sind die tatsächlich abgelaufenen Vorgangsinstanzen. Deren Graph kann sich von dem bei der Modellierung erzeugten unterscheiden. So wird in einer Instanz des obigen Workflows nur einer der möglichen Bearbeitungswege benutzt und es kann zu ad-hoc Workflows kommen. Die einzelnen Aufgaben werden meist von Personen bearbeitet. Ausnahme ist hier die Bearbeitung durch einen Software-Agent. Die folgende Grafik zeigt einige mögliche Instanzen des Workflows.

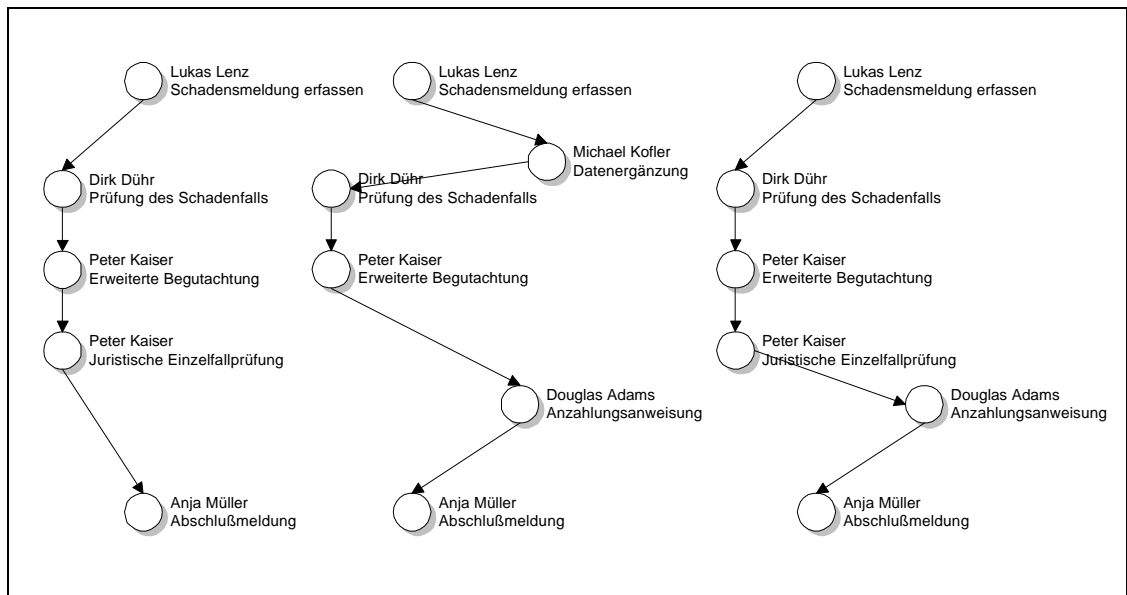


Abbildung 12: Instanzen der Schadensfallmeldung

Für die Analyse wird dieser tatsächliche Ablauf vom Analyser rekonstruiert. Dazu müssen während der Ausführung eines Vorganges die relevanten Daten aufgezeichnet (getrackt) werden. Welche Daten als relevant anzusehen sind hängt von der geforderten Aussagekraft, Genauigkeit und Art der Analyse ab. Um die Graphen des obigen Beispiels darstellen zu können, werden für jeden Arbeitsschritt der Name des Bearbeiters, die bearbeitete Aufgabe sowie die Nachfolger bzw. Vorgänger benötigt. Diese Informationen werden dann einer Instanz eines Workflowtyps zugeordnet. Da die Art der Auswertung auch die benötigten Daten bestimmt, kann jetzt noch nicht genauer

darauf eingegangen werden, welche Daten wie abgespeichert werden müssen. Dies wird in den folgenden Kapiteln nachgeholt.

4.1 Vorgänge, Aufgaben und Bearbeiter als Gegenstand der Analyse

Bei der Ausführung eines Vorganges werden die einzelnen Aufgaben von Bearbeitern ausgeführt. Damit der Analyzer die Aufgaben unterscheiden kann ist es nötig, ihnen eindeutige Bezeichnungen zu geben. Dies kann eine beliebige Zeichenkette sein. Es ist möglich und auch sinnvoll die Bezeichnung zu verwenden, die auch innerhalb des Modelers benutzt wurde. Dadurch kann die Aufgabe leicht durch den Benutzer identifiziert werden. Es kann aber auch eine beliebige andere Bezeichnung verwendet werden. Auf eine Bezeichnung der Aufgaben kann auch ganz verzichtet werden. Dann können aber nicht alle Möglichkeiten des Analyzers genutzt werden.

Bearbeiter einer Aufgabe kann, wie schon erwähnt, eine Person oder ein Software-Agent sein. Wenn man die Namen der Personen in geeigneter Weise abspeichert, kann eine Leistungsbewertung der Bearbeiter durchgeführt werden. Dies ist nicht immer erwünscht. Bei verteilter Vorgangsbearbeitung werden die einzelnen Aufgaben eines Vorganges von verschiedenen Organisationen bearbeitet. Diese werden nur im Ausnahmefall bereit sein den oder die Bearbeiter einer Aufgabe bekanntzugeben. Aus diesen Gründen wird der Begriff des Bearbeiters für den Analyzer erweitert. Ein Bearbeiter kann eine Person, eine Rolle, eine Gruppe, eine Abteilung, eine Organisation oder eine Systemkomponente sein.

Systemkomponenten sind dabei Teile des Workflowsystems, die zur Bearbeitung eines Vorganges beitragen. Insbesondere sind dies das Sending- und Receiving-Gateway. Aber auch Software-Agenten zählen zu den Systemkomponenten.

Bearbeiternamen können hierarchisch oder „flach“ sein. Bei der Notation von hierarchischen Namen lehne ich mich an die Namenskonvention des X.500 Standards¹⁸ an. Gültige Namen setzen sich aus Attributen zusammen, die einen Typ und einen Wert haben. Einzelne Attribute werden in der Form *Typ = Wert* geschrieben und durch das Zeichen „/“ getrennt.

¹⁸ (auch ISO-9594) Ein Dienst, der ein weltweites System zur eindeutigen Benennung, Beschreibung und Auffindung von Netzwerkressourcen zur Verfügung stellt. Vgl. [Kuri, 1996]

Typ	Beschreibung	Erklärung	Beispiel
CN	Common Name	Name einer Person, Rolle oder Gruppe	CN=Olaf Hein CN=Sachbearbeiter
SY	System	Name einer Systemkomponente oder eines Software-Agenten	SY=Sending Gateway
OU	Organizational Unit	Organisationseinheit oder Abteilung	OU=Vertrieb OU=Entwicklung
O	Organization	Name der Organisation	O=SNI
C	Country	Land	C=DE

Tabelle 1: Namenskonvention

Bis auf den Typ SY entspricht dies auch der Notes Namenskonvention. Die Typen CN und SY stehen auf derselben Ebene der Hierarchie. Es kann nur einer der beiden verwendet werden. Auf der nächsten Ebene folgt OU. Dieser darf bis zu viermal verwendet werden. Danach folgen die Typen O und C, die jeweils einmal verwendet werden dürfen. CN=Olaf Hein/OU=Einkauf/O=Abakus/C=DE oder SY=Sending Gateway/O=Peacock/C=DE sind zum Beispiel gültige Namen. Es ist auch erlaubt Attribute eines Typs nicht aufzuführen. Insbesondere muß, im Gegensatz zu Notes Benutzernamen, kein Attribut vom Typ CN oder SY existieren. Es sei noch einmal ausdrücklich darauf hingewiesen, daß sich die Namen für Bearbeiter an die Notes Namenskonvention anlehnen, es sich hierbei aber nicht um Notes Benutzernamen handeln muß.

Durch diese Namenskonvention ergeben sich einige Vorteile. Die Typen CN und SY ermöglichen die Unterscheidung zwischen Personen und Systemkomponenten als Bearbeiter einer Aufgabe. Bei der Analyse können so gezielt die einzelnen Typen betrachtet werden. Was als Wert der Typen eingetragen wird, kann durch das Workflowsystem flexibel bestimmt werden, da es keine Vorgaben gibt. Frei wählbar ist, ob die Namen der einzelnen Personen oder der Name einer Rolle bzw. Gruppe als Common Name gespeichert wird. Es ist auch möglich, den Namen der Person als Common Name und den Namen der Rolle bzw. Gruppe zu der er bei der Bearbeitung der Aufgabe gehört als Organizational Unit abzuspeichern. Dadurch wird eine rollen- oder

gruppenbezogene Auswertung ermöglicht und der Vergleich mit der Prozeßdefinition vereinfacht. Die Hierarchie erlaubt es die Daten zu verdichten. Es können die einzelnen Typen, anhand derer die Bearbeiter unterschieden werden sollen, gezielt ausgewählt werden. Beispielsweise können Bearbeiter nur nach der Organisation und dem Land unterschieden werden.

Aus den Bearbeitern CN=Olaf Hein/OU=Einkauf/O=Abakus/C=DE und CN=Jan Stahl/OU=Einkauf/O=Abakus/C=DE wird OU=Einkauf/O=Abakus/C=DE wenn der Common Name ignoriert wird und O=Abakus/C=DE wenn nur Organisation und Country berücksichtigt werden. Diese Möglichkeit der Datenverdichtung läßt sich besonders bei verteilter Vorgangsbearbeitung sinnvoll einsetzen. Es können dann die einzelnen Vorgänge auf Organisationsebene betrachtet werden. Da kein Attributtyp zwingend erforderlich ist, kann schon beim Tracking flexibel bestimmt werden, welche Informationen zur Analyse zur Verfügung stehen sollen. Bei extern bearbeiteten Aufgaben kann zum Beispiel nur die Organisation und das Land angegeben werden.



Abbildung 13: Bearbeiternamen filtern

Über einen Dialog können die Teile des Namens, die zur Unterscheidung verwendet werden sollen, einzeln ausgewählt werden.

Flache Namen bestehen aus einer Zeichenkette, die keinen der oben genannten Typen gefolgt von dem Zeichen „=" enthält. Sie werden vom Analyser automatisch in einen hierarchischen Namen, der nur ein Attribut vom Typ CN enthält, konvertiert. Aus dem

Namen „Olaf Hein“ wird dann „CN=Olaf Hein“. Bei ihrer Verwendung stehen allerdings einige Möglichkeiten des Analyzers nicht zur Verfügung.

In den folgenden Kapiteln wird auf die Möglichkeiten, die sich aus den hierarchischen Namen ergeben, noch weiter eingegangen.

4.2 Arten der Visualisierung von Vorgängen

Eine der wichtigsten Aufgaben des Analyzers besteht darin, Informationen über selektierte Workflowinstanzen einzulesen, aufzubereiten und anschließend unter verschiedenen Gesichtspunkten zu visualisieren.

In dem folgenden, stark vereinfachten, Beispiel sind zwei Instanzen des Vorgangstyps Bestellung dargestellt. Es gibt in dem Beispiel keine bedingten Verzweigungen. Dafür ist aber eine Parallelbearbeitung von Aufgaben enthalten. Zu Beginn wird ein Beschaffungsantrag ausgefüllt. Dieser geht dann an den Einkauf. Dort werden Angebote von beliebig vielen Anbietern eingeholt. Hier verläßt der Vorgang die Organisation und wird von den Lieferanten weiter bearbeitet. Nachdem die Angebote eingetroffen sind wird der Artikel bei einem der Lieferanten bestellt. Dieser wickelt dann die Bestellung ab und „liefert“ Ware und Rechnung. Rechnungs- und Wareneingang werden dann unabhängig voneinander bearbeitet.

Die Instanz CD-ROM bestellen läuft so ab, wie sie auch definiert wurde. Bei der Instanz PC bestellen wird die Aufgabe Antrag stellen jedoch zweimal und die Aufgabe Angebot anfordern dreimal bearbeitet. Dieser Fall tritt zum Beispiel ein, wenn ein Bearbeiter feststellt, daß der Vorgänger falsche oder ungenügende Angaben gemacht hat und der Vorgang dem Vorgänger wieder vorgelegt wird. Dieser nimmt dann die nötigen Korrekturen vor.

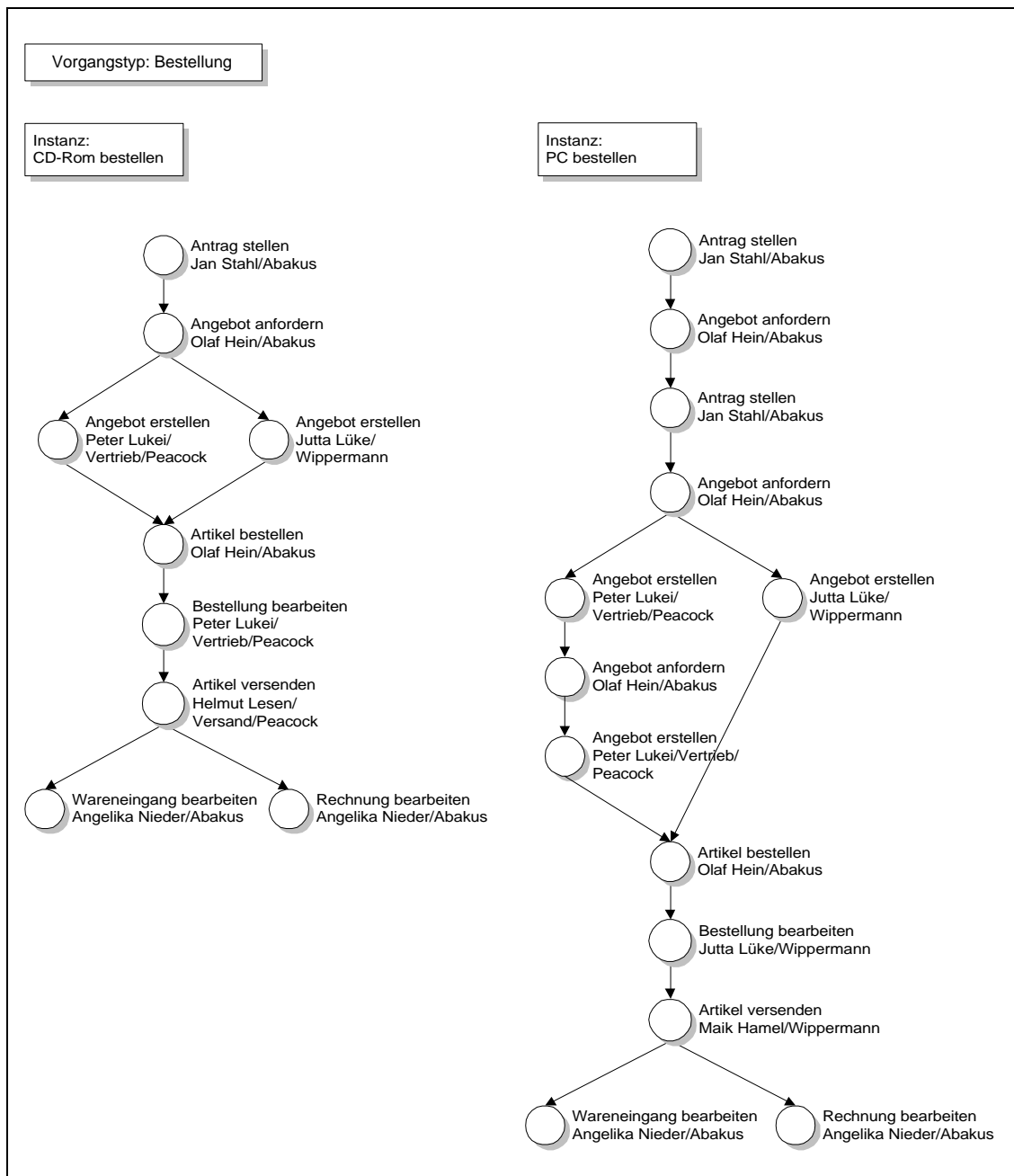


Abbildung 14: Beispielvorgänge einer Bestellung

Aus den Informationen der einzelnen Vorgänge können verschiedene Ansichten generiert werden. Für jede dieser Ansichten wird ein gerichteter Graph erzeugt und angezeigt. Die einzelnen Ansichten unterscheiden sich dabei durch die Art der Knoten im Graphen. Es gibt drei Arten von Knoten.

1. Bearbeiter

Für jeden Bearbeiter, der innerhalb eines Vorganges eine Aufgabe bearbeitet hat, wird ein Knoten erzeugt. Durch diese bearbeiterbezogene Sicht auf die Vorgänge können

Kommunikationsbeziehungen zwischen den Bearbeitern betrachtet werden. Bei Verwendung hierarchischer Namen lassen sich so die Kommunikationsbeziehungen zwischen Abteilungen oder Organisationen betrachten und auswerten.

2. Aufgabe

Jeder Knoten repräsentiert eine Aufgabe. Dadurch kann der Ablauf von Vorgängen auf Aufgabenebene verfolgt werden. Wiederkehrende Folgen von Aufgaben können so erkannt und analysiert werden.

3. Arbeitsschritt

Ein Knoten wird durch den Bearbeiter und die Aufgabe identifiziert. Dadurch ist es möglich den Ablauf einzelner Vorgänge zu visualisieren.

Für jede Art von Knoten hat der Analyzer drei Möglichkeiten einen Graph zu erzeugen.

4.2.1 Kommunikationsbeziehungen zwischen Bearbeitern, Aufgaben oder Arbeitsschritten

In dieser Ansicht wird jeder Knoten genau einmal dargestellt. Für jeden Knoten wird beim Aufbau des Graphen ermittelt, wie oft er in den einzelnen Vorgängen vorkommt. Eine Kante wird zwischen zwei Knoten gezeichnet, wenn diese beiden Knoten in mindestens einem Vorgang in einer Vorgänger Nachfolgerbeziehung zueinander stehen. Dabei wird für jede Kante die Anzahl dieser Beziehungen ermittelt und auf Wunsch angezeigt. Für die Beispielvorgänge ergeben sich dann die folgenden Graphen. In den Beispielen wurden die Knoten so angeordnet, daß der Ablauf der Vorgänge wiederzuerkennen ist. Bei der automatischen Anordnung innerhalb des Analyzers ist dies nicht möglich. Da ein Knoten innerhalb verschiedener Vorgänge vorkommen kann, diese Vorgänge im allgemeinen aber nicht gleich aufgebaut sind, ist es nicht möglich eine sinnvolle Struktur für den Graphen zu wählen. Zusätzlich können zwischen allen Knoten Kanten existieren. Es muß also sichergestellt werden, daß eine Kante nicht durch andere Kanten verdeckt wird. Sie darf auch nicht durch Knoten hindurch gezeichnet werden. Dadurch würde der Eindruck entstehen, dieser Knoten ist ein Endpunkt der Kante. Um diese Probleme zu vermeiden werden die Knoten standardmäßig kreisförmig angeordnet. Der Benutzer kann sie dann per Drag & Drop beliebig verschieben oder in Matrixform anordnen lassen.

Kommunikationsbeziehungen zwischen Arbeitsschritten

Diese Ansicht bietet eine Übersicht über alle Arbeitsschritte, die in den betrachteten Vorgängen enthalten sind.

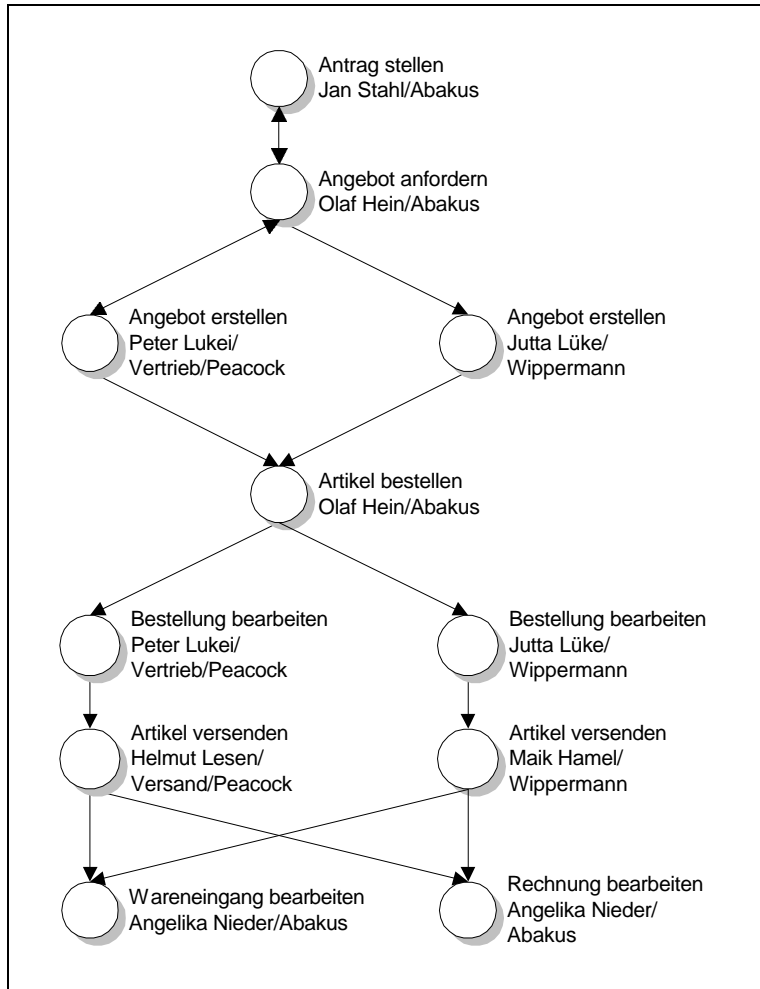


Abbildung 15: Kommunikationsbeziehungen zwischen Arbeitsschritten

Für jeden Knoten kann angezeigt werden, wie oft er bearbeitet wurde. Weiterhin kann für jeden Knoten die Liste der Vorgänge angezeigt werden, in denen er vorkommt. Durch die Auswahl eines Vorganges oder auch mehrerer Vorgänge werden die Kanten zwischen den Knoten, die in diesen Vorgängen nacheinander bearbeitet wurden, farbig hervorgehoben. Ist ein Knoten innerhalb eines Vorganges sein eigener Nachfolger, dann wird er farbig umrahmt. Dadurch erhält man einen ersten Überblick über den Ablauf eines Vorganges und kann erkennen, aus welchen Arbeitsschritten er besteht.

Wenn hierarchische Namen verwendet werden können die Informationen weiter verdichtet werden. Dann ist es zum Beispiel möglich diesen Graph auf

Organisationsebene anzuzeigen. Es ist dann leicht zu erkennen, welche Aufgaben von welcher Organisation bearbeitet wurden.

Kommunikationsbeziehungen zwischen Bearbeitern

Mit Hilfe dieser Ansicht kann man einen Überblick über die Bearbeiter und deren Kommunikationsbeziehungen erhalten.

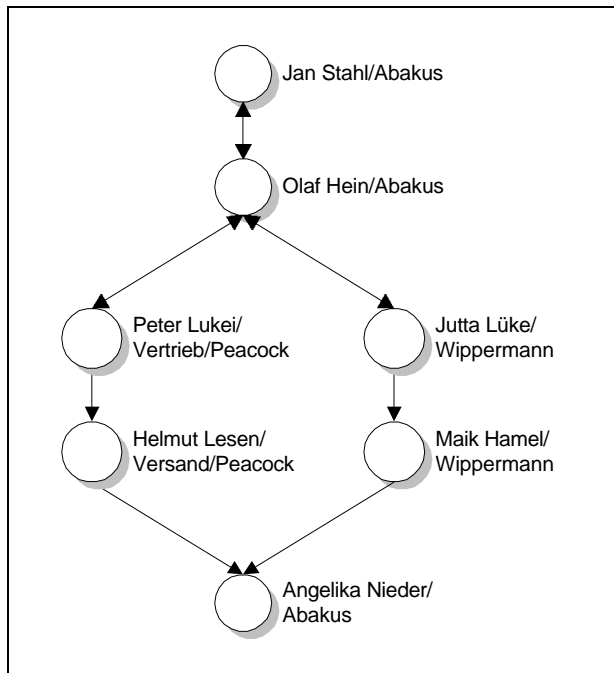


Abbildung 16: Kommunikationsbeziehungen zwischen Bearbeitern

Für jeden Bearbeiter kann eine Liste der bearbeiteten Vorgänge und eine Liste der bearbeiteten Aufgaben angezeigt werden. Dabei wird für jede Aufgabe angezeigt, wie oft sie von dem entsprechenden Bearbeiter bearbeitet wurde. Durch Auswahl eines Vorganges oder einer Aufgabe wird der gesamte Vorgang farblich hervorgehoben. Dadurch ist es möglich für einen Bearbeiter festzustellen, an welchen Vorgängen er beteiligt war und welche Aufgaben er bearbeitet hat. Diese Informationen können genutzt werden, um festzustellen, wie stark die einzelnen Bearbeiter ausgelastet sind oder mit welchen Aufgaben ein Bearbeiter bereits vertraut ist. Dadurch ist Kontrolle und Planung des Ressourceneinsatzes möglich.

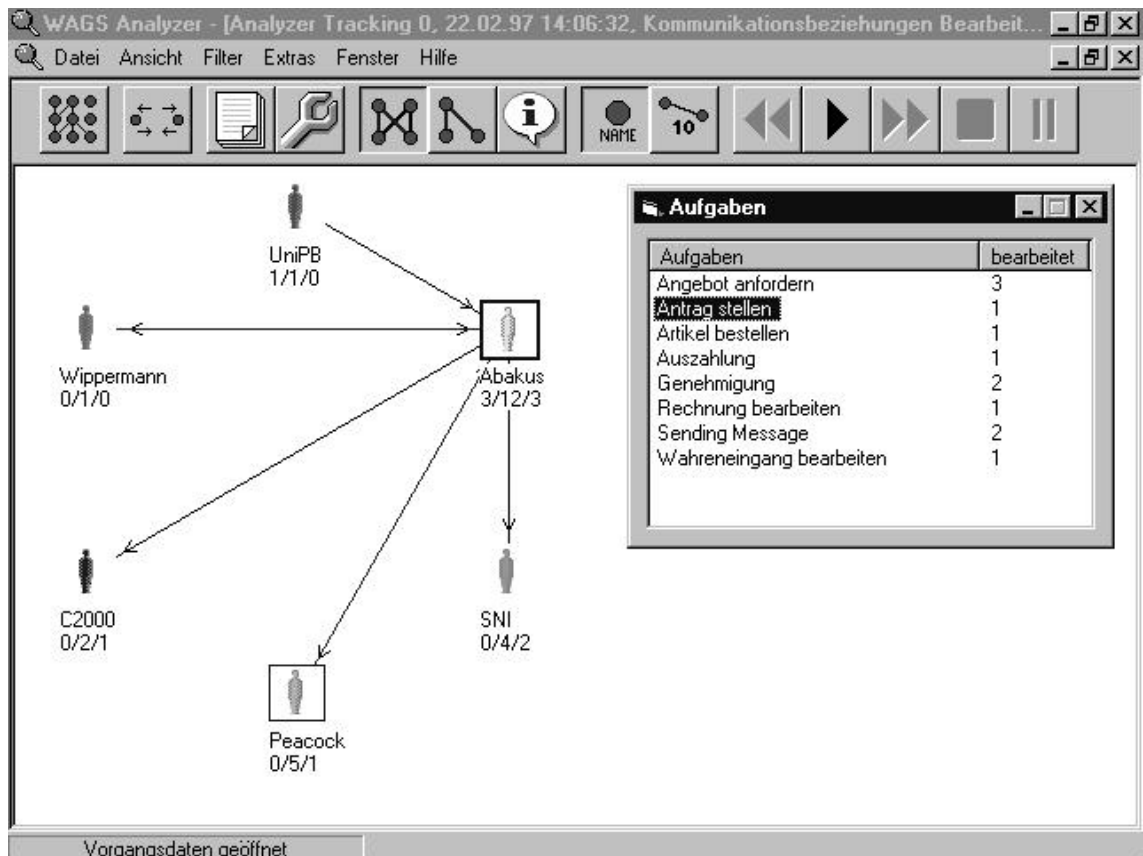


Abbildung 17: Bearbeitete Aufgaben einer Organisation

Auf Organisationsebene können die Kommunikationsbeziehungen mit den am Workflow beteiligten Organisationen betrachtet werden. Auf einen Blick kann man sehen, wie viele Vorgänge mit den einzelnen Partnern bearbeitet wurden. Im Beispiel werden die bearbeiteten Aufgaben der Organisation Abakus aufgelistet.

Kommunikationsbeziehungen zwischen Aufgaben

Diese aufgabenbezogene Ansicht ist besonders für die Modellierung von Prozessen nützlich. Sie stellt alle Aufgaben übersichtlich dar. Wenn vorhandene Aufgaben in neuen Prozessen wiederverwendet werden, dann besteht in dieser Ansicht die Möglichkeit, sich die bisherigen Bearbeiter dieser Aufgabe anzeigen zu lassen. Aus diesen bereits mit der Aufgabe vertrauten Bearbeitern kann man dann einen oder mehrere als Bearbeiter innerhalb des neuen Prozesses auswählen.

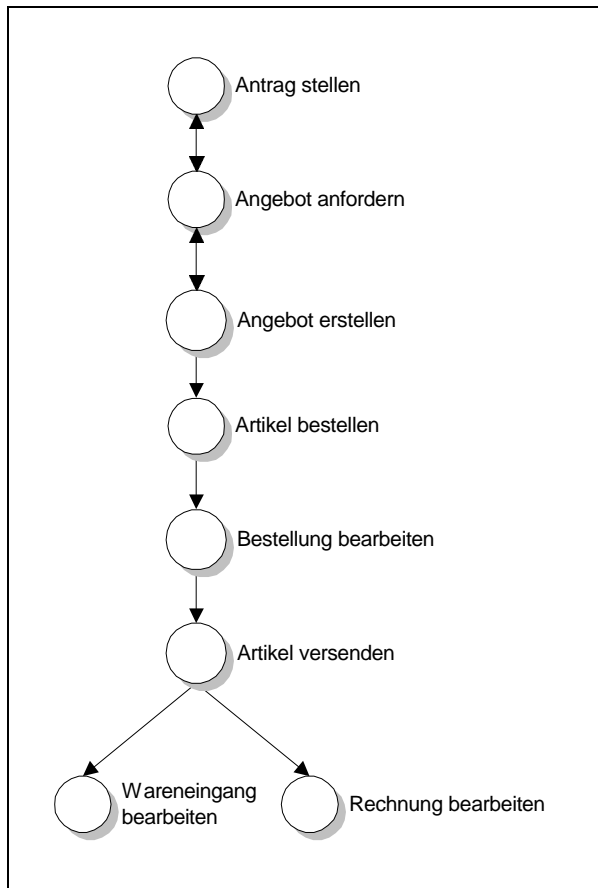


Abbildung 18: Kommunikationsbeziehungen zwischen Aufgaben

In dieser Ansicht können auch Aufgabenfolgen, die häufig vorkommen, erkannt werden. Mann erhält weiterhin einen Überblick darüber, welche Aufgaben wie oft bearbeitet wurden. Die am häufigsten bearbeiteten Aufgaben können dann genauer untersucht und eventuell optimiert werden.

4.2.2 Ein Bearbeiter, eine Aufgabe oder ein Arbeitsschritt im Mittelpunkt der Betrachtung

Diese Ansichten stellen einen zuvor selektierten Knoten und dessen Kommunikationsbeziehungen in den Mittelpunkt der Betrachtung. Für den Aufbau des Graphen werden nur die Vorgänge berücksichtigt, in denen dieser Knoten enthalten ist. Alle direkten Nachfolger und Vorgänger dieses Knotens werden zum Graphen hinzugefügt. Kanten werden von den Vorgängern zu dem selektierten Knoten und von diesem zu den Nachfolgern eingefügt. Weiterhin wird festgehalten, wie oft ein Knoten Nachfolger oder Vorgänger war. Bei der Anzeige des Graphen werden alle Nachfolger und Vorgänger kreisförmig um den selektierten Knoten angeordnet. Da nur die

unmittelbaren Nachbarn des selektierten Knotens angezeigt werden, sind diese Ansichten gut geeignet, um einen Knoten und dessen Kommunikationsbeziehungen genauer zu betrachten. Unnötige Knoten und Kanten, die den Graphen unübersichtlich machen würden, werden nicht angezeigt.

Ein Arbeitsschritt im Mittelpunkt

Es wird ein ausgewählter Arbeitsschritt in den Mittelpunkt gestellt. Im Beispiel ist dies die Aufgabe „Artikel bestellen“ bearbeitet von Olaf Hein/Abakus

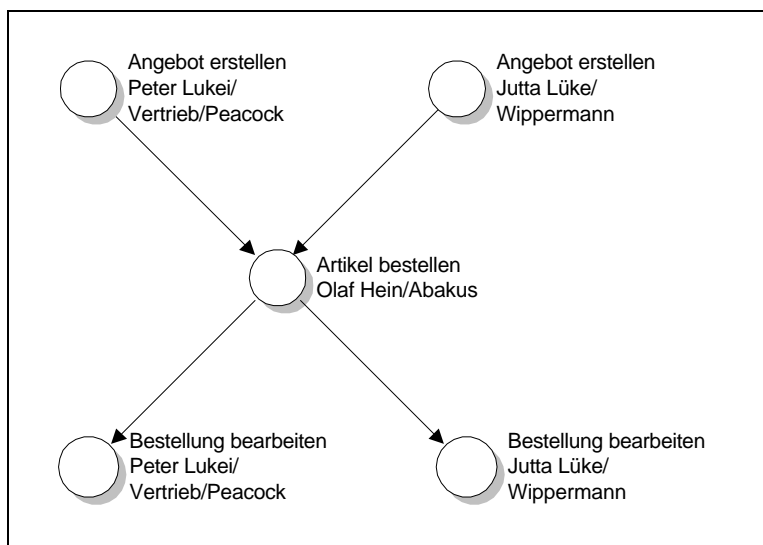


Abbildung 19: Ein Arbeitsschritt im Mittelpunkt

Für diesen Arbeitsschritt ist es möglich, die Liste der bearbeiteten Vorgänge anzuzeigen. Durch Auswahl eines Vorganges werden die Kanten zu den Vorgängern und Nachfolgern innerhalb dieses Vorganges farbig hervorgehoben.

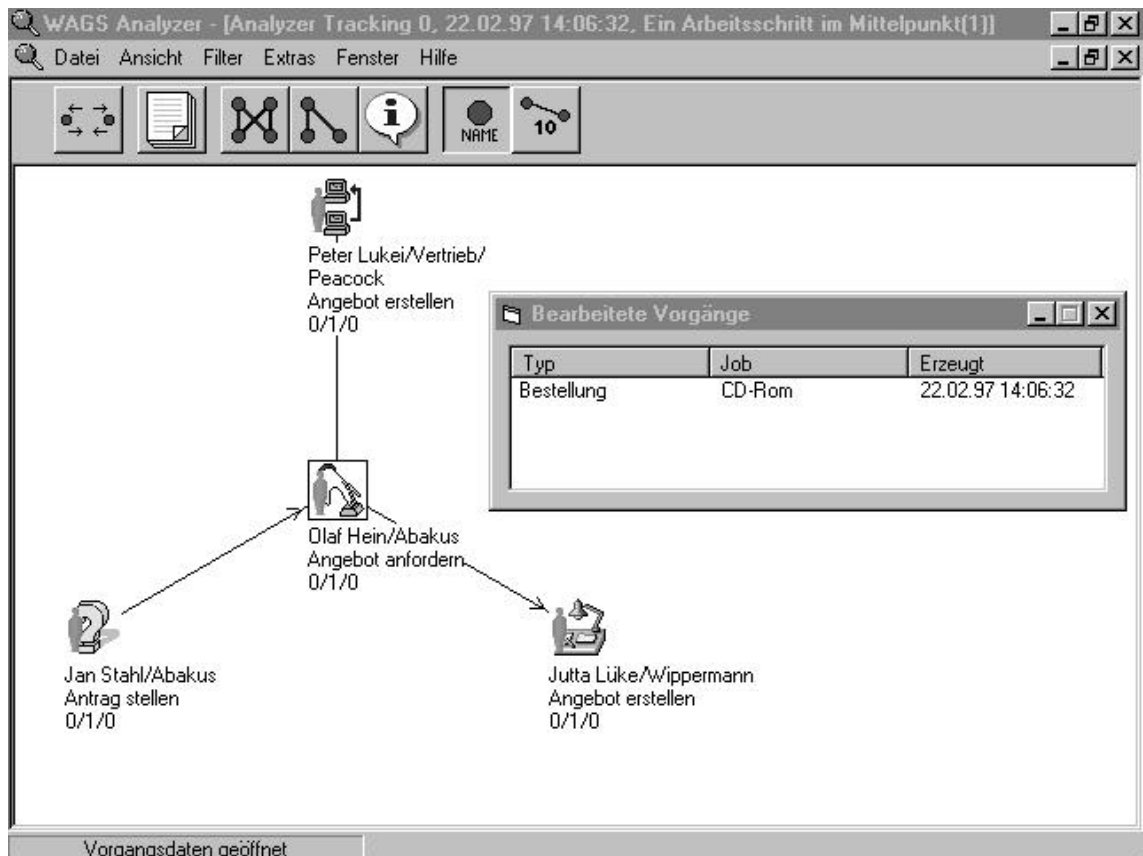


Abbildung 20: Ein Arbeitsschritt im Mittelpunkt der Betrachtung

In dem obigen Beispiel steht die Aufgabe „Angebot anfordern“ mit dem Bearbeiter Olaf Hein/Abakus im Mittelpunkt der Betrachtung. Die Aufgaben der Vorgänger und Nachfolger werden nach Bearbeitern getrennt dargestellt.

Eine Aufgabe im Mittelpunkt

Eine vorher selektierte Aufgabe kann in dieser Ansicht genauer betrachtet werden. Für jede Aufgabe kann neben den bearbeiteten Vorgängen auch eine Liste der Bearbeiter angezeigt werden.

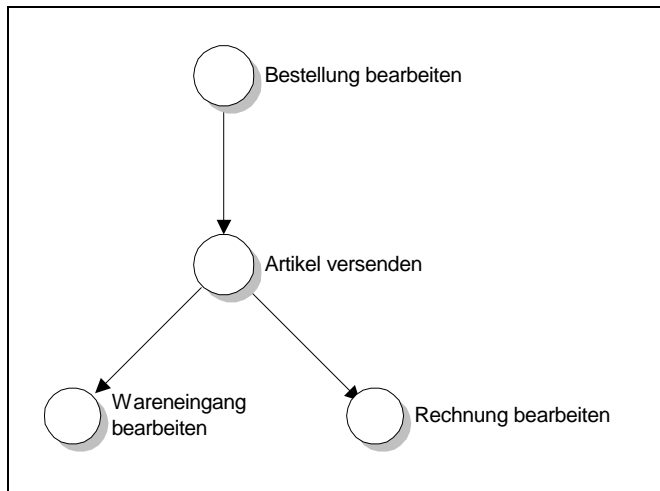


Abbildung 21: Eine Aufgabe im Mittelpunkt

Ein Bearbeiter im Mittelpunkt

In dieser Ansicht wird ein Bearbeiter in den Mittelpunkt gestellt. Neben der Liste der Vorgänge kann für jeden Bearbeiter auch eine Liste der bearbeiteten Aufgaben angezeigt werden.

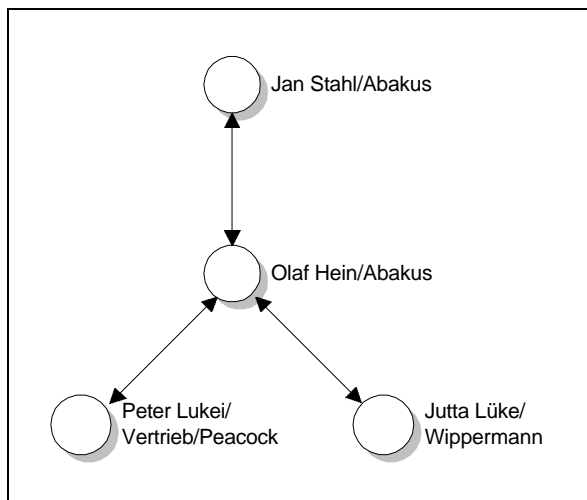


Abbildung 22: Ein Bearbeiter im Mittelpunkt

4.2.3 Bearbeitungsschritte von Vorgängen

Bei den folgenden Ansichten werden die Knoten zusätzlich durch ihre Position in der Bearbeitungsfolge unterschieden. Diese Position wird im folgenden Bearbeitungsschritt genannt. Der Bearbeitungsschritt wird beim Einlesen der Vorgangsdaten für jeden Arbeitsschritt berechnet. Er ist eine natürliche Zahl und gibt die Reihenfolge der Bearbeitung an. Er wird wie folgt definiert.

Es sei A die Menge aller Arbeitsschritte eines Vorganges.

Sei $a \in A$ ein Arbeitsschritt und $B(a) \in \mathbb{N}$ der Bearbeitungsschritt von a .

$B(a) = 1$, falls a keine Vorgänger hat

$B(a) = \max(\{B(b) \mid b \in A \text{ und } b \text{ ist Vorgänger von } a\}) + 1$, sonst

Auf die Ermittlung des Bearbeitungsschrittes wird in einem späteren Kapitel noch genauer eingegangen.

Wenn zum Beispiel die Aufgabe „Antrag stellen“ in einem Vorgang als erste Aufgabe ausgeführt wird und in einem anderen als zweite, dann sind dies im Graphen zwei unterschiedliche Knoten, da im ersten Fall der Bearbeitungsschritt 1 und im zweiten 2 ist. Ein Knoten existiert auf jeder Bearbeitungsstufe aber höchstens einmal. Dadurch wird es möglich, den genauen Ablauf der betrachteten Vorgänge darzustellen. Eine Kante zwischen zwei Knoten wird dann eingefügt, wenn innerhalb eines Vorganges zwischen diesen beiden Knoten eine Vorgänger-Nachfolger Beziehung existiert. Bei der Darstellung werden Knoten mit dem gleichen Bearbeitungsschritt in eine Zeile gezeichnet. Die oberste Zeile enthält die Knoten mit Bearbeitungsschritt 1. In der zweiten folgen die Knoten mit Bearbeitungsschritt 2, u.s.w. Der Bearbeitungsweg eines Vorganges ist dabei immer von oben nach unten. Es gibt keine Kanten von unten nach oben.

Bearbeitungsschritte Arbeitsschritt

Auch hier gibt es wieder drei verschiedene Graphen. Bei der Darstellung der Arbeitsschritte entsteht für die Beispielvorgänge der folgende Graph.

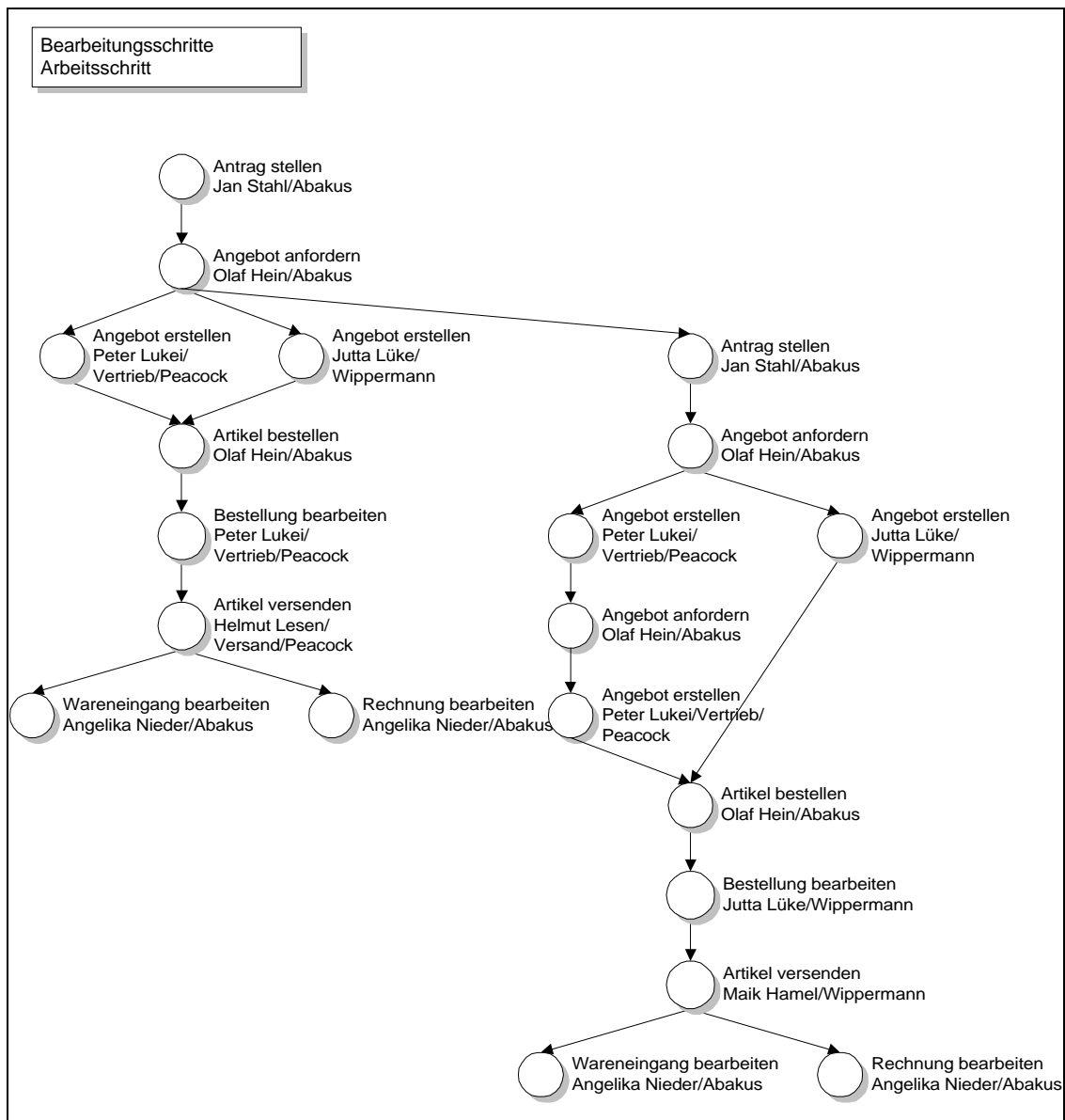


Abbildung 23: Bearbeitungsschritte Arbeitsschritt

Diese Darstellung visualisiert den Ablauf von Vorgängen. Der Bearbeitungsweg der Vorgänge kann genau untersucht werden. Dies wird durch die Möglichkeit, einzelne Vorgänge farblich hervorzuheben und unerwünschte auszublenden, noch unterstützt. Durch einen Vergleich mit dem modellierten Prozeß können Abweichungen erkannt werden. Es gilt dann die Ursachen dafür zu finden und zu beseitigen.

Bearbeitungsschritte Aufgabe

Durch die Darstellung der Aufgaben wird von den Bearbeitern abstrahiert. Auch dieser Graph kann verwendet werden, um Abweichungen vom Soll zu erkennen. Er ist aber auch geeignet, um natürlich entstandene Bearbeitungsfolgen von Aufgaben zu erkennen.

Diese können dann genauer untersucht werden, um aus den gewonnenen Informationen einen Prozeß zu modellieren.

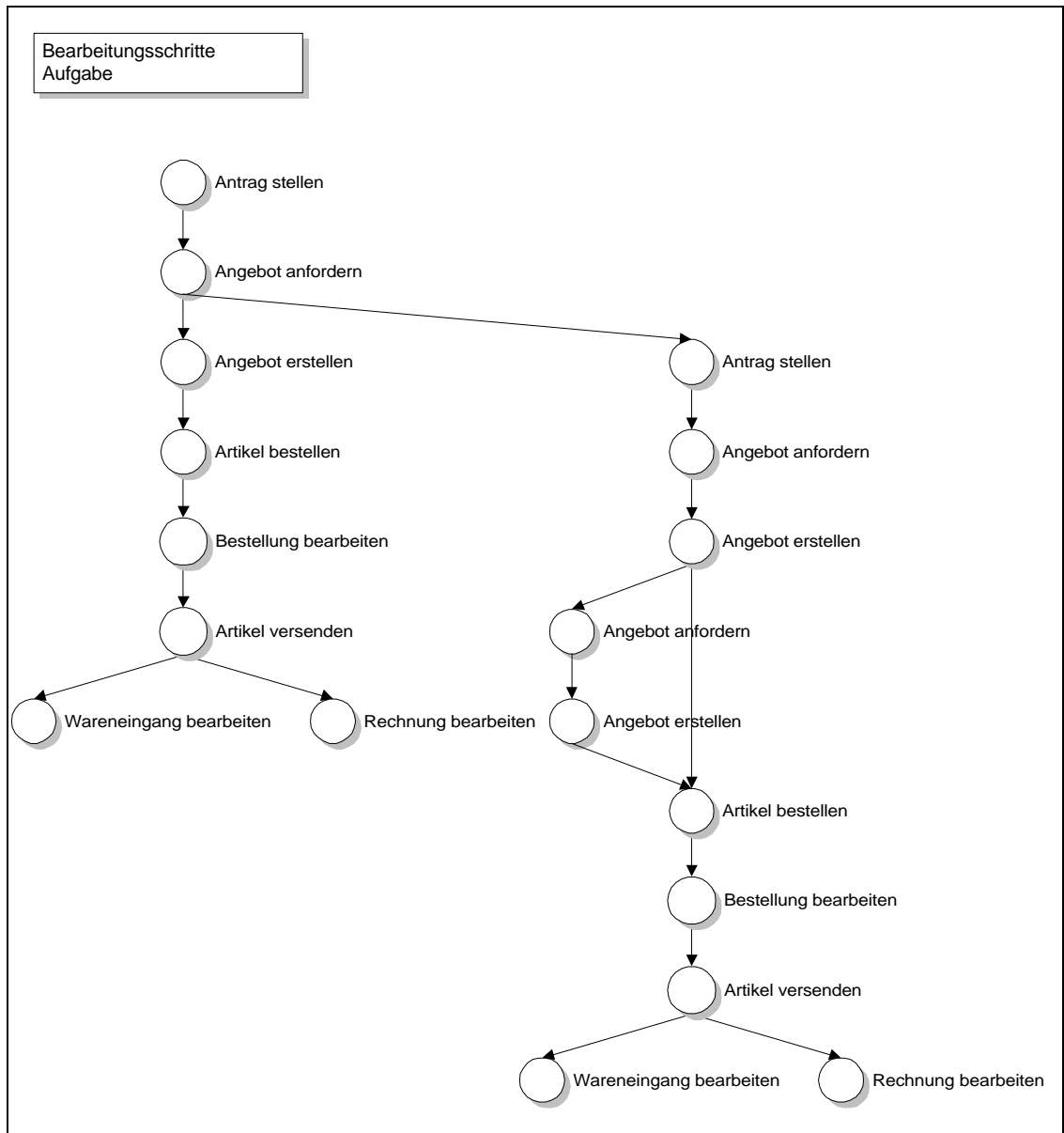


Abbildung 24: Bearbeitungsschritte Aufgaben

Bearbeitungsschritte Bearbeiter

Der letzte Graph stellt nur die Bearbeiter dar. Er dient dazu, die Bearbeitung auf Bearbeiterebene zu verfolgen. Dadurch wird die Funktion eines Bearbeiters innerhalb der Vorgangsbearbeitung deutlich. Eine Funktion kann der Initiator von Vorgängen, der letzte Bearbeiter oder ein beliebiger anderer Bearbeiter sein. Eine besondere Bedeutung fällt dieser Ansicht bei der Analyse von herkömmlichen Vorgangsdatenbanken zu. Darunter sind Lotus Notes Datenbanken zu verstehen, in denen die Bearbeitungshistorie

von Dokumenten in dem \$UpdatedBy Feld abgespeichert wird. In solchen Datenbanken stehen im allgemeinen keine Informationen über die bearbeiteten Aufgaben zur Verfügung. Mit dieser Ansicht besteht dennoch die Möglichkeit, natürlich entstandene Workflows zu erkennen.

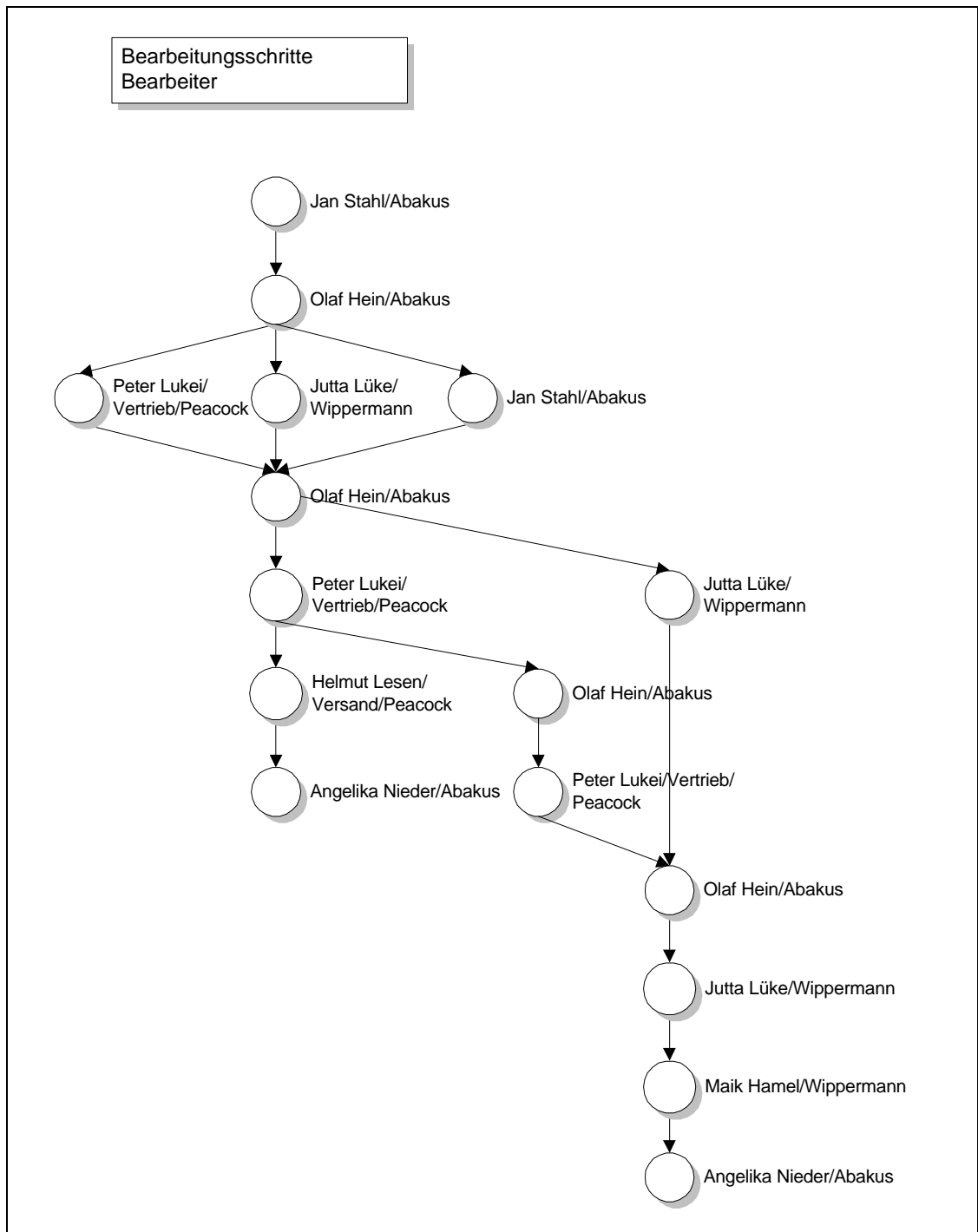


Abbildung 25: Bearbeitungsschritte Bearbeiter

So erkannte Workflows können dann genauer analysiert, optimiert und modelliert werden.

4.3 Selektion zu analysierender Vorgänge

Um eine gezielte Analyse von Vorgängen vornehmen zu können, ist es nötig, die Eingabedaten zu selektieren und zu filtern. Dafür stellt der Analyzer vielfältige Möglichkeiten zur Verfügung.

4.3.1 Einlesen von Daten aus einer Datenbank

Die zu analysierende Datenbank ist frei wählbar. Es werden keine besonderen Anforderungen an die Datenbank gestellt. Weiterhin ist die View¹⁹, aus der die Daten gelesen werden, einstellbar. Aus der gewählten View werden dann alle Notes Dokumente eingelesen. Jedes Dokument enthält dabei die Trackinginformationen eines Vorganges. Der Aufbau dieser Dokumente wird in einem späteren Kapitel genauer beschrieben. Durch das Erstellen einer View in der Notes Datenbank kann der Benutzer die zu lesenden Dokumente selektieren. Hierdurch können die Vorgänge zum Beispiel nach Status, Typ oder Erstellungsdatum ausgewählt werden. Status kann dabei zum Beispiel Entwurf, in Arbeit oder abgeschlossen sein. Eine zeitpunktbezogene Auswertung kann dadurch realisiert werden, daß in einer View alle Dokumente angezeigt werden, die noch in Arbeit sind. Für eine zeitraumbezogene Auswertung werden alle Dokumente, deren Erstellungsdatum innerhalb eines vorgegebenen Intervalls liegt, in einer View aufgelistet.

Neben der Auswahl einer View müssen auch die Namen der Felder angegeben werden, aus denen der Analyzer die Informationen für den Aufbau der Vorgangsgraphen lesen soll.

¹⁹ Hier ist die Ansicht in der Notes Datenbank gemeint und nicht die im WAGS Analyzer



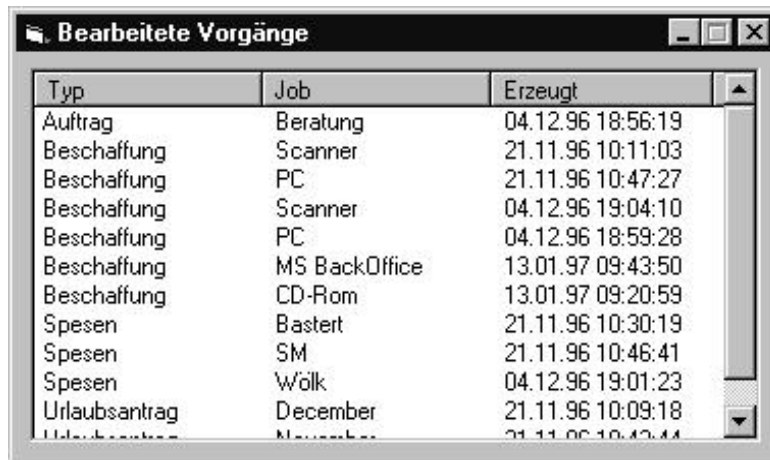
Abbildung 26: Auswahl der Feldnamen

Wichtig sind hier vor allem die Felder für Bearbeiter und Aufgabe. Eines dieser Felder muß angegeben werden, daß zweite kann. Alle anderen Felder sind optional. „Typ“, „Job“ und „Erzeugt“ dienen der Identifikation der Vorgänge gegenüber dem Anwender. Der Analyzer selbst braucht diese Felder nicht, um die einzelnen Vorgänge zu unterscheiden. „Geändert“ wird bei der Berechnung eines Differenzworkflows verwendet, „Vorgänger“ und „Nummer“, um die parallele Bearbeitung von Aufgaben korrekt abbilden zu können.

4.3.2 Gezielte Auswahl einzelner Vorgänge

Nach dem Einlesen werden aus den Daten die Vorgänge rekonstruiert. Die Graphen für die einzelnen Ansichten werden dann aus diesen Daten erzeugt. Um eine gezielte Analyse von Vorgängen, Bearbeitern oder Aufgaben zu ermöglichen, ist es möglich, die Vorgänge zu selektieren, die als Datenbestand für die Ansichten verwendet werden.

Dazu besteht die Möglichkeit, die Vorgänge direkt auszuwählen. Dies geschieht durch Selektion der Vorgänge in einer Listbox.



Typ	Job	Erzeugt
Auftrag	Beratung	04.12.96 18:56:19
Beschaffung	Scanner	21.11.96 10:11:03
Beschaffung	PC	21.11.96 10:47:27
Beschaffung	Scanner	04.12.96 19:04:10
Beschaffung	PC	04.12.96 18:59:28
Beschaffung	MS BackOffice	13.01.97 09:43:50
Beschaffung	CD-Rom	13.01.97 09:20:59
Spesen	Bastert	21.11.96 10:30:19
Spesen	SM	21.11.96 10:46:41
Spesen	Wölk	04.12.96 19:01:23
Urlaubsantrag	December	21.11.96 10:09:18
Urlaubsantrag	November	21.11.96 10:12:44

Abbildung 27: Selektion von Vorgängen

Die Vorgänge können auch indirekt ausgewählt werden. Dazu werden Bearbeiter oder Aufgaben ausgewählt. Alle Vorgänge, die von einem der gewählten Bearbeiter oder einer der gewählten Aufgaben bearbeitet wurden, werden dann in den Ansichten dargestellt.

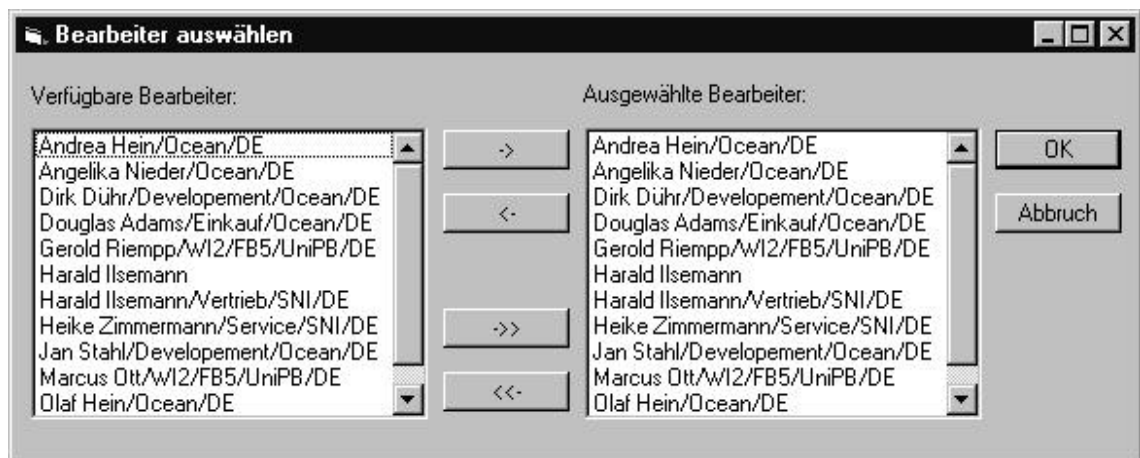


Abbildung 28: Bearbeiter auswählen

Dadurch können gezielt Vorgänge untersucht werden, die von einem Bearbeiter bearbeitet wurden oder die eine bestimmte Aufgabe enthalten. In den einzelnen Ansichten werden aber in der Regel auch andere als die gewählten Bearbeiter oder Aufgaben dargestellt. Dies ist durchaus gewollt, da durch diese Selektion indirekt die Vorgänge ausgewählt werden. Diese werden aber immer als ganzes dargestellt.

4.3.3 Zeitraumbezogene Analyse durch Differenzworkflows

Der Analyzer hat die Möglichkeit, die eingelesenen Vorgangsdaten abzuspeichern. Dabei werden allerdings nur die reinen Vorgangsdaten gespeichert und nicht die einzelnen geöffneten Ansichten. Die Vorgänge können zu einem späteren Zeitpunkt wieder geladen und analysiert werden.

Die Daten können auch mit den Vorgangsdaten eines anderen Zeitpunktes verglichen werden. Dazu werden zwei Differenzworkflows D_1 und D_2 wie folgt gebildet.

Es wird eine Menge von Vorgängen V_1 zum Zeitpunkt t_1 gelesen. Zu einem Zeitpunkt t_2 wird eine Menge von Vorgängen V_2 gelesen. Ein sinnvoller Vergleich ist dabei nur möglich wenn $V_1 \neq V_2$ und $V_1 \cap V_2 \neq \emptyset$.

Es sei v ein beliebiger Vorgang. $C(v, V)$ sei der Zeitpunkt der letzten Änderung von v in der Vorgangsmenge V .

$$D_1 := \{v \in V_1 \mid v \notin V_2 \vee C(v, V_2) < C(v, V_1)\}$$

$$D_2 := \{v \in V_2 \mid v \notin V_1 \vee C(v, V_2) > C(v, V_1)\}$$

Statt V_1 bzw. V_2 können dann auch die Vorgänge in den Differenzworkflows D_1 und D_2 für die Berechnung der Ansichten verwendet werden. Hierdurch wird es möglich, den Fortschritt der Vorgangsbearbeitung innerhalb des Workflowsystems zu analysieren.

4.4 Statistische Auswertung von Vorgangsdaten

Neben der Visualisierung der Vorgänge wird auch eine statistische Auswertung ermöglicht. Hierzu werden die visualisierten Daten in Tabellen und Diagrammen dargestellt.

4.4.1 Informationen über die Anzahl der bearbeiteten Vorgänge

Beim Erzeugen der Graphen für die einzelnen Ansichten werden für jeden Knoten verschiedene Informationen ermittelt. Dies ist die Anzahl der Vorgänge, in denen der Knoten enthalten ist. Kommt ein Knoten innerhalb eines Vorganges zweimal vor, dann wird dieser Vorgang auch zweimal gezählt. Weiterhin wird die Anzahl der Vorgänge, in denen der Knoten keine Nachfolger hat und die Anzahl der Vorgänge, in denen er keine Vorgänger hat, festgehalten. Im ersten Fall ist dieser Knoten der erste Knoten innerhalb der Bearbeitung, und im zweiten Fall ist er der letzte Knoten des Vorganges. In

Ansichten ist es möglich, diese Informationen für einzelnen Knoten anzeigen zu lassen. Sie sind hilfreich, um zu ermitteln, welche Bearbeiter Vorgänge erzeugt oder abschließend bearbeitet haben. Weiterhin stehen sie für die statistische Auswertung zur Verfügung.

Die Kanten repräsentieren die Weiterleitung eines Vorganges von einem Knoten zu seinen Nachfolgern. Für die Kanten wird die Anzahl der Vorgänge festgehalten, die zwischen den Endknoten weitergeleitet wurden.

Diese mengenmäßigen Angaben können zum Beispiel verwendet werden, um festzustellen, ob Mitarbeiter überlastet sind oder noch freie Kapazitäten haben.

4.4.2 Bearbeitete Aufgaben der Bearbeiter

Diese Auswertung steht nur zur Verfügung, wenn für die einzelnen Arbeitsschritte die Aufgaben und die Bearbeiter eingelesen wurden. Je Bearbeiter wird für alle Aufgaben angezeigt, wie oft er diese bearbeitet hat.

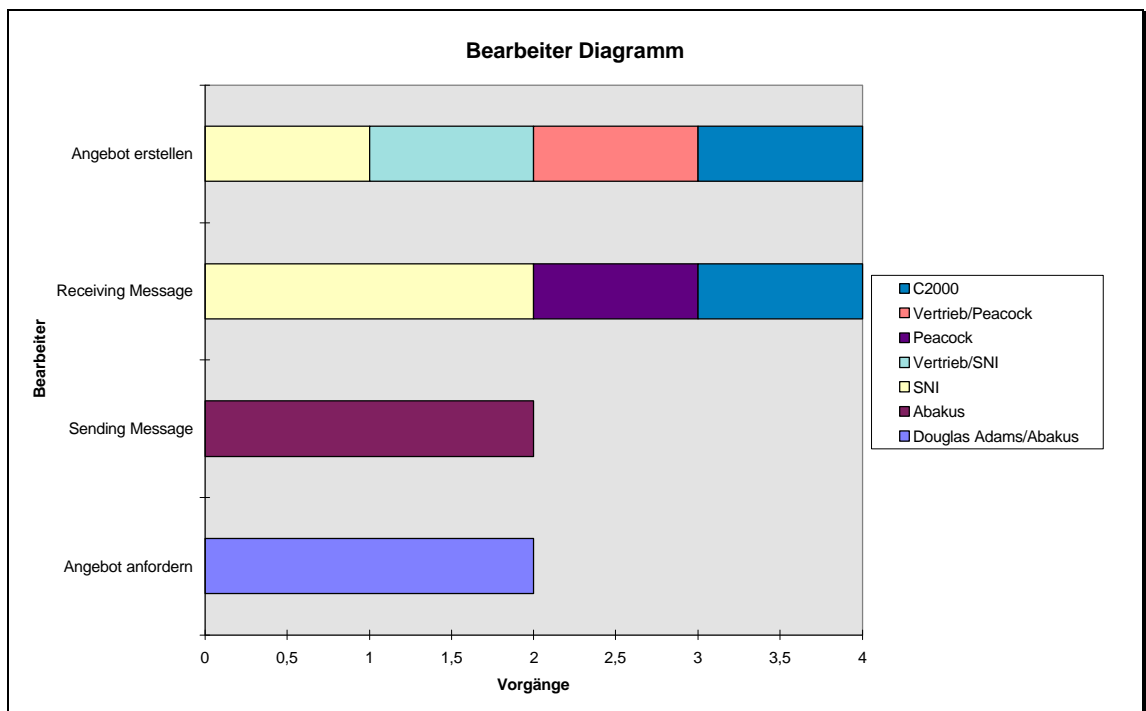


Abbildung 29: Bearbeiter Diagramm

Anhand des Diagramms kann sehr schnell ermittelt werden, wie viele Vorgänge ein Bearbeiter insgesamt bearbeitet hat. Weiterhin kann man einen Überblick über die einzelnen bearbeiteten Aufgaben erhalten. Solch ein Diagramm existiert auch für

Aufgaben, so daß die Anzahl der Vorgänge, in denen eine bestimmte Aufgabe bearbeitet wurde und die Bearbeiter, die diese Aufgabe bearbeitet haben, abgelesen werden können. Die Diagramme können zum Beispiel bei der Modellierung neuer Workflows behilflich sein. So können mit ihnen schnell Fragen wie: Wer hat mit einer gegebenen Aufgabe schon Erfahrung? Oder: Welche Aufgaben hat ein bestimmter Bearbeiter schon bearbeitet? beantwortet werden.

4.4.3 Analyse eines Bearbeiters oder einer Aufgabe

Um einen Bearbeiter oder eine Aufgabe genauer zu untersuchen, gibt es die Bearbeiter und die Aufgaben Statistik. Es können für einen Bearbeiter alle bearbeiteten Aufgaben oder für eine Aufgabe alle Bearbeiter betrachtet werden.

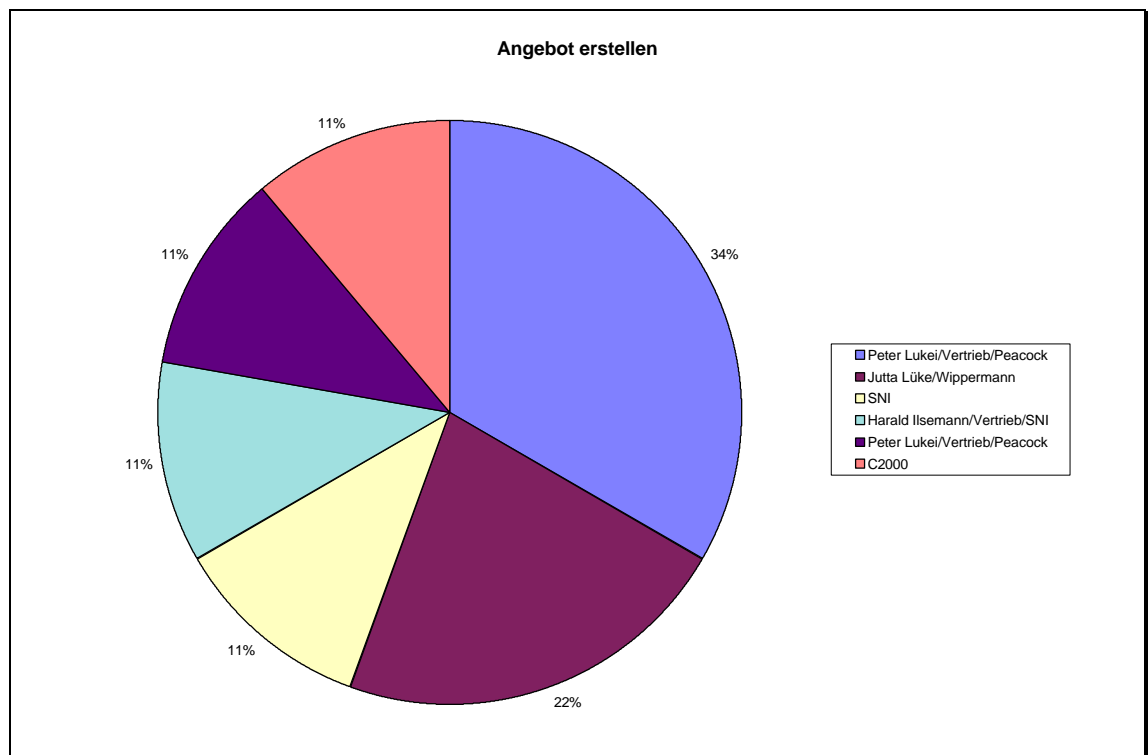


Abbildung 30: Aufgaben Diagramm

4.4.4 Kommunikationsbeziehungen zwischen Bearbeitern oder Aufgaben

Anhand dieser Tabellen und Diagramme können die Kommunikationsbeziehungen zwischen einzelnen Knoten untersucht werden. Ein Knoten kann hierbei ein Bearbeiter, eine Aufgabe oder ein Arbeitsschritt sein. In der Tabelle werden in Zeilen die Namen der Vorgänger innerhalb der Vorgangsbearbeitung eingetragen. In den Spalten stehen die

Namen der Nachfolger. In die so entstandene Matrix wird die Anzahl der Vorgänge eingetragen, die von einem Vorgänger zu einem bestimmten Nachfolger weitergegeben wurde.

4.5 Anforderungen an Eingabedaten

Um die Vorgangsgraphen aufbauen zu können, müssen folgende Informationen zur Verfügung stehen.

- Jeder Knoten muß eindeutig identifizierbar sein
- Bearbeiter und Aufgabe für die einzelnen Arbeitsschritte
- Jeder Knoten muß seine Nachfolger oder seine Vorgänger kennen

Diese Informationen müssen bei der Vorgangsbearbeitung in geeigneter Form aufgezeichnet werden. Diese Aufzeichnung wird auch Tracking genannt. Dabei gibt es einige Probleme, die gelöst werden müssen. Beim WAGS handelt es sich um ein System zur verteilten Vorgangsbearbeitung. Bei der Ausführung einzelner Vorgänge auf einem Rechner muß dieser nicht mit einem zentralen Server verbunden sein. Dies gilt insbesondere für Vorgänge, die von mehreren Organisationen bearbeitet werden. Um aber eine Analyse aller Vorgänge durchführen zu können, sollten die Vorgangsdaten zentral verfügbar sein. Dies kann durch eine zentrale Notes Datenbank erreicht werden. In dieser Datenbank, die auch Trackingdatenbank genannt wird, werden dann die Ausführungsinformationen gespeichert. Die Trackingdatenbank sollte je eine Schnittstelle zum Workflowsystem und zum Analyser besitzen. Durch die Schnittstellenfunktion der Datenbank braucht bei Änderungen am Workflowsystem nicht zwangsläufig der Analyser angepaßt zu werden. Es wird in den meisten Fällen ausreichen, die Schnittstelle zum Workflowsystem zu ändern. Dies ist besonders wichtig, da sich das WAGS noch in der Entwicklung befindet. Die Änderung der Datenbank dürfte in den meisten Fällen wesentlich einfacher sein, als die Änderung des Analyzers. Dies liegt daran, daß Veränderungen am Design von Datenbanken von vorhandenen Datenbanken, durch einen Updatemechanismus, übernommen werden können. Weiterhin brauchen Datenbanken nicht kompiliert werden. Bei Änderungen am Analyser muß dagegen der Quelltext verändert und eine neue ausführbare Datei erzeugt werden. Der vorhandene Analyser muß dann durch den neuen ersetzt werden.

Da die genaue Gestaltung der Trackingdatenbank nicht Thema dieser Arbeit ist, konzentriere ich mich auf die Beschreibung der für den Analyzer relevanten Eigenschaften.

In der Datenbank wird für jeden Vorgang ein Dokument angelegt. In dieses Dokument werden nach der Beendigung einer Aufgabe die Trackinginformationen eingefügt. Falls bei der Ausführung einer Vorgangsinstantz auf die Trackingdatenbank zugegriffen werden kann, kann dies sofort geschehen. Ist dies nicht möglich, werden die Daten gesammelt und bei der nächsten Möglichkeit in die Datenbank geschrieben. Dies kann zum Beispiel mit Hilfe von E-Mail geschehen. Dadurch sind die Trackinginformationen aber nicht immer vollständig. Es können Informationen auch verloren gehen. Weiterhin ist eine Rückmeldung der Trackingdatenbank an die Vorgangsinstantz nicht immer möglich. Im Folgenden sollen die Anforderungen, die der Analyzer an die Trackingdatenbank stellt, genauer beschrieben werden. Dabei werden die genannten Probleme berücksichtigt.

Da die Informationen visualisiert werden sollen, ist es nötig, daß die Informationen, die einen Knoten identifizieren, für einen Menschen aussagekräftig sind. Über ID's (z.B. Nummerieren der Knoten) ist es zwar möglich einen Knoten zu identifizieren. Ein Benutzer, der sich den Graphen dann anschaut, kann damit im allgemeinen aber nur wenig anfangen. Dies kann durch ein Mapping von ID's auf aussagekräftige Namen vermieden werden. Ein anderes Problem, daß bei der Verwendung der Knoten ID's auftritt ist, daß für einen Arbeitsschritt eine Gruppe als Bearbeiter definiert werden kann. Wenn solch ein Arbeitsschritt in einem Vorgang zweimal vorkommt, aber von unterschiedlichen Bearbeitern aus der Gruppe bearbeitet wird, sind dies für das Tracking und den Analyzer zwei unterschiedliche Knoten. Sie müssen also auch unterschiedliche ID's haben. Dies kann erreicht werden, indem nicht die ID, sondern der tatsächliche Bearbeiter und die Aufgabe zur Identifizierung verwendet werden. Das ist aber noch nicht ganz eindeutig. Es kann vorkommen, daß ein Arbeitsschritt in einem Vorgang an den vorherigen Bearbeiter zur Korrektur zurückgeschickt wird. In diesem Fall existieren zwei Knoten, die beide denselben Bearbeiter und dieselbe Aufgabe haben. Es wird also noch ein zusätzliches Kriterium zur Unterscheidung benötigt. Um das Problem zu lösen, kann in jeder Vorgangsinstantz ein Zähler für die einzelnen Arbeitsschritte geführt werden. Dieser wird immer dann erhöht, wenn der Arbeitsschritt bearbeitet wird. Diese

Methode hat noch einen kleinen Nachteil. Wenn Arbeitsschritte mit gleichem Bearbeiter und gleicher Aufgabe parallel bearbeitet werden, dann sind die Zähler dieser beiden Arbeitsschritte voneinander unabhängig. Sie können die gleiche Nummer erhalten. Dadurch können sie vom Analyzer nicht mehr unterschieden werden.

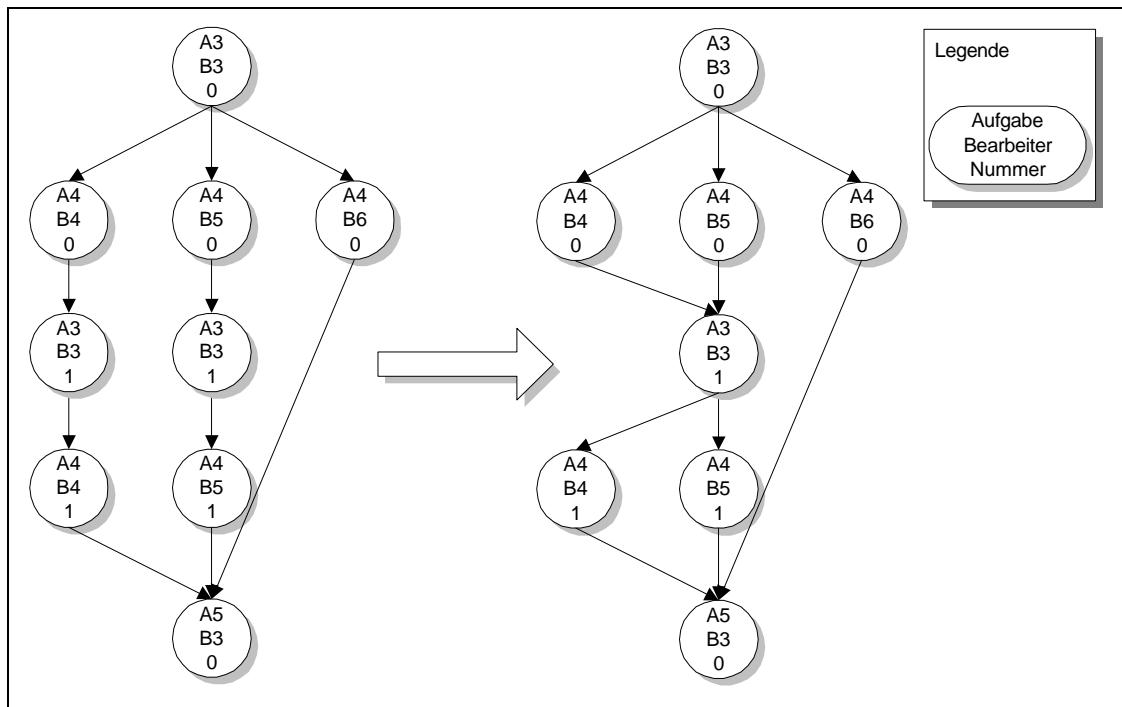


Abbildung 31: Falsche Rekonstruktion eines Vorganges

Im obigen Beispiel ist dieser Fall dargestellt. Im linken Graph wird im zweiten Bearbeitungsschritt von zwei Bearbeitern der Vorgang an den vorherigen Bearbeiter zurückgeschickt. In beiden Fällen wird die fortlaufende Nummer des Arbeitsschrittes (A3, B3) von 0 auf 1 erhöht. Aus diesen Informationen entsteht der Graph auf der rechten Seite. Dieses Problem kann durch eine zentrale Verwaltung der Folge Nummern, zum Beispiel in der Trackingdatenbank, gelöst werden. Dies ist aber wegen der verteilten Ausführung der Vorgänge nicht immer möglich. Da die Möglichkeit der verteilten Vorgangsbearbeitung aber wichtiger ist, als eine exakte Analyse wird dieser Fehler in Kauf genommen.

Zusätzlich zu den Informationen einzelner Knoten ist es noch sinnvoll, Daten über den Vorgang abzuspeichern. Hierzu zählen neben dem Vorgangstyp und der Vorgangsinanz auch der Zeitpunkt, zu dem der Vorgang erstellt bzw. das letzte Mal bearbeitet wurde.

Der Typ, die Instanz und das Erstellungsdatum können vom Analyzieren angezeigt werden.
Anhand dieser Daten kann ein Anwender die Vorgänge unterscheiden.

5 Implementierung des Wide Area GroupFlow Analyzer

5.1 Verwendete Werkzeuge

5.1.1 Visual Basic 5.0 als Programmiersprache

Als Sprache für die Implementierung wurde Visual Basic gewählt. Von Visual Basic werden, wie in anderen Hochsprachen auch, Funktionen und Prozeduren sowie die üblichen Schleifenkonstrukte unterstützt. Dadurch ist eine strukturierte Programmierung möglich. Weiterhin können auch Klassen verwendet werden. Die neueste Version, Visual Basic 5.0, unterstützt auch Polymorphismus in Form von abstrakten Basisklassen. Vererbung wird aber noch nicht unterstützt. Daher ist eine objektbasierte aber keine objektorientierte Programmierung möglich.

Die Entwicklungsumgebung ist sehr komfortabel. Sie enthält einen graphischen Editor, mit dem die Oberfläche des Programms entwickelt werden kann. Ein integrierter Debugger hilft bei der Fehlersuche und die umfangreichen Hilfsfunktionen bei der Fehlervermeidung.

Ein weiterer Punkt, der für Visual Basic spricht, ist die Möglichkeit, ein Programm starten zu können, ohne daß es vorher kompiliert werden muß. Dadurch wird die Zeit für die Fehlersuche und somit auch die Entwicklungszeit verkürzt.

Die größten Nachteile sind das Laufzeitverhalten und der Speicherplatzverbrauch des fertigen Programms. In der Version 5.0 verfügt Visual Basic zwar über einen Compiler, die erreichbare Geschwindigkeit ist aber mit C++ Code nicht zu vergleichen. Am deutlichsten ist dies bei Grafikoperationen zu spüren. Ein weiteres Problem ist die fehlende Vererbung von Klassen. Dadurch kann nicht objektorientiert programmiert werden. Als Folge muß Code oft doppelt geschrieben werden.

5.1.2 MS Excel für die Statistische Auswertung

Die statistische Auswertung wird mit Hilfe einer Tabellenkalkulation realisiert. Hierzu wird Excel 5.0 oder höher verwendet. Dazu wird Excel über OLE vom Analyzer gesteuert. Dadurch wird eine hohe Flexibilität erreicht. Für die einzelnen Diagramme und Tabellen gibt es Schablonen. Diese werden mit den Daten gefüllt und können

anschließend abgespeichert, ausgedruckt und beliebig weiter bearbeitet werden. Weiterhin können die Schablonen vom Anwender an seine Bedürfnisse angepaßt werden. Dieses Vorgehen hat den Vorteil, daß die Anwendungsentwicklung vereinfacht und beschleunigt wird. Das Erstellen von Diagrammen ist mit der Excel Funktionalität sehr einfach. Für den Anwender ergeben sich die Vorteile einer hohen Flexibilität und sehr gute Möglichkeiten der Weiterbearbeitung. So ist es im Rahmen einer Diplomarbeit nicht möglich, die Möglichkeiten, die eine Tabellenkalkulation zur Verfügung stellt, auch nur ansatzweise zu implementieren. Excel als Tabellenkalkulation hat in diesem Fall den Vorteil, daß die Programmiersprache Visual Basic for Applications (VBA), die von Excel verwendet wird, eine Teilmenge von Visual Basic ist. Dadurch kann im gesamten Programm die gleiche Syntax verwendet werden. Ein weiteres Argument für Excel ist dessen sehr hohe Verbreitung. Von Nachteil ist, daß Excel beim Anwender vorhanden sein muß. Ein weiteres Argument gegen die Verwendung einer Tabellenkalkulation ist die mangelnde Kontrolle über die Applikation. Der Anwender kann zum Beispiel die vom Analyzer geöffneten Tabellen schließen, ohne daß der Analyzer dies bemerkt. Der nächste Funktionsaufruf des Analyzers, der eine dieser Tabellen voraussetzt, erzeugt dann eine Fehlermeldung. In diesem Fall ist das aber kein Problem, da die Tabellen erst angezeigt werden, nachdem der Analyzer alle erforderlichen Operationen auf ihnen ausgeführt hat.

5.2 Beschreibung der Schnittstelle zum Wide Area GroupFlow System

In der Tracking Datenbank wird für jeden Vorgang ein Dokument erzeugt. Dieses enthält in einzelnen Feldern die allgemeinen Vorgangsdaten. Immer, wenn die Bearbeitung einer Aufgabe beendet wurde, werden der Bearbeiter, die Aufgabe und die fortlaufende Nummer des Arbeitsschrittes in das entsprechende Vorgangsdokument eingefügt. Zusätzlich werden die Vorgänger dieses Arbeitsschrittes gespeichert.

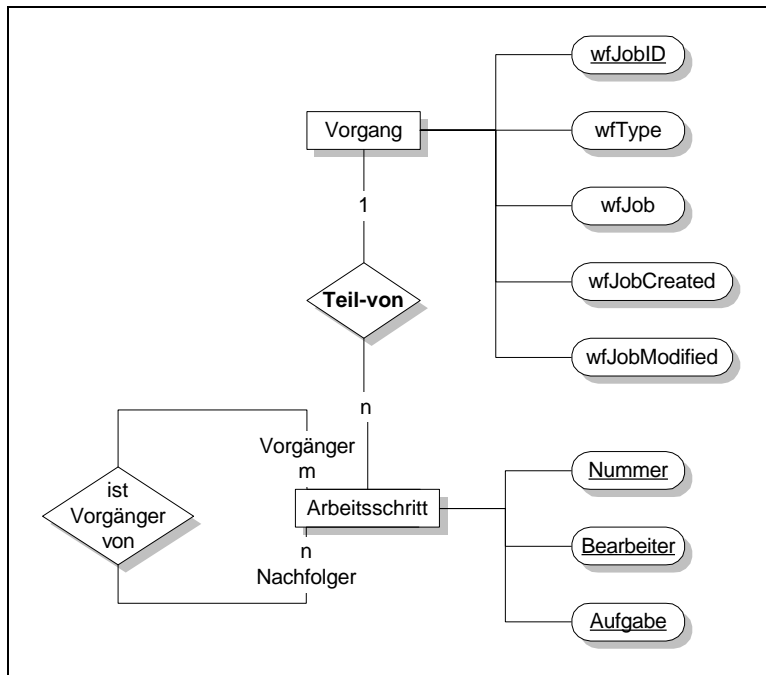


Abbildung 32: Entity-Relationship-Modell eines Vorganges

Die Abbildung beschreibt einen Vorgang, so wie er in der Tracking Datenbank angelegt wird. Ein Vorgang wird eindeutig durch eine ID identifiziert und besteht aus einzelnen Arbeitsschritten. Diese werden durch eine Nummer, den Bearbeiter und die Aufgabe identifiziert. Die Nummer ist, wie weiter oben bereits beschrieben, eine fortlaufende Nummer. Weiterhin können Arbeitsschritte einen Vorgänger und einen Nachfolger besitzen.

Feldname	Typ	Beschreibung	Beispiel
wfJobID	Text	Eindeutige Nummer oder Zeichenfolge, anhand der ein Vorgang innerhalb der Tracking DB vom Workflowsystem identifiziert werden kann.	
WfType	Text	Typ des Vorganges	Bestellung
wfJob	Text	Name der aktuellen Instanz eines Vorgangstyps	Monitor
wfJobCreated	Time	Datum und Zeit an dem der Vorgang angelegt wurde	03.03.97 12:34:23
wfJobModified	Time	Datum und Zeit der letzten Änderung des Vorganges	04.03.97 10:25:28
UpdatedBy	Text	Liste der Bearbeiter des Vorganges	CN=Olaf Hein/O=Abakus; CN=Peter Lukei/O=Peacock
UpdatedTask	Text	Liste der Aufgaben des Vorganges	Angebot anfordern; Angebot erstellen
Number	Text	Liste von ganzen Zahlen	0; 0
Predecessors	Text	Liste der Vorgänger	!!; CN=Olaf Hein/O=Abakus

Tabelle 2: Felder im Trackingdokument

Die Einträge in den Listen werden dabei implizit durch ihre Position in der Liste durchnummeriert. Alle Einträge an einer Listenposition gehören zu einem Arbeitsschritt. Die Felder Predecessors und UpdatedBy werden noch genauer spezifiziert.

UpdatedBy = (Bearbeiter + { ";" + Bearbeiter })
Predecessors = (Vorgängerliste + { ";" + Vorgängerliste })
Vorgängerliste = Vorgänger + { "|" + Vorgänger }
Vorgänger = (Bearbeiter) + "!" + (Aufgabe) + "!" + (Folgenummer)
Folgenummer = * fortlaufende Nummer*
Aufgabe = * Beschreibung der Aufgabe, Aussagekräftige Zeichenfolge *
Bearbeiter = { XName | Name }

XName	= ([CommonName System]) + 0{OrganizationalUnit}4 + (Organization) + (Country) * Benutzername in der X.500 Notation, so wie er auch von Notes verwendet wird. Die einzelnen Einträge sind durch „/“ zu trennen. z.B.: „CN=Olaf Hein/OU=Entwicklung/O=Ocean/C=DE“ „OU=SNI“*
CommonName	= „CN=“ + * Name des Bearbeiters, der Gruppe oder der Rolle *
System	= „SY=“ + * Name der Systemeinheit. Z.B.: „Sending Gateway“
OrganizationalUnit	= „OU=“ + * Bezeichnung der Organisationseinheit, 4 Stufige Hierarchie ist erlaubt *
Organization	= „O=“ + * Name der Organisation *
Country	= „C=“ + * Länderkürzel oder Name , z.B. „DE“, „Deutschland“ *

5.3 Datenstruktur für das Abspeichern eines Vorganges innerhalb des WAGS Analyzers

5.3.1 Allgemeine Informationen über Graphen

Ein gerichteter Graph G ist ein Paar (V, E) . V ist dabei eine endliche Menge und E eine binäre Relation auf V . V ist die Knotenmenge des Graphen G , die Elemente von V werden Knoten genannt. E ist die Kantenmenge von G , die Elemente von E werden Kanten genannt. Die Abbildung zeigt einen Graphen mit der Knotenmenge $\{1, 2, 3, 4\}$ und der Kantenmenge $\{(1, 2), (1, 3), (1, 4), (3, 2), (3, 4)\}$. Die Knoten werden dabei als Kreise dargestellt und die Kanten als Pfeile zwischen den Knoten.

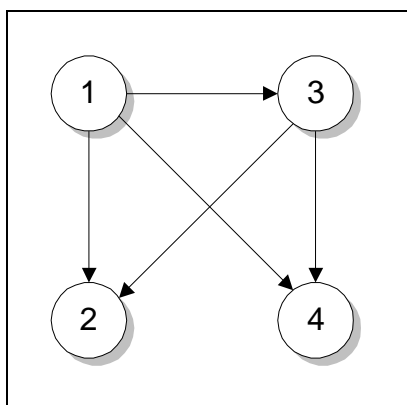


Abbildung 33: Graph

$|E|$ ist die Anzahl der Kanten und $|V|$ die Anzahl der Knoten. Ist $(u, v) \in E$, dann sagt man v ist adjazent zum Knoten u . Der Ingrad eines Knoten v ist die Anzahl der Kanten, die in den Knoten hineingehen, der Outgrad ist die Anzahl der Kanten, die ihn verlassen. Im Beispiel hat der Knoten 3 einen Ingrad von 1 und einen Outgrad von 2.

Ein Weg von u nach v in einem Graphen G ist eine Sequenz von Knoten $[v_0, v_1, v_2, \dots, v_k]$. Hierfür gilt $v_0 = u$, $v_k = v$ und $(v_{i-1}, v_i) \in E$ für $i = 1, 2, \dots, k$. Dabei ist k die Länge des Weges. Ein Weg ist ein einfacher Weg, wenn alle Kanten auf dem Weg verschieden sind.

5.3.2 Adjazenzliste als Repräsentation eines Vorganges

Es gibt zwei Standardmöglichkeiten, einen Graph zu repräsentieren. Als Adjazenzliste oder als Adjazenzmatrix. Die Adjazenzmatrix ist effizient für dichte Graphen, d.h. wenn $|E|$ nicht viel kleiner als $|V|^2$ ist, oder wenn es wichtig ist, sehr schnell zu erfahren, ob es eine Kante $(u, v) \in E$ gibt. Für eine Adjazenzmatrix wird ein zweidimensionales Array $A = (a_{i,j})$ der Größe $|V| \times |V|$ angelegt. Dabei ist

$$a_{i,j} = \begin{cases} 1, & \text{wenn } (i, j) \in E \\ 0, & \text{sonst} \end{cases}$$

Die Anzahl der Kanten bei der Darstellung von Vorgängen ist wesentlich kleiner als $|V|^2$, denn meist hat ein Knoten nur einen oder zwei Nachfolger. Ein schneller Zugriff auf einzelne Kanten wird ebenfalls nicht benötigt. Daher wird nicht weiter auf die Adjazenzmatrix eingegangen. Die Adjazenzliste ist für Graphen, bei denen $|E|$ wesentlich kleiner als $|V|^2$ ist, wesentlich speichereffizienter. Sie besteht aus einem Feld mit $|V|$ Listen. Für jeden Knoten $v \in V$ gibt es eine Liste. Diese Liste enthält die Nummern der adjazenten Knoten von v . Diese Implementierung einer Adjazenzliste benötigt $O(V+E)$ Speicherplatz im Gegensatz zu $\Theta(V^2)$ der Adjazenzmatrix.

Im WAGS Analyzer werden in einigen Funktionen auch die Vorgänger eines Knoten benötigt. Um dies effizient zu implementieren, wird in der Adjazenzliste zusätzlich eine Liste mit den Vorgängern der Knoten abgespeichert.

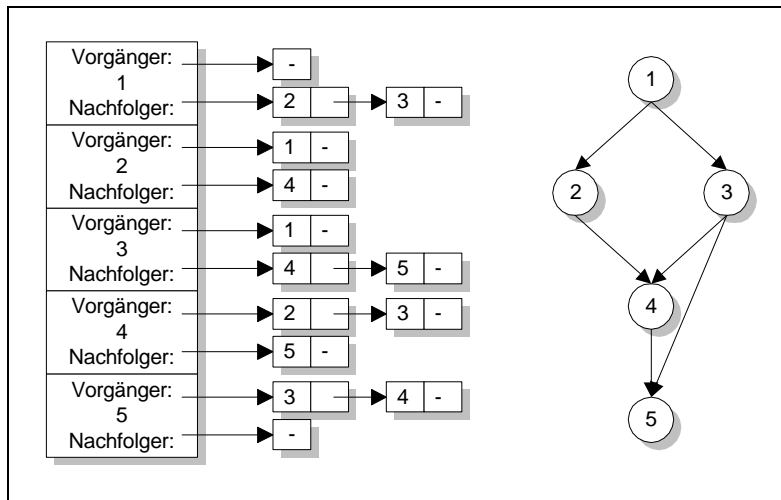


Abbildung 34: Adjazenzliste

Die Liste der Knoten wird als Array implementiert. Die Elemente des Arrays bestehen aus den Daten des Knoten und den beiden Listen für die Vorgänger und Nachfolger. Die Daten des Knoten sind der Bearbeiter, die Aufgabe und die fortlaufende Nummer des Arbeitsschrittes. Innerhalb der Adjazenzliste wird ein Knoten durch eine Knotennummer identifiziert. Um Speicherplatz zu sparen, wird die Knotennummer auch in den Listen der Vorgänger und Nachfolger abgespeichert. Um für gegebene Bearbeiter, Aufgabe und Nummer den entsprechenden Knoten zu finden, existiert eine Funktion, die zu (Bearbeiter, Aufgabe, Nummer) die entsprechende Knotennummer liefert.

5.3.3 Einlesen der Vorgangsdaten und Aufbau des Graphen

Für jeden Vorgang wird im Analyzer eine Adjazenzliste erzeugt. Die Knoten entsprechen dabei den Arbeitsschritten des Vorganges. Die Daten aus den Feldern UpdatedBy, UpdatedTask, Number und Predecessor der Note werden gelesen und in gleichnamige Arrays geschrieben. Feld i in diesen Arrays enthält Informationen zu einem Knoten im Graphen. Das Erzeugen der Adjazenzliste für einen Knoten geschieht dann wie folgt:

Für jeden Knoten v ist die Menge $\text{Predecessors}(v)$ wie folgt definiert:

$$\text{Predecessor}(v) := \{ w \in V \mid (w, v) \in E \}$$

$\text{BuildGraph}(G, \text{UpdatedBy}, \text{UpdatedTask}, \text{Number}, \text{Predecessors})$

1. $G = \text{new Graph}$
2. for $i = 1$ to NumOfNodes
3. $v = \text{new Vertex}$

4. $v.Bearbeiter = UpdatedBy(i)$
5. $v.Aufgabe = UpdatedTask(i)$
6. $v.Nummer = Number(i)$
7. $G.InsertVertex(v)$
8. for each $w \in Predecessor(i)$
9. $v.InsertEdge(w, v)$
10. next
11. next

Nachdem in Zeile 1 eine neue Adjazenliste erzeugt wurde, wird eine Schleife für alle Knoten durchlaufen. Als erstes wird ein neuer Knoten in Zeile 3 erzeugt. Diesem werden dann die Attribute Bearbeiter, Aufgabe und Nummer zugewiesen. Danach wird er in den Graph eingefügt. Anschließend werden alle Vorgänger w von v gelesen und die Kante (w,v) in den Graph eingefügt. Falls der Knoten w noch nicht existiert, wird er dabei automatisch erzeugt.

5.3.3.1 Algorithmus zur Bestimmung des Bearbeitungsschrittes eines Knoten

Nach dem Einlesen muß noch der Bearbeitungsschritt für die einzelnen Knoten berechnet werden. Der folgende Algorithmus weist jedem Knoten im Graphen einen Bearbeitungsschritt zu. Knoten ohne Vorgänger erhalten dabei den Bearbeitungsschritt 1. Diese Knoten werden auch als Startknoten bezeichnet. Für alle anderen Knoten v wird der längste Weg w von einem Startknoten zu diesem Knoten bestimmt. Ist k die Länge des Weges w , dann wird der Bearbeitungsschritt von v auf $k + 1$ gesetzt.

Gegeben ist ein gerichteter, azyklischer Graph $G = (V, E)$ mit Knotenmenge V und Kantenmenge E . Der Algorithmus verwendet die Arrays InGrad, Step und Color sowie eine first-in, first-out Queue Q . Das Array Steps enthält für jeden Knoten $v \in V$ den Bearbeitungsschritt. Das Array Color speichert für jeden Knoten die aktuelle Farbe. Ein Knoten kann die Farbe weiß, grau oder schwarz haben. Die Farbe wird verwendet, um zu bestimmen, ob ein Knoten in die Queue eingefügt werden soll oder nicht. Schwarze Knoten brauchen vom Algorithmus nicht mehr betrachtet werden. Sie wurden aus der Queue entnommen. Weiße Knoten besitzen mindestens einen Vorgänger, der nicht schwarz ist. Graue Knoten haben nur schwarze Vorgänger. Sie sind in der Queue

enthalten und ihnen wurde bereits ein Bearbeitungsschritt zugewiesen. $\text{Ingrad}(v)$ bestimmt die Anzahl der Kanten (w, v) mit $\text{Color}(w) \neq \text{schwarz}$.

$\text{Predecessors}(v)$ und $\text{Adj}(v)$ werden wie folgt definiert:

$$\text{Predecessors}(v) := \{ w \in V \mid (w, v) \in E \}$$
$$\text{Adj}(v) := \{ w \in V \mid \text{Color}(w) = \text{WHITE} \wedge (v, w) \in E \}$$

$\text{SetWorkstep}(G, \text{Step})$

1. $\text{CurStep} := 1$
2. for each vertex $v \in V$
3. $\text{InGrad}(v) := |\text{Predecessors}(v)|$
4. if $\text{InGrad}(v) = 0$ then
5. $\text{Step}(v) := \text{CurStep}$
6. $\text{Color}(v) := \text{GRAU}$
7. $\text{EnQueue}(Q, v)$
8. else
9. $\text{Color}(v) := \text{WEISS}$
10. fi
11. next
12. while $Q \neq \emptyset$
13. $v := \text{Head}(Q)$
14. $\text{DeQueue}(Q)$
15. $\text{NextStep} := \text{Step}(v) + 1$
16. $\text{Color}(v) := \text{SCHWARZ}$
17. for each $w \in \text{Adj}(v)$
18. $\text{InGrad}(w) := \text{InGrad}(w) - 1$
19. if $\text{InGrad}(w) = 0$ then
20. $\text{Step}(w) := \text{NextStep}$
21. $\text{Color}(w) := \text{GRAU}$
22. $\text{EnQueue}(Q, w)$
23. fi
24. next
25. wend

Der Algorithmus baut auf dem Algorithmus zur Breitensuche²⁰ auf und arbeitet wie folgt. In den Zeilen 1 - 11 werden die Knoten und die Queue initialisiert. In Zeile 3 wird der Ingrad für jeden Knoten auf die Anzahl der Vorgänger gesetzt. In den Zeilen 5 - 7 werden die Knoten, die keinen Vorgänger haben, grau markiert und in die Queue eingefügt. Ihr Bearbeitungsschritt wird auf 1 gesetzt. Für die Knoten, die mindestens einen Vorgänger haben, wird in Zeile 9 die Farbe auf weiß gesetzt. Der eigentliche Algorithmus beginnt in Zeile 12. Solange die Queue nicht leer ist, wird das erste Element aus der Queue entnommen und schwarz gefärbt. In Zeile 15 wird der Bearbeitungsschritt für die nächsten Knoten bestimmt. Danach wird für alle Nachfolger, die noch weiß sind, der Ingrad um eins verringert. Dies wird getan, da der Knoten v jetzt schwarz ist und $\text{InGrad}(w)$ nur die Kanten von nicht schwarzen Vorgängern zählt. Wurde mit dem Knoten v der letzte Vorgänger von w schwarz gefärbt, dann ist jetzt $\text{InGrad}(w) = 0$. Der Bearbeitungsschritt von w wird gesetzt, anschließend wird w grau gefärbt und in die Queue eingefügt.

5.3.3.2 Laufzeit des Algorithmus

Die Queue Operationen Head, DeQueue und EnQueue können in $\Theta(1)$ Zeit ausgeführt werden. Die Anzahl der Vorgänger wird im günstigsten Fall beim Erzeugen des Graphen in einer zusätzlichen Variablen festgehalten. Dann kann Zeile 3 in $\Theta(1)$ ausgeführt werden. Im ungünstigsten Fall müssen die Vorgänger gezählt werden. Da die Vorgänger jedes Knoten in einer zusätzlichen Liste gespeichert werden, kann auf sie direkt zugegriffen werden. Das Zählen wird für jeden Knoten einmal gemacht. Insgesamt wird für Zeile 3 dann $O(E)$ Zeit benötigt. Die Zeilen 2 bis 10 benötigen also im günstigsten Fall $\Theta(V)$, im ungünstigsten $O(V+E)$ Zeit. Ein Knoten wird nach der Initialisierung nicht wieder weiß eingefärbt. Der Test in Zeile 19 stellt sicher, daß nur weiße Knoten in die Queue eingefügt werden. Daher wird ein Knoten höchstens einmal in die Queue eingefügt. Die Adjazenzliste eines Knoten wird nur dann durchlaufen, wenn der Knoten

²⁰ Algorithmus um einen Graph zu durchsuchen. Dabei wird bei einem Startknoten s begonnen und es werden alle Nachfolger dieses Knoten nacheinander besucht und die Länge des Weges vom Startknoten berechnet. Danach werden die Nachfolger der bereits besuchten Knoten besucht. Allgemein werden erst alle Knoten mit Distanz k von s besucht und danach alle Knoten mit Distanz $k + 1$. Vgl. [Corman, Leiserson, Rivest, 1990, S. 469 ff.]

aus der Queue entnommen wurde. Jeder Knoten wird höchstens einmal entnommen und die Gesamtlänge der Adjazenzlisten beträgt $\Theta(E)$. Daher wird die Schleife ab Zeile 17 $O(E)$ mal durchlaufen. Der Test in Zeile 19 kann in $\Theta(1)$ ausgeführt werden. Da die Zeilen 20 - 22 $\Theta(1)$ Zeit benötigen, können die Zeilen 18-24 insgesamt in $O(E)$ ausgeführt werden. Der Rest der Schleife wird $O(V)$ mal ausgeführt, da jeder Knoten höchstens einmal in die Queue eingefügt und somit auch nur einmal entnommen werden kann. Insgesamt ergibt sich so eine Laufzeit von $O(V+E)$ für die Schleife ab Zeile 12. Die Initialisierung in den Zeilen 1 bis 11 kann im ungünstigsten Fall auch in $O(V+E)$ ausgeführt werden. Daraus ergibt sich eine Gesamtlaufzeit von $O(V+E)$.

6 Grenzen und Risiken der Analyse

6.1 Rechtliche Aspekte

Zur Analyse der Vorgänge werden umfangreiche und detaillierte Daten benötigt. Diese Daten enthalten Informationen über die Mitarbeiter und deren ausgeführte Tätigkeiten. Diese Informationen können dann aufbereitet und ausgewertet werden. Es ist möglich für einzelne Mitarbeiter zu analysieren, welche Tätigkeiten sie wie oft ausgeführt haben. Weiterhin können die Kommunikationsbeziehungen einzelner Mitarbeiter untereinander ausgewertet werden. Da es sich hierbei um personenbezogene Daten handelt, muß auf jeden Fall das Bundesdatenschutzgesetz beachtet werden. Weiterhin muß das Betriebsverfassungsgesetz berücksichtigt werden, denn es ist auch eine Kontrolle der Arbeit einzelner Mitarbeiter möglich. Für genauere Informationen verweise ich auf die Diplomarbeit von Katrin Wolke²¹.

6.1.1 Datenschutz

Durch den Einsatz moderner EDV ist der Umgang mit großen Datenmengen vereinfacht worden. Zugleich ist der Informationsbedarf von Staat und Wirtschaft enorm angestiegen. Die Folge ist, daß immer mehr Daten gespeichert und ausgewertet werden. Hiermit sind aber hohe Risiken verbunden. Der Einzelne hat keinen Überblick darüber, welche Stellen welche Informationen über ihn gespeichert haben und wie diese genutzt werden. Es ist auch nahezu unmöglich dies herauszufinden. Weiterhin ist es nur schwer möglich, die Richtigkeit der gespeicherten Informationen zu überprüfen.

Ein Beispiel ist Direktwerbung per Post. Die absendenden Firmen haben häufig detaillierte Informationen über die potentiellen Kunden. Der Kunde selbst hat aber meist keinen Überblick darüber, welche Informationen diese Firmen besitzen und woher sie diese erhalten haben.

Um einem Mißbrauch von personenbezogenen Daten vorzubeugen, muß der Umgang mit ihnen reguliert werden. Diese Aufgabe hat das Bundesdatenschutzgesetz (BDSG),

²¹Vgl. [Wolke, 1994, S. 35]

das 1977 erlassen wurde. Es gilt für öffentliche Stellen des Bundes, der Länder und für nicht öffentliche Stellen.²²

Im BDSG heißt es in §13 (2): „Das Erheben personenbezogener Daten ist zulässig, wenn ihre Kenntnis zur Erfüllung der Aufgabe der erhebenden Stelle erforderlich ist.“ und in §28 (1): „Das Speichern, Verändern oder Übermitteln personenbezogener Daten oder ihre Nutzung als Mittel für die Erfüllung eigener Geschäftszwecke ist zulässig ... 2. Soweit es zur Wahrung berechtigter Interessen der speichernden Stelle erforderlich ist und kein Grund zu der Annahme besteht, daß das schutzwürdige Interesse des Betroffenen an dem Ausschluß der Verarbeitung oder Nutzung überwiegt. ...Die Daten müssen nach Treu und Glauben und auf rechtmäßige Weise erhoben werden.“

Im Falle der Aufzeichnung von Tracking Informationen liegt ein berechtigtes Interesse vor. Ohne diese Informationen ist es nur sehr schwer möglich, Vorgänge zu optimieren. Da es sich dabei um Vorgänge handelt, die dem Geschäftsbetrieb dienen, ist davon auszugehen, daß die schutzwürdigen Interessen der Betroffenen nicht überwiegen. Entscheidend für die Erhebung der Daten ist, daß die Betroffenen über die Datenerhebung und deren Nutzung informiert werden. Eine heimliche Datenerhebung oder die Täuschung der Betroffenen sind mit Treu und Glauben in aller Regel nicht zu vereinbaren. Weiterhin müssen Vorkehrungen getroffen werden, um einem Mißbrauch zu verhindern²³.

6.1.2 Betriebsverfassungsgesetz

Das Betriebsverfassungsgesetz (BetrVG) reguliert die verfassungsmäßigen und demokratischen Rechte zwischen Arbeitnehmern und Arbeitgebern innerhalb eines Unternehmens. Unter anderem wird darin geregelt, inwieweit ein Arbeitgeber die Mitarbeiter bei der Ausführung ihrer Tätigkeit überwachen darf. Im §87 (6) des BetrVG heißt es: „Der Betriebsrat hat ... in folgenden Angelegenheiten mitzubestimmen: ... Einführung und Anwendung von technischen Einrichtungen, die dazu bestimmt sind, das Verhalten oder die Leistung der Arbeitnehmer zu überwachen.“

²² §1,2 BDSG

²³ §9 BDSG

Dazu kann der Analyzer unter Verwendung der Tracking Informationen verwendet werden. Sein Einsatz muß, falls die Bearbeiter einer Aufgabe abgespeichert werden, vom Betriebsrat genehmigt werden. Wird allerdings nur eine Gruppe oder Abteilung als Bearbeiter abgespeichert, so braucht dies nicht genehmigt werden²⁴. Dabei ist aber zu beachten, daß der tatsächliche Bearbeiter anhand des Vorganges ermittelt werden kann. Innerhalb eines Vorganges ist es häufig unumgänglich, die Bearbeiter einzelner Aufgaben abzuspeichern. Wenn zum Beispiel etwas bestellt, ein Angebot geschrieben oder ein Antrag genehmigt wird, muß der Bearbeiter abgespeichert werden.

Der Analyzer erlaubt nur eine mengenmäßige Auswertung von bearbeiteten Aufgaben eines Bearbeiters. Es ist möglich, herauszufinden, wie viele und welche Aufgaben ein Mitarbeiter bearbeitet hat. Es ist auch möglich, diese Angaben für ein gewähltes Intervall zu erhalten. Wie lange ein Bearbeiter für eine einzelne Aufgabe benötigt hat, kann aber nicht ermittelt werden. Eine solche Option wäre rechtlich bedenklich und würde sehr wahrscheinlich am Betriebsrat scheitern. Weiterhin ist es mit Notes Mitteln nahezu unmöglich, die Bearbeitungszeit einzelner Aufgaben exakt zu ermitteln. Daher wären die Ergebnisse einer zeitbezogenen Analyse sehr ungenau bis völlig falsch.

6.2 Technische Grenzen

Auch die mit der Version 5.0 von Visual Basic erzeugten Programme haben noch Laufzeitprobleme. Insbesondere die Grafikausgabe ist sehr langsam. Beim Analyzer merkt man dies sehr deutlich, wenn man die Kantenbeschriftung einschaltet und danach die Knoten per drag and drop verschiebt.

Um Bearbeitungszeiten von Vorgängen festzuhalten, könnte man wie folgt vorgehen. Wenn ein Anwender ein Dokument zum Bearbeiten öffnet, wird ein Ereignis erzeugt, auf das reagiert werden kann. Der Zeitpunkt wird als Bearbeitungsbeginn festhalten. Wenn das Dokument dann gespeichert und geschlossen wird, könnte dieser Zeitpunkt als Bearbeitungsende erfaßt werden. Der Anwender kann das Dokument aber beliebig lange geöffnet lassen und in der Zwischenzeit weitere Dokumente bearbeiten, telefonieren oder eine Pause machen. Es ist unmöglich festzustellen, ob der Mitarbeiter den Vorgang bearbeitet oder nicht. Weiterhin kann zur Bearbeitung eines Vorganges auch das reine

²⁴ Vgl. [Wolke, 1994, S. 35]

Lesen des Dokumentes gehören. Dazu muß es nicht zum Bearbeiten geöffnet werden. Ein Dokument kann aber auch gelesen werden, um für einen anderen Vorgang etwas nachzuschauen. Dies führt dazu, daß es unter Notes nicht möglich ist die Bearbeitungszeiten von Dokumenten auch nur einigermaßen genau zu ermitteln. Neben den bereits erwähnten rechtlichen Aspekten ein weiterer Grund darauf ganz zu verzichten.

Ein weiteres Problem bei der Analyse besteht darin, daß nur quantifizierbare Daten erfaßt und ausgewertet werden. Wichtige qualitative Aspekte, wie die Kunden oder Mitarbeiterzufriedenheit, werden nicht betrachtet.

6.3 Soziale Gesichtspunkte

Immer wenn es um Automatisierung von Arbeiten geht, müssen auch die sozialen Aspekte betrachtet werden. Die Mitarbeiter, die ein Workflow System benutzen sollen, müssen dieses auch akzeptieren. Aber gerade die Akzeptanz des Trackings und der Analyse können ein großes Problem darstellen. So sind für eine exakte Analyse auch exakte Daten nötig. Wenn das System aber nicht akzeptiert wird, werden die Mitarbeiter versuchen, die Lücken und Fehler im System auszunutzen, um sich einer Kontrolle zu entziehen. Um diesem vorzubeugen, ist es nicht zwingend notwendig, beim Tracking die Namen der Bearbeiter zu speichern. Es können auch Rollen, Gruppen oder Abteilungen als Bearbeiter abgespeichert werden. Sollte auch dieses ein Problem darstellen, so kann auf die Bearbeiter ganz verzichtet werden und eine rein aufgabenbezogene Analyse durchgeführt werden.

Man sollte sich immer darüber im klaren sein, daß mit einem technischen Produkt, wie dem WAGS auch nur technische Probleme gelöst werden können. Es ist nicht möglich, soziale Probleme zu lösen. Bei der Analyse sollte man daher immer versuchen, die Ursachen für Probleme im Ablauf der Vorgänge zu ermitteln.

7 Ausblick

7.1 Erweiterung der Funktionalität des WAGS Analyzers

Um die Funktionalität zu erweitern, sollte eine direkte Schnittstelle zum Modeler geschaffen werden. Dadurch könnte ein besserer Soll-Ist Vergleich von Vorgängen durchgeführt werden. Auch könnte die Darstellung der Graphen an die Darstellung im Modeler angepaßt werden. Es könnten zum Beispiel die gleichen Icons verwendet werden. Dadurch wäre es möglich Abläufe besser wieder zu erkennen.

Durch das Abspeichern von mehr Trackinginformationen und das anschließende Auswerten könnte die Funktionalität weiterhin stark erweitert werden. Insbesondere wäre es von Interesse, bei bedingten Verzweigungen im Ablauf die entsprechenden Bedingungen und Parameter der Workflowinstanzen auszuwerten.

7.2 Unterschiede zum GroupFlow Analyzer

Einige Features, die im GroupFlow Analyzer enthalten sind, wurden nicht implementiert. So werden die Icons den einzelnen Knoten im Graphen automatisch zugeordnet und können vom Anwender nicht verändert werden. Beim Anzeigen der bearbeiteten Vorgänge wird nicht nach erstellten und bearbeiteten Vorgängen unterschieden. Die „receives to“ und „receives from“ Optionen der „Workflow Agent“ Ansicht wurden auch nicht implementiert. Dies geschah hauptsächlich aus Zeitgründen. Der Schwerpunkt dieser Arbeit liegt auf neuen Möglichkeiten, insbesondere um verteilte Vorgangsbearbeitung analysieren zu können. Da der Quelltext des GroupFlow Analyzers nicht weiter verwendet werden konnte, mußte alles neu implementiert werden. Eine Kopie mit zusätzlichen Funktionen hätte den zeitlichen Rahmen von 4 Monaten gesprengt.

7.3 Verbesserung des Laufzeitverhaltens und des Ressourcenbedarfs des WAGS Analyzers

Um das Laufzeitverhalten zu verbessern, könnten einige Programmteile in C++ programmiert werden. Besonders nützlich wäre ein OCX²⁵, das einen Pfeil mit Beschriftung zur Verfügung stellt. Im Augenblick wird dieser Pfeil in den Grafiken aus drei Linien und einem Label zusammengesetzt. Dies ist sehr langsam und verbraucht recht viele Ressourcen. Weiterhin können die Linien nicht auf das Mausklick Ereignis reagieren. Mit einem entsprechendem OCX wäre es möglich, auf dieses Ereignis zu reagieren und zum Beispiel eine Liste aller Vorgänge anzuzeigen, die über eine Kante weitergeleitet wurden. Es könnte aber auch die gesamte Grafik als ein OCX programmiert werden. Dies würde eine deutlich schnellere Grafikausgabe erlauben. Weiterhin kann die grundlegende Datenstruktur der Adjazenzlisten als DLL²⁶ implementiert werden. Auch dadurch ist ein Geschwindigkeitsgewinn möglich.

²⁵ OLE Control Extension: Spezielle OLE Objekte, die bei der Anwendungsentwicklung verwendet werden können.

²⁶ Dynamic Link Library: Eine Sammlung von Funktionen, die von anderen Programmen verwendet werden können.

8 Zusammenfassung

An der Universität Gesamthochschule Paderborn wurde im Rahmen des GroupFlow Projektes ein Werkzeug zur Analyse von Workflows entwickelt. Mit diesem GroupFlow Analyzer können bearbeitete Vorgänge visualisiert und analysiert werden. Die dabei erzeugten Graphen enthalten als Knoten die Bearbeiter der betrachteten Vorgänge.

Im Rahmen dieser Diplomarbeit wurden die Konzepte des GroupFlow Analyzers aufgegriffen und weiterentwickelt. Dazu wurden zuerst grundlegende Begriffe aus den Bereichen Groupware und Workflow Management vorgestellt. Anschließend wurden die besonderen Anforderungen an verteiltes Workflowmanagement erörtert sowie das Wide Area GroupFlow System und die zugrundeliegende Basis-Umgebung Lotus Notes vorgestellt.

Danach wurden die Anforderungen an ein Analysewerkzeug herausgearbeitet. Dazu wurden neben den Konzepten des GroupFlow Analyzers auch Arbeiten anderer Autoren berücksichtigt. Als eine der wichtigsten Anforderungen wurden dabei die Visualisierung von Workflows und deren statistische Auswertung identifiziert. Im Rahmen des DWM sind insbesondere Darstellungen auf Abteilungs- oder Organisationsebene erforderlich. Weiterhin sind die erhöhten Sicherheitsbedürfnisse und die Probleme verteilter Ausführung zu beachten.

Im Anschluß daran wurden Konzepte zur Erfüllung der Anforderungen vorgestellt. Weiterhin wurde dargelegt, wie der WAGS Analyzer diese Konzepte umsetzt. Insbesondere ist hier die Verwendung von hierarchischen Namen für Bearbeiter von Aufgaben zu nennen. Dadurch stehen vielfältige Möglichkeiten zur Datenverdichtung zur Verfügung. Es werden Ansichten auf Abteilungs- oder Organisationsebene möglich. Weiterhin können als Bearbeiter beim Tracking auch Abteilungen, Rollen oder Organisationen abgespeichert werden, um so den Sicherheitsanforderungen gerecht zu werden. Weiterhin besteht die Möglichkeit, Zeitpunkt und Zeitraum bezogene Analysen durchzuführen. Durch das Erstellen von Differenzworkflows kann so die Entwicklung innerhalb eines Zeitraumes betrachtet werden. Die Daten stehen in Excel Tabellen und Diagrammen zur Verfügung, wo sie weiter bearbeitet werden können.

Da der Sprachumfang der verwendeten Programmiersprache Visual Basic 5.0 gegenüber der Version 3, in der der GroupFlow Analyzer implementiert wurde, stark erweitert wurde, konnte der vorhandene Quelltext nicht wieder verwendet werden. Der WAGS Analyzer Prototyp wurde komplett neu implementiert. Einige der Implementierungsdetails wurden auch in der Arbeit vorgestellt.

Weiterhin wurde auf Grenzen und Risiken der Analyse eingegangen. Ein wichtiges Ergebnis dabei ist, daß eine Bearbeitungszeit bezogene Analyse von Workflows erlaubt ist. In der Praxis ist sie jedoch vom Betriebsrat zu genehmigen und somit nur schwer einzuführen. Weiterhin ergeben sich technische Probleme, die Bearbeitungszeiten von Vorgängen oder einzelnen Arbeitsschritten exakt zu erfassen. Aus diesen Gründen wurde davon abgesehen, eine zeitbezogene Auswertung zu implementieren.

Abschließend wurden mögliche Verbesserungen und Erweiterungen des WAGS Analyzer Prototyps genannt. Hierzu zählen neben den fehlenden Funktionen des GroupFlow Analyzers vor allem der Einsatz von DLL's oder OCX'en, um das Laufzeitverhalten und den Ressourcenbedarf zu verbessern.

9 Anhang

9.1 Data Dictionary

Zur Beschreibung der Daten wurde eine modifizierte Backus-Naur-Form verwendet.²⁷

Notation:

Symbol	Bedeutung	Beispiel	
=	ist äquivalent zu	$A = A$	
+	Sequenz	$A = B + C$	
{ }	Wiederholung, 0 mal oder öfter	$A = \{B\}$	$A = BBB..$
$M\{ \}N$	Wiederholung von M bis N	$A = 1\{B\}2$	$A = B$ oder $A = BB$
()	Option, = 0{ }1	$A = B + (D)$	$A = B$ oder $A = BD$
* *	Kommentar	$A = * \text{Kommentar} *$	
„ „	Schließt Zeichenkette ein	$A = \text{„Zeichenkette“}$	$A = \text{Zeichenkette}$

²⁷ Vgl. [Balzert, 1996, S. 122 ff]

9.2 Literaturangaben

Balzert, Helmut (1996): Lehrbuch der Software-Technik. Spektrum, Akademischer Verlag, 1996

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990): Introduction to Algorithms; MIT Press, 1990

Derszteler, Gérard (1997): WorkFlow Analyzer - Prozeßbewertung auf Basis von Workflow-Protokollen. Aus Arbeitsbericht Nr. 54, Organisatorische und technische Aspekte beim Einsatz von Workflow Management Systemen, Institut für Wirtschaftsinformatik der Universität Münster, 1997

Hilpert, Wolfgang (1994): Effiziente und wirtschaftliche Einsatzplanung von Workflow Management-Systemen auf Groupware-Basis - vom Pilotprojekt zur unternehmensweiten Lösung. Paper, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Universität-Gesamthochschule Paderborn, 1994

Hollingsworth, David (1996): Workflow Management Coalition - The Workflow Reference Model, Vers. 1.1. Workflow Management Coalition, 1996

Kuri, Jürgen (1996): Wenn der Postmann zweimal klingelt - Namen und Adressen im TCP/IP-Netzwerk und im Internet. c't, 12/1996, S. 334 ff

Lauer, Thomas (1993): Paketdienst - Grundlagen zu Object Linking and Embedding. c't, 4/1993, S. 242 ff

Lichtner, Viktor (1997): Effektiveres Arbeiten durch Workgroup-Computing - Trend zum Team. Gateway, 2/1997, S. 36-38

Nastanski, Ludwig; Schicker, Till; Behrens, Olav; Ehlers, Peter (1995): Büroinformationssysteme. In: Bausteine der Wirtschaftsinformatik. S+W Steuer und Wirtschaftsverlag GmbH, Hamburg 1995, S. 267-369

Piesk, Jens; Pötter, Klaus (1994): Grafisches Analysewerkzeug zur Visualisierung von Workflow in Datenbankanwendungen unter Lotus Notes (Workflow Analyzer). Seminararbeit, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Universität-Gesamthochschule Paderborn, 1994

Riempp, Gerold; Nastansky Ludwig (1996): Workflow Management between distributed organizations - the Wide Area GroupFlow System. Paper, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Universität-Gesamthochschule Paderborn, April 1996

Wolke, Karin (1994): Business Process Management in Distributed Systems - Concept and development of a protocol tracking module for a distributed workflow management system reflecting open as well as predefined processes; Diplomarbeit, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Universität-Gesamthochschule Paderborn, 1994