



Universität-Gesamthochschule Paderborn

Diplomarbeit

***Change Management im Bereich von Plattform-Technologien
für das Informationsmanagement und die Kommunikation in
Organisationen - am Beispiel/Szenario einer Großbank***

vorgelegt bei

Prof. Dr. Nastansky

im März 1998

vorgelegt von

Ralf Lutter
Am Mühlengraben 16
32676 Lügde-Elbrinxen

Matrikelnummer 3059232

Studiengang Wirtschaftsinformatik

Bedanken möchte ich mich bei allen Mitarbeitern der beteiligten Firmen für die gute Zusammenarbeit.

Mein besonderer Dank gilt allen Freunden, die es mir erst durch ihre Unterstützung ermöglicht haben, die Kraft für diese Diplomarbeit in einer für mich sehr schwierigen, privaten Zeit aufzubringen.

Inhaltsverzeichnis

1	Einführung.....	1
1.1	Szenario und Motivation	1
1.2	Zielsetzung.....	2
1.3	Aufbau der Arbeit.....	3
2	Thematische Abgrenzung und begriffliche Definitionen	4
2.1	Plattform-Technologien.....	4
2.2	Change Management als Teil des Systems Management.....	6
2.3	Workflow Management.....	7
3	Anforderungen und Konzepte aus Theorie und Praxis.....	10
3.1	Systems Management-Modelle.....	10
3.2	Anforderungen an Workflow Management-Systeme	15
3.3	Grundsätze der Software-Ergonomie	18
4	Konkretes Szenario der Deutschen Bundesbank	22
4.1	Systems Management-Ansatz.....	22
4.2	Projektumfeld	23
4.2.1	Projekthintergrund/Motivation.....	23
4.2.2	Projektziele.....	24
4.2.3	Projektverlauf.....	25
4.3	Umsetzung des Change Management-Prozesses.....	26
4.3.1	Konzept der Geschäftsvorfälle und Change-Typen.....	26
4.3.2	Prozeßstruktur	27
4.3.3	Rollen und Aufgabenbereiche	29
4.3.4	Übersicht über den Prozeßfluß.....	30
4.3.5	Qualitätsparameter	31

4.3.6	Eskalationsmanagement	33
4.3.7	Systems Management Portfolio-Datenbank	34
5	Verwendete Arbeitsumgebung.....	36
5.1	Lotus Notes	36
5.2	Genii Midas Rich Text LSX	39
5.3	PAVONE Espresso	40
5.3.1	Espresso Datenbanken.....	42
5.3.2	Espresso ProcessModeler	44
5.3.3	Espresso OrganizationModeler	45
6	Ausgewählte Implementierungsaspekte des Prototypen	46
6.1	Überblick	46
6.1.1	Architekturmodell	46
6.1.2	Realisierter Prozeßverlauf	47
6.2	Modularer Lösungsansatz	49
6.2.1	Vergleich möglicher Prozeßarchitekturen.....	49
6.2.2	Synchronisation von Abhängigkeiten an einem Praxis-Beispiel.....	51
6.2.3	Hintergrundverarbeitung zur Realisierung der benötigten Funktionalität	53
6.3	Die Anforderungsliste als zentrale Komponente	55
6.3.1	Benötigte Architekturmerkmale	55
6.3.2	Überblick über die technische Realisierung	57
6.4	Benutzerführung und -interaktion.....	59
6.4.1	Konzeptionelles Maskendesign.....	60
6.4.2	Ausgewählte Aspekte der allgemeinen Benutzerführung	62
6.4.3	Benutzerinteraktion in der Portfolio-Datenbank	65
6.4.4	Bewertung auf Basis der Grundsätze zur Dialoggestaltung	66
7	Fazit und Perspektiven	68

A Verzeichnisse	70
A.1 Literatur	70
A.2 Abbildungen	78
A.3 Tabellen	80
A.4 Abkürzungen	80
B Eidesstattliche Erklärung.....	82
C Anlagen	83
C.1 Beschaffungsantrag in Papierform.....	83
C.2 Endanwenderschulung	84
C.2.1 Agenda	84
C.2.2 Auswertungsbogen	85
C.2.3 Ergebnis der Befragung.....	86
C.3 Realisierter Prototyp	88
C.3.1 Dateien des Prototypen.....	89
C.3.2 Anforderungen an Hard- und Software	89
C.3.3 Installationsanleitung.....	89
C.3.4 De-Installationsanleitung.....	91
C.3.5 Dokumentation	91
C.4 Online-Quellen	92
C.5 Diplomarbeit in elektronischer Form.....	93

1 Einführung

1.1 Szenario und Motivation

Mit der Neuausrichtung der Geschäftsoperationen fand Anfang der 90er Jahre ein Paradigmenwechsel in der Datenverarbeitung statt: Client-/Server-Architekturen lösten die Großrechner-zentrierten Strukturen ab und schrittweise wurden die alleinstehenden Abteilungs- und Büro-PCs zu immer größer werdenden Netzwerken zusammengeschlossen [vgl. Dehn 1995, S. 17; Magura 1995, S. 20; Wess 1996, S. 23 f.].

Die Betreuung dieser verteilten Umgebungen und ihrer Anwendungen ist aber im Vergleich zum Mainframe-Ansatz viel komplexer, wobei die Gründe hierfür vielschichtig sind und in groben Zügen der Tabelle 1.1-1 entnommen werden können. Als Hauptgrund hierfür ist die Heterogenität in Form unterschiedlicher Rechnerarchitekturen sowie Netzwerk- und Datenbanksystemen anzusehen, die durch Unternehmensfusionen, Akquisitionen und Joint-Ventures, aber auch durch unterschiedliche Einkaufsmethoden innerhalb einer Organisation entstehen [vgl. Arnold 1993, S. 14].

Mainframe-Architektur	Verteilte Umgebung
• homogen	• heterogen
• wenige zentrale Server	• viele verteilte Server
• zentrale Datenhaltung	• verteilte Datenhaltung
• statische Applikationen	• dynamische Applikationen
• IT-getrieben	• Endbenutzer-getrieben

Tab. 1.1-1: Verteilte Umgebung erhöht die Komplexität [vgl. Tivoli 1997]

Für die Zukunft erwartet das Marktforschungsunternehmen Gartner Group, daß neue Technologien wie Mobile Computing, Komponenten-basierte Software, Multimedia-Anwendungen und Telefon- sowie Videokonferenzsysteme die Komplexität verteilter Umgebungen sogar noch weiter erhöhen [vgl. Ottomeier 1996, S. 11]. Mit steigender Komplexität einer DV-Umgebung wachsen jedoch auch die Probleme überproportional an, wie das Marktforschungsunternehmen Meta Group festgestellt hat [vgl. Hamerschmidt 1998, S. 32]. Dies schlägt sich in einem erhöhten Support-Bedarf nieder und verlagert somit die Kostenverteilung (siehe Abbildung 1.1-1).

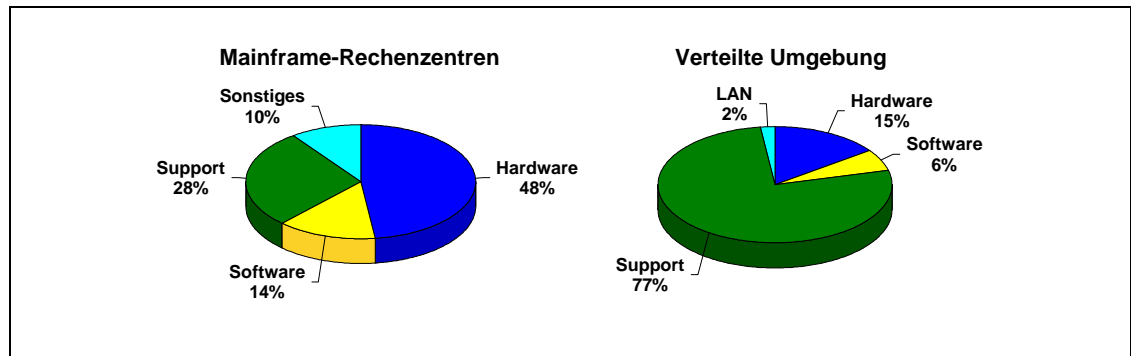


Abb. 1.1-1: Vergleich der DV-Kostenverteilung [vgl. Wesseler 1995, S. 21]

Da im Rahmen heutiger Kostensenkungsbestrebungen gerade aber auch die Datenverarbeitung als Sparpotential aufgefaßt wird, empfehlen Marktforschungsunternehmen den sogenannten Managed-PC, bei dem über das Systemmanagement Kostensenkungspotentiale erschlossen werden. Zur Bewertung der Effektivität wird dabei die Berechnung der Total Cost of Ownership (TCO) herangezogen, bei der nicht nur die Kosten der Hard- und Software ermittelt, sondern insbesondere auch die sogenannte Mitarbeiter-TCO mit eingeschlossen wird, die sich aus unproduktiven Arbeitszeiten auf Grund von Problemen mit dem PC ergibt. Durch unterschiedliche Berechnungsmethoden kommt es dabei zu abweichenden Kosten und damit verbundener Sparpotentiale zwischen 12% und 50% [vgl. Holzmann 1998, Koll 1998, Rademacher 1997, Wimmer 1997, Gursky/Thole 1996].

Zwar bietet sich mit dem Network Computer (NC) eine noch günstigere Alternative an, jedoch schützt diese nicht die getätigten Investitionen in die bereits vorhandene Hard- und Software. Außerdem rangiert die geringere TCO für IT-Manager nur auf Platz 5 der Hauptvorteile von NCs [vgl. o. V. 1997, S. 1].

Mit der ständig zunehmenden Komplexität verteilter Umgebungen und dem dadurch notwendigen Konzept des Managed-PCs ergibt sich als Motivation für diese Arbeit der Bedarf nach einer innovativen Lösung für das Change Management als zentralen Bestandteil des Systemmanagements.

1.2 Zielsetzung

Ein Schwerpunkt dieser Arbeit liegt in der Entwicklung einer einsatzfähigen Lösung zur Abbildung der administrativen Geschäftsprozesse zum Change Management für den Client-/Server-Bereich der Deutschen Bundesbank mit Hilfe eines elektronischen Workflow Management-Systems.

Diese Entwicklung wurde im Rahmen eines Projekts vorgenommen, in dem die IBM Unternehmensberatung die vorhandenen Geschäftsprozesse analysierte und in Kooperation mit der Firma PAVONE Informationssysteme auf Basis der Rahmenanforderungen zum Systems Management in ein realisierbares Modell überführte. Dieses wurde dann von der PAVONE Informationssysteme praktisch implementiert, wobei der Autor als Consultant dieses Unternehmens mit für die Überführung und Umsetzung des Modells verantwortlich war.

Der vorliegende schriftliche Teil der Diplomarbeit soll die realisierte Lösung in zentralen Implementierungsaspekten vorstellen und in einen theoretischen und praktischen Bezugs- und Relevanzrahmen einordnen.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in sieben Hauptkapitel.

Im zweiten Kapitel werden die grundlegenden Themengebiete abgegrenzt und alle wichtigen Begriffe zur konsistenten Nutzung im Rahmen dieser Arbeit definiert.

Kapitel 3 bettet die Themenstellung in ein Rahmenkonzept mit Ansprüchen aus Theorie und Praxis ein.

Im Anschluß wird in Kapitel 4 das konkrete Szenario der Deutschen Bundesbank beschrieben. Dazu wird schwerpunktmäßig ein umfassender Überblick über den erarbeiteten theoretischen Rahmen für die praktische Implementierung des Change Management-Prozesses gegeben.

Um dem Leser das Verständnis der umgesetzten Implementierung zu erleichtern, werden im fünften Kapitel die zu deren Realisierung genutzten Plattform-Technologien beschrieben.

Kapitel 6 stellt dann die implementierte Lösung auf Basis des theoretischen Rahmens und der Plattform-Technologien vor. Begonnen wird mit einem Überblick über das Architekturmodell und den realisierten Prozeßverlauf, bevor drei zentrale Implementierungsaspekte ausführlich erläutert werden.

Abschließend wird in Kapitel 7 eine kritische Diskussion der konkreten Implementierung durchgeführt sowie Perspektiven zu deren Weiterentwicklung gegeben.

2 Thematische Abgrenzung und begriffliche Definitionen

Dieses Kapitel soll die für das Verständnis der Arbeit notwendigen Grundlagen erörtern, indem wichtige Begriffe der Plattform-Technologien sowie des Workflow und des Systems Managements eingeführt werden.

2.1 Plattform-Technologien

Plattform bedeutet laut Duden „Basis, Standpunkt, von dem ... ausgegangen wird“ [Duden 1996, S. 1157], und eine Technologie bezeichnet „technisches Wissen; Gesamtheit der technischen Kenntnisse, Fähigkeiten und Möglichkeiten“ [Duden 1996, S. 1521]. Damit kann eine *Plattform-Technologie* im Rahmen der Informatik als eine Basis verstanden werden, die gewisse technische Möglichkeiten bietet.

Das grundlegende DV-Paradigma stellt eine solche Plattform-Technologie dar. Früher war dieses durch den *Großrechner-Ansatz* gekennzeichnet, bei dem sowohl Daten als auch Applikationen von großen, zentralen Rechnern (auch Mainframe oder Host genannt [vgl. Schneider 1997, S. 514]) verwaltet wurden, während die Benutzer über Terminals auf diese Rechner zugriffen. Später wurde durch den Einsatz von Personal Computern (PCs) Verarbeitungintelligenz am Benutzerarbeitsplatz selbst zur Verfügung gestellt, was einen Wechsel zum *Client-/Server-Konzept* einleitete. Dieses nutzt die Möglichkeit, „Computer unterschiedlicher Leistungsklassen und Zweckbestimmungen über verteilte Anwendungsarchitekturen miteinander zu verbinden.“ „Client/Server bedeutet die Loslösung vom traditionellen Mainframe-Terminal-Modell zu vernetzten Umgebungen“ [Riemensperger 1993, S. 14]. Ein Server übernimmt dabei „die Funktion des Prozeß- und Daten-Servers“, während der Client „die Steuerung der graphischen Benutzerschnittstelle sowie die lokale Verarbeitung“ übernimmt [Riemensperger 1993, S. 14].

Im Rahmen dieser Arbeit wird mit *Groupware* eine weitere Plattform-Technologie eingesetzt, die auf dem Client-/Server-Paradigma basiert. „Groupware stellt einen generischen Begriff für spezialisierte computerbasierte Werkzeuge dar, die darauf ausgerichtet sind, für die kollaborative Arbeit in Teams eingesetzt zu werden.“ [Ott 1996, S. 1] Synonym wird oft auch der Begriff des Workgroup Computing verwendet [vgl.

Nastansky et al. 1994, S. 273], allerdings herrscht hier eine Begriffsunschärfe vor, so daß z. B. Kickuth den Bereich des Workgroup Computing nur als einen Teilbereich der Groupware auffaßt (vgl. Abbildung 2.1-1).

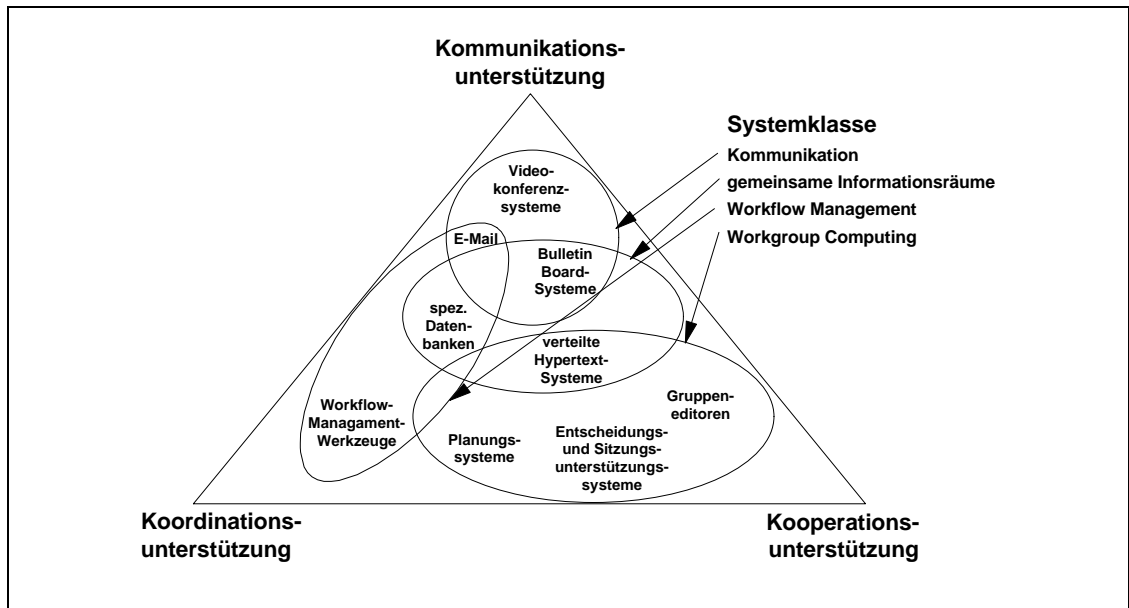


Abb. 2.1-1: Klassifikationsschema für Groupware [Kickuth 1996, S. 162]

Ein zentraler Aspekt von Groupware ist die *Kommunikation*. Hierunter versteht man „den expliziten Informationsaustausch zwischen verschiedenen Informationspartnern“, wobei „in erster Linie Raum- und Zeitdifferenzen überbrückt“ werden [Kickuth 1996, S. 161]. Eine Möglichkeit besteht dabei in der Nutzung von elektronischer Post, der sogenannten E-Mail. Als weiteres Kernaufgabengebiet von Groupware kann das *Informationsmanagement* angesehen werden, bei dem es sich um einen Managementansatz handelt, „in dessen Mittelpunkt die wirtschaftliche Versorgung betrieblicher Stellen mit Informationen steht, die zum Erreichen der Unternehmensziele beitragen.“ [Schneider 1997, S. 417] Dies geschieht häufig in Form gemeinsam genutzter Datenbanken, die in dieser Funktion auch als Repositories bezeichnet werden können (ein Repository ist ein „umfassendes Verzeichnis über alle Daten und Informationen innerhalb von EDV-Anwendungen“ [Schneider 1997, S. 724]).

Auf Basis der beiden erläuterten Plattform-Technologien lassen sich nun eigenständige Lösungen entwickeln, die die technischen Möglichkeiten ausnutzen. Die im Projekt realisierte Implementierung kann dabei als *Prototyp* bezeichnet werden, da ein Prototyp laut Duden eine „zur Erprobung und Weiterentwicklung bestimmte erste Ausführung (von Fahrzeugen, Maschinen o. Ä.)“ bezeichnet [Duden 1996, S. 1190]. Bei der Projekt-

implementierung handelt es sich um eine erste, vollständig einsetzbare Lösung, die noch nicht alle gewünschten Bereiche abdeckt, die aber in Zukunft entsprechend erweitert werden soll.

2.2 Change Management als Teil des Systems Management

In der direkten Übersetzung bedeutet *Change Management* soviel wie „Verwaltung von Änderungen“, weshalb der Begriff auch auf alle Formen von Änderungen angewandt wird. So werden z. B. Änderungen am Programmcode in der Applikationsentwicklung mit Hilfe eines Change Managements erfaßt, das Änderungsmanagement an Dokumenten als Change Management bezeichnet oder Änderungen in Unternehmen im Allgemeinen und im Rahmen der Geschäftsprozesse im Besonderen unter dem Begriff Change Management zusammengeführt [vgl. Reiß 1997]. Diese indifferente Nutzung läßt es dem Autor daher als besonders wichtig erscheinen, den Definitionsrahmen des Change Managements für diese Arbeit klar abzugrenzen.

Der Begriff „Change Management“ soll im Rahmen dieser Arbeit als Teil des *Systems Managements* verstanden werden, das übersetzt auch als *Systemmanagement* bezeichnet wird. Dieser Begriff wird häufig synonym für den Begriff des *Netzwerkmanagements* verwendet, was jedoch nicht völlig korrekt ist, weshalb zuerst eine Trennung dieser beiden Begriffe vorgenommen werden soll.

Das „Netzwerkmanagement kümmert sich um die Funktionalität, Wartung, Überwachung und Optimierung aller Komponenten eines Netzwerks“ [Matzer 1997, S. 16], womit alle physischen Bestandteile wie z. B. Router und Switches gemeint sind. Wichtigstes Element zur Erfüllung dieser Aufgaben ist dabei das Simple Network Management Protocol (SNMP), das fast 60 Prozent aller Netzwerkmanagementsysteme nutzen [vgl. Kauffels 1995, S. 252]. „Das Konzept von SNMP basiert .. auf einem Netzwerk-Management Programm (SNMP-Manager), welches z. B. auf einer Netzwerk-Management Workstation oder NMS (Network Management Station) implementiert ist, und SNMP-Agenten in den zu überwachenden Geräten. ... SNMP ist eine Funktionsammlung zur Kommunikation zwischen Management-Station und Agenten“ [Kauffels 1995, S. 237].

„Im Gegensatz dazu kümmert sich das Systemmanagement um die Vorgänge, Mitarbeiter und Werkzeuge. Im Fokus stehen die Ressourcen, die sicherstellen, daß die DV eines Unternehmens am effektivsten und produktivsten ausgenutzt wird.“ [Matzer 1997, S. 16] Als Ressourcen werden dabei sowohl die Hardware-Komponenten als auch die eingesetzten Betriebssysteme und Anwendungen aufgefaßt [vgl. Schneider 1997, S. 578]. Umfaßte das Systemmanagement ursprünglich nur den Großrechner-Bereich, so nahm mit der in Kapitel 2.1 angesprochenen Entwicklung zum Client-/Server-Computing sowie der Internet-Ausrichtung auch die Verwaltung der Rechnernetzwerke einen immer höheren Stellenwert ein [vgl. Matzer 1997, S. 16]. Dies führte zu einer Verzahnung von Netzwerk- und Systemmanagement, so daß heute das Systemmanagement den Bereich des Netzwerkmanagements einschließt.

Im Rahmen des Systems Managements umfaßt das Change Management damit alle Änderungen in der Hard- und Software-Konfiguration einer Organisation (z. B. neuer Rechner für eine Abteilung, Einbau einer neuen Festplatte oder Installation einer neuen Software), wobei eng mit den anderen Managementgruppen zusammengearbeitet wird [vgl. Kauffels 1995, S. 109 ff.]. In der deutschsprachigen Literatur werden synonym häufig auch die Begriffe Änderungsmanagement und Erweiterungsplanung benutzt.

Im weiteren Verlauf der Arbeit werden insbesondere noch die Begriffe *Change-Antrag* und *Change Request* verwendet. Diese bezeichnen synonym eine Beantragung einer Veränderung im Rechnerumfeld eines Endanwenders, wobei die (durchgeführte) Änderung an sich auch mit dem Begriff *Change* bezeichnet wird.

2.3 Workflow Management

„Ein *Workflow* [Herv. durch Verf.] ist eine endliche Folge von Aktivitäten, wobei die Folge von Ereignissen ausgelöst und beendet wird.“ [Kickuth 1996, S. 161] Dieser sehr allgemeine Ansatz wird daher an gleicher Stelle sofort näher spezifiziert als „organisationsweite arbeitsteilige Prozesse, in die viele Akteure einbezogen sind.“

Somit kann der Begriff des Workflows in der deutschen Sprache als Prozeß bezeichnet werden. Sofern es sich um einen Prozeß mit betriebswirtschaftlichem Hintergrund handelt, kann auch von einem Geschäftsprozeß gesprochen werden. Dieser „subsumiert im allgemeinen alle Leistungs-, Koordinations- und Administrationsprozesse eines Unternehmens.“ [Rose 1996, S. 320] Heilmann behandelt Workflow, Geschäftsprozeß

und Vorgang synonym als „ein abgrenzbarer, meist arbeitsteiliger Prozeß .., der zur Erstellung und Verwertung betrieblicher Leistungen führt.“ [Heilmann 1994, S. 9]

Wichtiger als eine Abgrenzung der Begriffe untereinander erscheint dem Autor für diese Arbeit die Unterscheidung von Begrifflichkeiten, die Teilschritte von Workflows bezeichnen. So spricht Kickuth lediglich von einer Folge von Aktivitäten [Kickuth 1996, S. 161]. Dagegen gehört für Becker/Vossen zur Spezifikation eines Workflows „eine Beschreibung derjenigen Aufgaben, aus denen er zusammengesetzt ist, sowie der Verarbeitungseinheiten, welche diese ausführen können“ [Becker/Vossen 1996, S. 20]. Mithin können zumindest Aufgaben und Aktivitäten als Teile eines Workflows unterschieden werden.

Innerhalb dieser Arbeit werden weiterhin noch die Begriffe des Subprozesses und des Subprozessschritts eingeführt, wobei die Beziehungen zwischen den einzelnen Begriffen in Abbildung 2.3-1 dargestellt werden: der Gesamtprozeß kann in verschiedene Subprozesse aufgeteilt werden, die wiederum in Teilschritte zerfallen, die ihrerseits eine Reihe von Aktivitäten beinhalten. Im Projekt wurde genau diese schrittweise Verfeinerung des Gesamtprozesses durchgeführt, bis auf der Aktivitätenebene eine genaue Prozeßbeschreibung vorlag. In der konkreten Implementierung wurden dann aufeinander folgende Aktivitäten, die dieselbe Verarbeitungseinheit aufweisen, zu Aufgaben zusammengefaßt, und als solche in der Benutzerführung abgebildet.

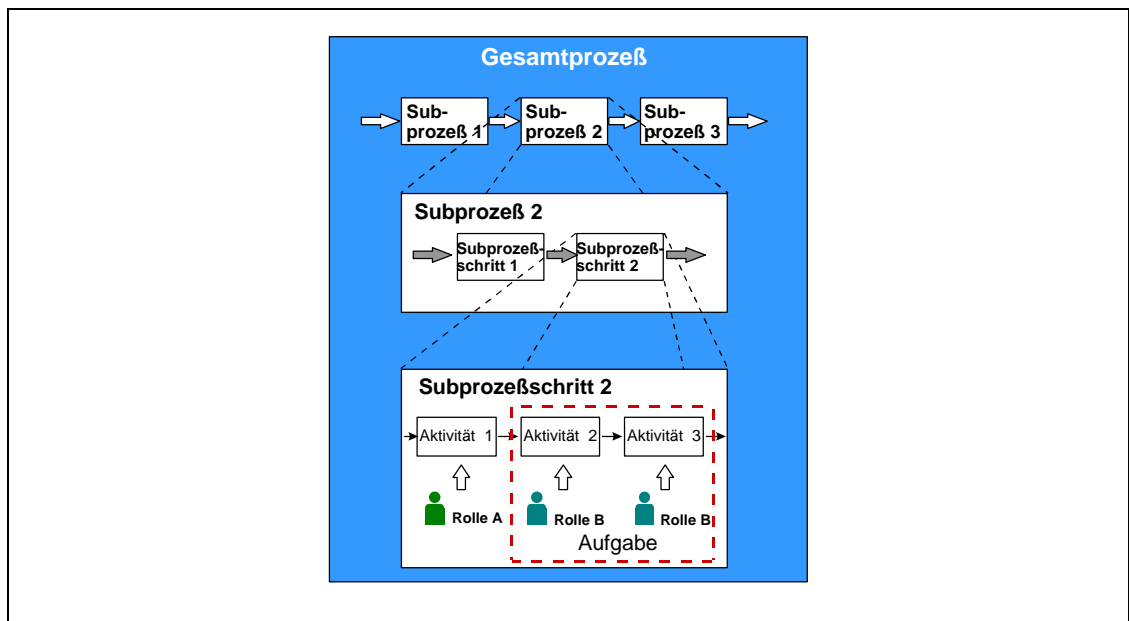


Abb. 2.3-1: Begriffsdefinitionen zum Workflow Management im Projekt [vgl. IBM 1998a, S. 35]

Die Abbildung deutet auch bereits eine Verbindung zwischen den einzelnen Elementen an, wobei bei einer Abhängigkeitsbetrachtung der Teileinheiten untereinander von einem Prozeßfluß gesprochen wird. Den größten Detaillierungsgrad zur Beschreibung eines Prozeßflusses erreicht die Aktivitätenabfolge, bei der jede einzelne Aktivität aufgeführt und beschrieben wird. Auf Grund des Projektumfangs betrachtet diese Arbeit aber nur eine Prozeßgrobstruktur, in der die Abhängigkeiten der Subprozesse oder Subprozeßschritte untereinander dargestellt werden.

Der Begriff des *Workflow Managements* umfaßt für Kickuth „alle Aufgaben, die bei der Modellierung, der Simulation sowie bei der Ausführung und Steuerung von Workflows erfüllt werden müssen.“ [Kickuth 1996, S. 161] Dabei faßt Kickuth das Workflow Management als einen Teilbereich der Groupware auf (vgl. Abbildung 2.1-1). Dagegen benutzt die Workflow Management Coalition den Begriff des Workflow Managements synonym für Workflow [vgl. WfMC 1996, S. 7] und kommt der Definition von Kickuth erst bei der konkreten Betrachtung eines *Workflow Management-Systems* nach: „Ein System, das die Ausführung von Workflows mit Hilfe von Software definiert, erstellt und verwaltet, laufend auf einer oder mehrerer Workflow Engines, und welches in der Lage ist, die Prozeßdefinition zu interpretieren, mit Workflow-Teilnehmern zu interagieren und, sofern benötigt, andere Tools und Applikationen aufzurufen.“ [WfMC 1996, S. 8 - Übersetzung durch den Autor] Erklärtes Ziel eines solchen Vorgangsteuerungssystems ist es, „die Logik zur Steuerung von Prozessen aus Anwendungsprogrammen herauszulösen.“ [Scheer et al. 1997, S. 29] Dies wird an gleicher Stelle mit dem Einsatz von Datenbankverwaltungssystemen verglichen, mit denen die Funktionen zur Datenverwaltung aus den Anwendungsprogrammen ausgelagert wurden.

Bei der Modellierung wird in einem Workflow Management-System ein *Vorgangstyp* (Synonym: Workflow-Klasse) entworfen, der die abstrakte Aufgabenfolge zur Behandlung einer bestimmten Klasse von Geschäftsvorfällen (z. B. Geschäftsvorfall „Angebotserstellung“) bezeichnet [vgl. Hasenkamp 1987, S. 56]. In der Ausführung wird dann ein *Vorgangsexemplar* (Synonym: Workflow-Instanz) als Instanzierung des Vorgangstypen erzeugt, welches einen konkreten Geschäftsvorfall abwickelt (z. B. „Angebotserstellung für das Change Management bei der Deutschen Bundesbank“).

3 Anforderungen und Konzepte aus Theorie und Praxis

Zur Einordnung in ein theoretisches und praktisches Rahmenkonzept sollen in diesem Kapitel zuerst verschiedene Ansätze des Systems Managements zur besseren Verdeutlichung der operativen Problemstellung vorgestellt werden. Anschließend wird auf allgemeine Anforderungen an ein in der prototypischen Implementierung als Basisplattform genutztes Workflow Management-System eingegangen, bevor abschließend Grundprinzipien der Software-Ergonomie als Einflußfaktoren der konkreten Implementierung im Rahmen von Workflow Management-Systemen diskutiert werden.

3.1 Systems Management-Modelle

Bedingt durch das in Kapitel 1.1 geschilderte Szenario und der in Kapitel 2.1 geschilderten Entwicklung des Systems Management ist auf dem Markt eine große Intransparenz durch eine Vielzahl unterschiedlicher Konzepte und Lösungen zu beobachten.

Generell kann dabei zwischen „punktorientierten Einzellösungen“, die eine „für eine Umgebung optimal angepaßte Lösung“ darstellen, und „Suiten“ unterschieden werden, die „eine Ein-Paket-Lösung aus verschiedenen, miteinander verknüpften Produkten eines Anbieters“ bilden. Dazwischen können sogenannte „Frameworks“ angesiedelt werden, die ein Gerüst für miteinander verbundene Punktlösungen bieten. So ermöglichen sie „die individuelle Anpassung einer Managementlösung an die spezifischen Anforderungen im Unternehmen“ [Matzer 1997, S. 16].

Diese Hersteller-Frameworks (allgemein auch Modelle genannt), die auch von Normierungsgremien wie z. B. der International Standards Organization (ISO), von unterschiedlichen Herstellerzusammenschlüssen wie der Open Software Foundation (OSF), von Marktforschungsunternehmen wie der Meta Group und auch in der Literatur dargestellt werden, weisen sämtlich unterschiedliche Teilbereiche auf, so daß kein einheitliches, gesamtgültiges Modell des Systems Management existiert. Der Autor möchte daher im Folgenden einige ausgewählte Frameworks vorstellen und den Bereich des Change Managements dem jeweiligen Modell zuordnen. Je nach Fokussierung der Modelle auf das Netzwerk- oder Systemmanagement wird dabei auch das Change Management unterschiedlich stark innerhalb der Modelle gewichtet. Für einen um-

fassenderen Überblick über die Thematik des Systems Managements sei auf die Ausführungen von [Gulla 1996], [Kauffels 1995] und [Langsford/Moffett 1993] verwiesen.

„Das *OSI Management Framework* [OSI: Open Software Interconnection, Anm./Herv. durch Verf.] ist ein ISO-Standardisierungsdokument ..., welches Richtlinien für die Koordination der Weiterentwicklung bestehender OSI-Management-Standards angibt.“ [Kauffels 1995, S. 270] Dabei unterscheidet das Framework fünf spezifische Management-Funktionsbereiche [vgl. Kauffels 1995, S. 272 ff.; Gulla 1996]:

- Accounting Management (Abrechnungsfunktion): Basis für Benutzungskosten
- Configuration Management (Konfigurationsfunktion): u. a. Installation von Komponenten und deren Namengebung
- Fault Management (Fehlerfunktion): Bearbeitung von Fehlerzuständen
- Performance Management (Leistungsfunktion): Bewertung der Effektivität
- Security Management (Sicherheitsfunktion): umfaßt alle Sicherheitsmechanismen

Da es sich bei diesem Modell in erster Linie um ein Netzwerkmanagement-Framework handelt [vgl. Kauffels 1995, S. 386], das Change Management jedoch eine Teildisziplin des Systemmanagements darstellt, beinhaltet das OSI Management Framework auch keine Erweiterungsplanung, sondern konzentriert sich auf die eigentliche Konfigurationsverwaltung [vgl. Kauffels 1995, S. 279].

Für die *Distributed Management Environment (DME)* der Open Software Foundation (OSF) wurden Technologien führender Hersteller ausgewählt, „die zur Integration und Vereinfachung des Netzwerk-Managements heterogener Systemumgebungen dienen sollen.“ [Kauffels 1995, S. 295] „Die Implementierung des OSF DME schafft einen Rahmen für die Entwicklung von Anwendungen und Servern für die Pflege der verwalteten Objekte.“ [Kauffels 1995, S. 304] Damit kann die DME nicht mit dem zuvor vorgestellten OSI Management Framework verglichen werden, da mit der DME lediglich eine Grundstruktur definiert wird, die nur rudimentäre Anwendungsdienste zur Software-Verteilung und -Installation, zur Druckverwaltung, zum Hostmanagement und zur Lizenzverwaltung beinhaltet. Zur eigentlichen Implementierung gehört dann die Auswahl von Werkzeugen unterschiedlicher Hersteller und deren Integration über das definierte Framework. Dies umfaßt auch ein Werkzeug für das Change Management.

Das *Modell der Meta Group* ist insofern im Ansatz mit dem OSI Management Framework vergleichbar, als auch hier eine Gruppierung in Funktionsbereiche

vorgenommen wird (siehe Abbildung 3.1-1), wobei allerdings eindeutig auf das System- und nicht auf das Netzwerkmanagement abgezielt wird. Das Change Management wird in der schematischen Darstellung nicht explizit erfasst, kann aber der Bestands- und Inventarverwaltung zugeordnet werden, wobei sich bei Änderungen Auswirkungen auf das Konfigurationsmanagement ergeben.

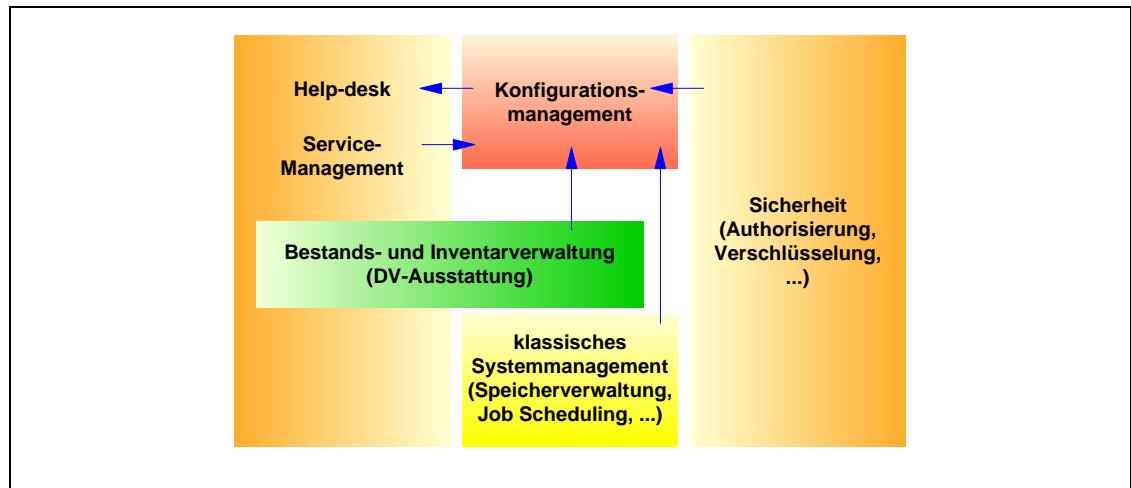


Abb. 3.1-1: Systemmanagement-Modell der Meta Group in schematischer Übersicht
[o. V. 1998a, S. 13]

In der *Literatur* werden ebenfalls Einteilungen in Funktionsbereiche vorgenommen. So führt beispielsweise Kauffels die fünf Aufgabenbereiche Konfigurations-, Fehler-, Leistungs-, Sicherheits- und Abrechnungsmanagement an, wobei er das Change Management unter der deutschen Bezeichnung „Änderungs-Management“ dem Konfigurationsmanagement zurechnet [vgl. Kauffels 1995, S. 93 ff.]. Damit folgt Kauffels dem OSI Management Framework, erweitert dieses jedoch um eine Systemmanagement-Komponente. Dagegen unterscheidet Wessler sieben Funktionen des Client-/Server-Managements: Bestands-, Konfigurations-, Problem-, Performance-, Security-, Accounting-Management und Administration [vgl. Wessler 1995, S. 22]. Somit folgt Wessler der Meta Group in der Trennung von Konfigurations- und Bestandsmanagement in zwei separate Bereiche, wobei das „Änderungsmanagement“ dem Bestandsmanagement zugeordnet wird, was die Einordnung im Modell der Meta Group bestätigt.

Als letztes Modell soll das *SystemView Framework* vorgestellt werden, das „die übergreifende Management-Strategie der IBM“ bildet [Kauffels 1995, S. 382]. Da über „die klassischen Ziele des Netz-Managements hinaus ... die Beherrschung heterogener Netze

angestrebt“ wird [Kauffels 1995, S. 384], kann eine starke Ausrichtung auf das Systemmanagement festgestellt werden. SystemView definiert dabei eine offene Managementarchitektur, die folgende Anwendungsdimensionen unterscheidet [vgl. Kauffels 1995, S. 390; Gulla 1996]:

- Operations Management: Verwaltung der Nutzung von Ressourcen und Systemen im Hinblick auf das Erreichen von Unternehmenszielen
- Problem Management: Problemverfolgung vom Auftreten bis zur Behebung
- Configuration Management: physische und logische Verknüpfungen zwischen Ressourcen
- Performance Management: Sammlung von Leistungsdaten, Tuning, Kapazitätsplanung
- Change Management: Planung, Durchführung und Protokollierung von Änderungen
- Business Management: Bestandsverwaltung, Finanzverwaltungsfunktion, Planung aller Ressourcen eines Unternehmens wie Mitarbeiter, Anlagen und Daten

Dieses Framework unterscheidet damit nicht nur wie Wessler und die Meta Group das Bestands- und Konfigurationsmanagement in zwei unterschiedliche Gruppen, sondern stellt auch das Change Management als eine eigenständige Disziplin dar.

Fazit

Die Vorstellung der verschiedenen Modelle zeigt die unterschiedliche Schwerpunktsetzung allgemein im Systems Management wie auch insbesondere im Bereich des Change Managements. Dabei wird diese Komponente um so stärker betont, je mehr sich das Framework auf das Systemmanagement konzentriert.

Auch muß eine große Interdependenz zwischen den einzelnen Disziplinen der jeweiligen Modelle festgestellt werden, wie sie bereits im Modell der Meta Group angedeutet und von Kauffels bestätigt wurde [vgl. Kauffels 1995, S. 111]. Wimmer verdeutlicht dies in einem Integrationsschaubild (siehe Abbildung 3.1-2), in dem das Change Management im Fokus der Betrachtung steht [vgl. Wimmer 1997, S. 45].

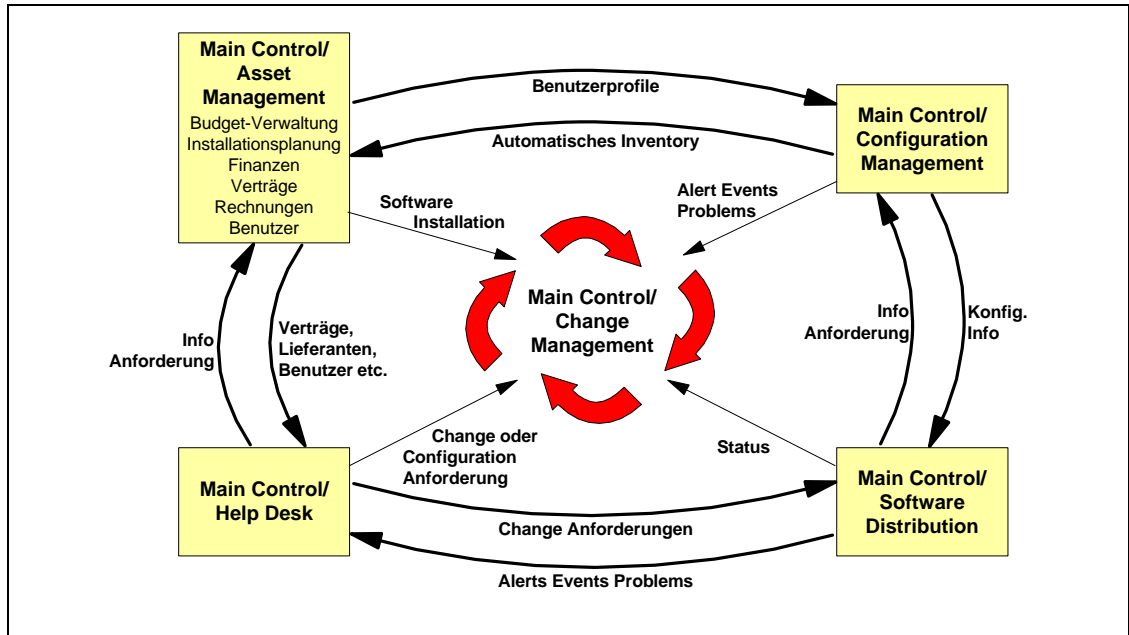


Abb. 3.1-2: Interdependenzen zwischen den Framework-Disziplinen

Zusätzlich fällt dem Change Management im Rahmen der Verzahnung in die Geschäftsprozesse eine zentrale Rolle zu (siehe Abbildung 3.1-3). In Zukunft soll das Systemmanagement mit der Geschäftsprozessautomation zur Unternehmenssystemautomation verschmelzen [vgl. Born 1998, S. 13].

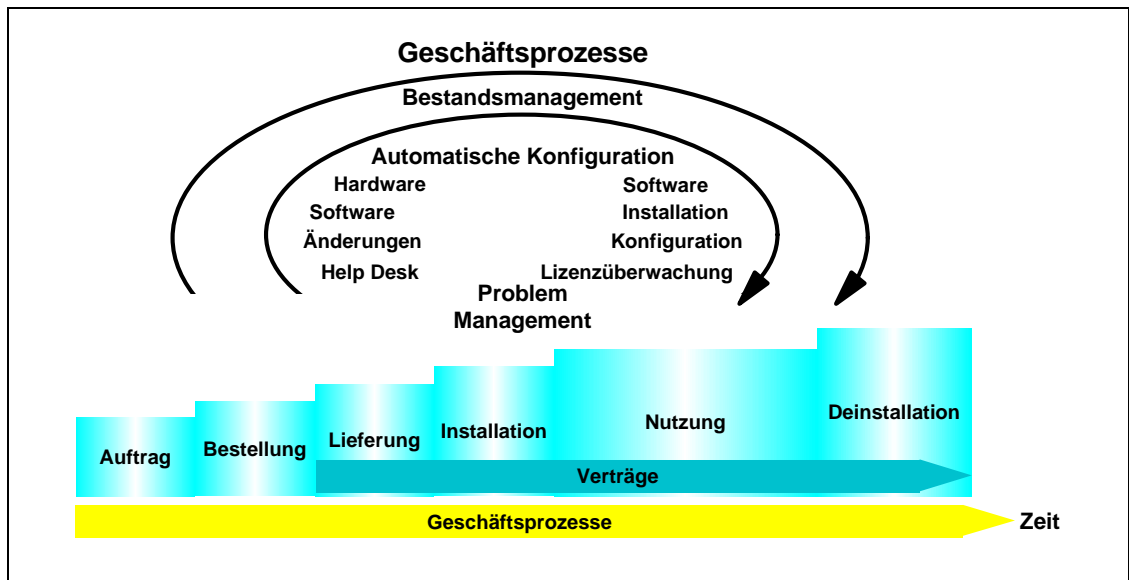


Abb. 3.1-3: Verzahnung technischer und betriebswirtschaftlicher Anwendungen [vgl. Wimmer 1997, S. 44]

3.2 Anforderungen an Workflow Management-Systeme

Im Rahmen dieses Abschnitts sollen einige, für das weitere Verständnis der Arbeit erforderliche konzeptionelle Anforderungen an Workflow Management-Systeme vorgestellt werden.

Workflow-Referenzmodell der Workflow Management Coalition

Großer Einfluß auf die Konzeptionierung von Workflow Management-Systemen geht von der Workflow Management Coalition (WfMC) aus. Die WfMC „ist ein internationaler, freiwilliger Zusammenschluß von Softwareherstellern, Workflow-Absatzermittlern, Benutzern und Analysten mit über 100 Mitgliedern. Das Ziel der WfMC ist die Förderung des Workflow-Einsatzes durch Verabschiedung von Standards für Begriffe, Interoperabilität und Verbindung zwischen den Vorgangsteuerungssystemen.“ [Weiß/Krcmar 1996, S. 505 f.]

Als wichtigstes Ergebnis der Arbeit dieses Gremiums kann das Workflow-Referenzmodell angesehen werden (siehe Abbildung 3.2-1). Dieses beinhaltet die Beschreibung von Begriffen, Werkzeugen und Schnittstellen und setzt damit einen Standard für die Entwicklung von Workflow Management-Systemen.

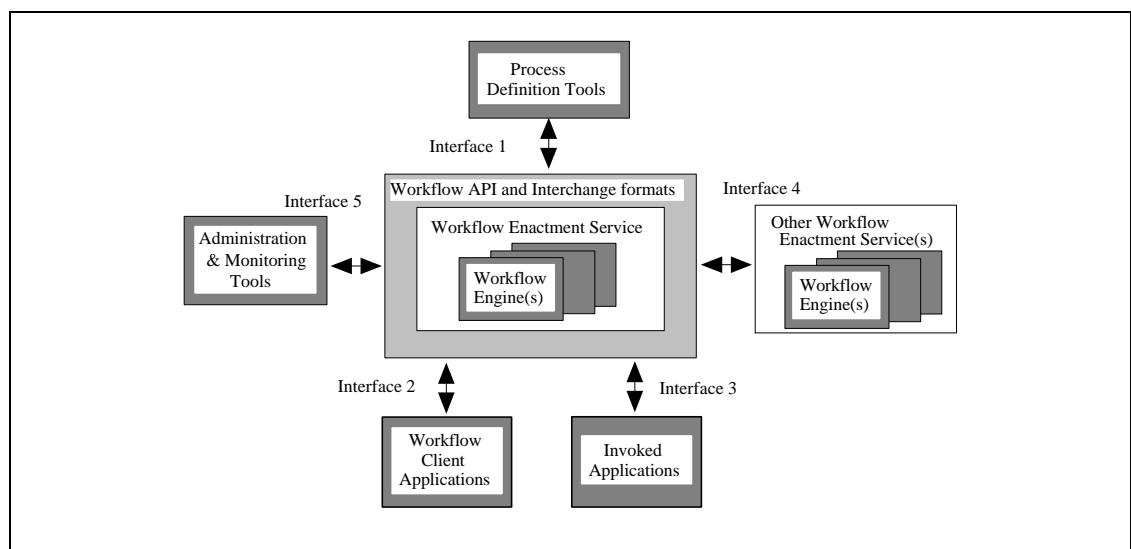


Abb. 3.2-1: Workflow-Referenzmodell - Komponenten und Schnittstellen

[WfMC 1994, S. 20]

Mit Hilfe der *Process Definition Tools* (dt. Vorgangsentwurfswerkzeuge) können Vorgangstypen entworfen, beschrieben und simuliert werden. Über eine Workflow API (API: Application Programming Interface, dt. Anwendungsprogrammchnittstelle) oder

ein Austauschformat kann die Workflow-Klasse dann im *Workflow Enactment Service* (dt. Vorgangsausführungsdienst) abgelegt werden, in dem eine oder mehrere Workflow Engines die Anordnung der Tätigkeiten zur Laufzeit übernehmen.

Die *Workflow Client Applications* (dt. Vorgangsendanwenderapplikationen) stellen dem Anwender die zu bearbeitenden Tätigkeiten zur Verfügung. Zusätzlich binden sie die *Invoked Applications* (dt. aufgerufene Applikationen) ein.

Die *Administration und Monitoring Tools* (dt. Administrations- und Überwachungswerkzeuge) ermöglichen die Auswertung und Verfolgung der Vorgangsexemplare, während das Interface 4 die Interoperabilität mit anderen Workflow Management-Systemen sicherstellen soll [vgl. Weiß/Krcmar 1996, S. 506 f.].

Der Ansatz des Workflow Life Cycles

Planung, Ausführung und Kontrolle eines Arbeitsprozesses können als die Aufgaben eines Workflow Management-Systems angesehen werden [vgl. Elgass/Krcmar 1994, S. 70]. Deshalb schlagen sowohl Heilmann [Heilmann 1994, S. 13 f.] als auch Hilpert [Hilpert 1994, S. 2 ff.] einen kontinuierlichen Workflow Management-Zyklus vor (siehe Abbildung 3.2-2). Dieser Zyklus „kann mehrfach durchlaufen werden, bis ein hinreichend optimierter Vorgangstyp vorliegt.“ [Heilmann 1994, S. 13]

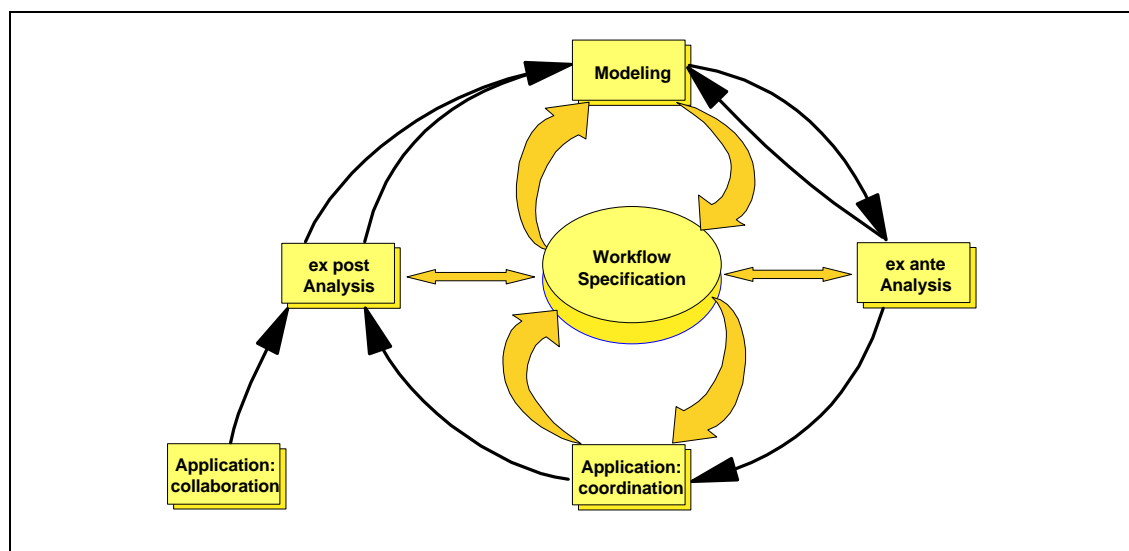


Abb. 3.2-2: Workflow Life Cycle [Hilpert 1994, S. 3]

Aus dem Zyklus lassen sich auch die Komponenten eines Workflow Management-Systems ableiten. Ausgehend von einer konkreten Situation der Zusammenarbeit (*Application: collaboration*) wird diese nachträglich analysiert (*ex post Analysis*).

Anschließend wird in einem Modellierungssystem (*Modeling*) die Workflow-Klasse definiert und in einer Spezifikationsdatenbank (*Workflow Specification*) abgelegt. Diese Datenbank dient als zentrales Daten-Repository über den gesamten Lebenszyklus.

Die modellierten Vorgangstypen können nun auf Basis festgelegter Parameter in einem Simulationssystem getestet werden (*ex ante Analysis*), bevor sie operativ im Ausführungssystem instanziiert werden (*Application: coordination*). Werden in der Simulation Probleme festgestellt, so können die Workflow-Klassen ggf. verändert und neu simuliert werden.

Die Workflow-Instanzen können zu einem beliebigen Zeitraum in einem Vorgangsinformationssystem kontrolliert werden (*ex post Analysis*), was auch als Tracking oder Monitoring bezeichnet wird. Wird dabei ein Verbesserungspotential aufgezeigt, so beginnt der Zyklus erneut.

Das Workflow-Kontinuum

Je nach Strukturierungsgrad der Geschäftsprozesse ergeben sich verschiedene Anforderungen an Flexibilität und Automatisierung der Workflow-Strukturen. Dieser Tatsache kann durch ein Workflow-Kontinuum Rechnung getragen werden, das die Anforderungen mittels vier Workflow-Klassen abdeckt (siehe Abbildung 3.2-3).

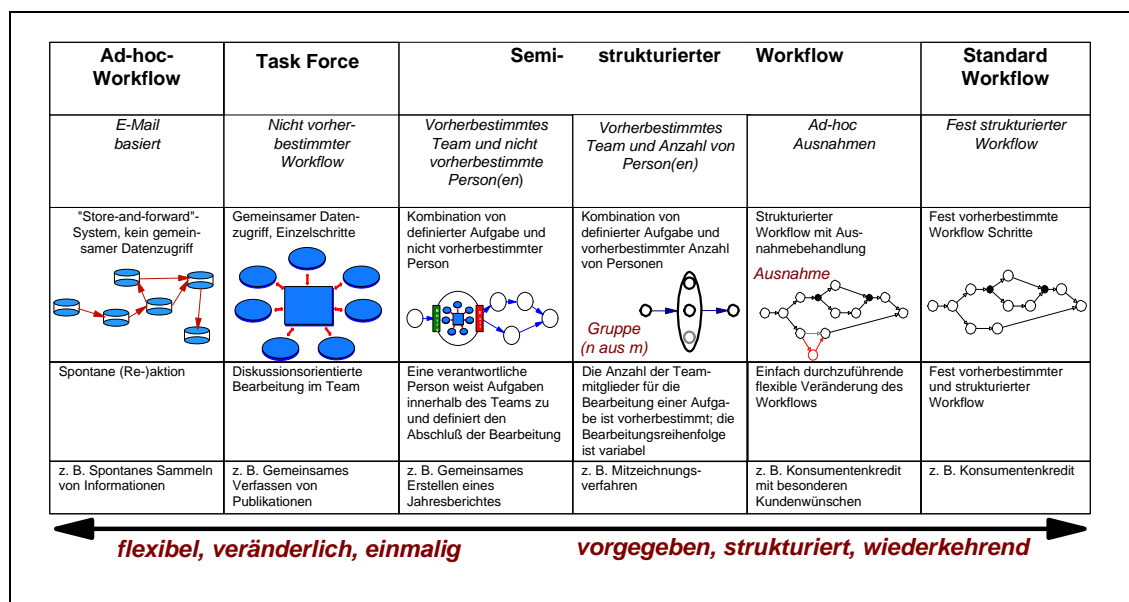


Abb. 3.2-3: Workflow-Kontinuum [vgl. Nastansky/Hilpert 1994, S. 2]

„Einmalige und kurzlebige Prozesse, die in ihrem Komplexitätsgrad stark variieren und generell schlecht strukturierbar sind, werden dem *Ad-hoc-Workflow* zugeordnet.“
[Weiß/Krcmar 1996, S. 509]

Die Klasse der *Task Force Workflows* unterscheidet sich von den E-Mail-basierten Ad-hoc-Workflows dahingehend, daß autonome Arbeitsgruppen diskussionsorientiert arbeiten, da sie auf eine gemeinsame Datenbasis zugreifen können. Dies schränkt die Flexibilität insofern ein, daß nur Gruppenmitglieder an der Bearbeitung teilnehmen können.

Standard Workflows sind „gleichartig wiederholbare Prozesse mit Routineaufgaben in verschiedenen Varianten“ [Weiß/Krcmar 1996, S. 510]. Sie sind vorherbestimmt und fest strukturiert.

Zwischen den Klassen der Standard und Task Force Workflows können verschiedene Gruppen von *semi-strukturierten Workflows* festgestellt werden. Dies beginnt mit einfachen Ad-hoc-Ausnahmen von der fest vorgegebenen Struktur und reicht über „controlled team tasks“ [vgl. Nastansky/Hilpert 1994, S. 2] mit einer vorherbestimmten Gruppe und einer vorherbestimmten Anzahl an Bearbeitern bis zur Einbindung von autonomen Arbeitsgruppen als je eine Aufgabe im Vorgangstyp.

3.3 Grundsätze der Software-Ergonomie

„Wenn die *Ergonomie* die menschengerechte Gestaltung der Arbeit bedeutet, ist die *Software-Ergonomie* die menschengerechte Gestaltung der Interaktion zwischen Mensch und Computer.“ [Lauter 1987, S. 1]

Die Ziele der Software-Ergonomie lassen sich dabei wie folgt angeben [vgl. Lauter 1987, S. 1]:

- optimale Anpassung von Mensch und dialogorientiertem Computersystem
- Entwicklung eines Mensch-orientierten Computersystems
- hohe Benutzerakzeptanz des Systems
- Förderung von Motivation und Kreativität des Anwenders
- bestmögliche Arbeitsergebnisse unter menschengerechten Bedingungen erzielen
- großen Handlungsspielraum und -kompetenz für den Anwender bereitstellen

Der Stellenwert der Informationspräsentation durch die Mensch-Maschine-Schnittstelle kann dem Umstand entnommen werden, daß heute mehr als 50% des Programmcodes zu diesem Zweck verwandt werden [Reiterer 1994, S. 1]. Dieser Aufwand ist verständlich, ist doch ein „Software-Produkt .. nur so gut, wie seine Benutzer damit umgehen können.“ [Lauter 1987, S. 1]

Die Beziehungen zwischen Aufgabe, Benutzer und Computer lassen sich dabei in ein Modell fassen, das in Abbildung 3.3-1 dargestellt und für viele Richtlinien und Normen als Basis herangezogen wird. Beispielhaft sind hier die DIN/EN 29241 Teil 10, EVADIS der Gesellschaft für Mathematik und Datenverarbeitung (GMD), die ISO 9241 Teil 14-17, die EU-Bildschirmrichtlinie (90/270 EWG) und die VDI Richtlinie 5005 zu nennen [vgl. Keil-Slawik 1995, S. 8].

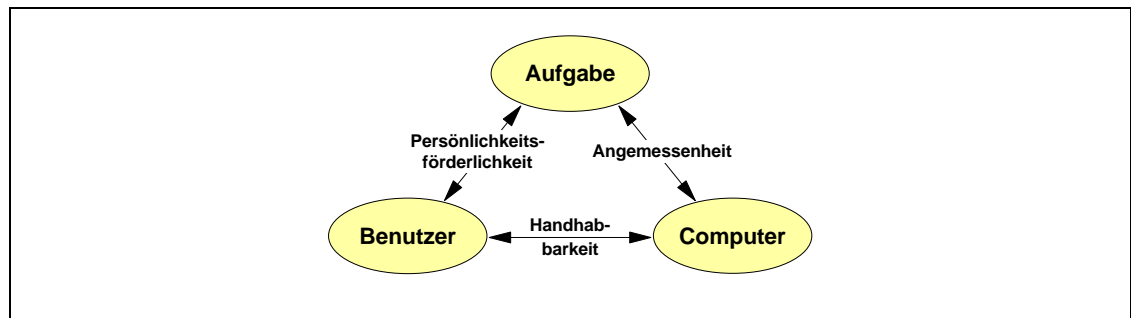


Abb. 3.3-1: ABC-Modell der Software-Ergonomie [vgl. Keil-Slawik 1995, S. 8]

Exemplarisch sei an dieser Stelle die DIN/EN 29241 Teil 10 näher betrachtet, da sie einen besonders ganzheitlichen Ansatz auf Basis der weltweiten Norm ISO 9241 darstellt und ihre Anwendung als europäische Norm maßgeblich ist.

Die DIN/EN 29241 Teil 10 „enthält ergonomische Grundsätze, die als grundlegende Begriffe formuliert sind, d. h. sie werden ohne Bezug auf die Arbeitssituation, Anwendungen, Umgebungen und Technik wiedergegeben.“ [DIN/EN 29241, Anwendungsbereich] Dieser sehr breite Ansatz bedeutet, daß die Grundsätze frei auf die jeweilige Situation interpretiert werden können, wobei die aufgelisteten Empfehlungen und Beispiele der einzelnen Grundsätze einen insgesamt positiven Ansatz verglichen mit den überwiegenden Negativkriterien der traditionellen Software-Ergonomie darstellen [vgl. Keil-Slawik 1995, S. 10]. Daher können die Prinzipien auch „bei der Gestaltung und Bewertung von Dialogsystemen angewandt werden.“ [DIN/EN 29241, Anwendungsbereich] In Kapitel 6.4.4 soll aufgrund dieser Eignung der implementierte Prototyp im Hinblick auf seine Benutzerführung und -interaktion bewertet werden.

Im Folgenden sollen die in der DIN/EN 29241 Teil 10 definierten Grundsätze im Kontext von Workflow Management-Systemen eingeordnet werden:

- *Aufgabenangemessenheit:* „Ein Dialog ist in dem Maße aufgabenangemessen, wie er den Benutzer unterstützt, seine Arbeitsaufgabe effizient und effektiv zu erledigen.“ [DIN/EN 29241, Aufgabenangemessenheit]

Dieser Grundsatz muß als zentrales Gestaltungsziel eines jeden Workflow Management-Systems angesehen werden. Dabei darf aber nicht nur die Unterstützung der eigentlichen Aufgabe betrachtet werden. Vielmehr ist auch ein einfaches Navigationskonzept zu berücksichtigen, da dieses als zentraler Bestandteil des Arbeitsumfelds maßgeblich die Erledigung der Arbeitsaufgabe beeinflusst.

- *Selbstbeschreibungsfähigkeit:* „Ein Dialog ist in dem Maße selbstbeschreibungsfähig, wie jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer erklärt wird, wenn er die entsprechenden Informationen verlangt.“ [DIN/EN 29241, Selbstbeschreibungsfähigkeit]

Dieses Kriterium stellt sicher, daß sich der Anwender mit der eigentlichen Aufgabe auseinandersetzen kann und nicht durch Bedienungsfragen zum System in seinem Arbeitsfluß gestört wird. Damit wird der Effizienzgedanke der Aufgabenangemessenheit unterstützt.

- *Steuerbarkeit:* „Ein Dialog ist in dem Maße steuerbar, wie der Benutzer in der Lage ist, den gesamten Dialogablauf bis zu dem Punkt, an dem das Ziel erreicht ist, zu beeinflussen.“ [DIN/EN 29241, Steuerbarkeit]

Damit zielt dieser Grundsatz auf allgemeine Anforderungen ab wie z. B. die Dialoggeschwindigkeit, die angezeigte Informationsmenge, die Möglichkeit der Unterbrechung mit erneuter Arbeitsaufnahme an diesem Punkt oder die Rücknahme des letzten Dialogschrittes.

- *Erwartungskonformität:* „Ein Dialog ist in dem Maße erwartungskonform, wie er den Kenntnissen aus bisherigen Arbeitsabläufen, der Ausbildung und der Erfahrung des Benutzers sowie den allgemein anerkannten Übereinkünften entspricht.“ [DIN/EN 29241, Erwartungskonformität]

Wichtig sind im Hinblick auf Workflow Management-Systeme die Unterstützung der gewohnten Arbeitsabläufe sowie die einheitliche Darstellung der Sachverhalte über verschiedene Aufgaben hinweg, um gleiche Bearbeiter mehrerer Aufgaben möglichst effektiv zu unterstützen.

- *Fehlerrobustheit:* „Ein Dialog ist in dem Maße fehlerrobust, wie das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben mit minimalem oder ohne Korrekturaufwand erreicht wird.“ [DIN/EN 29241, Fehlerrobustheit]

Da die Aufgabenweiterleitung oft automatisiert auf Basis der eingegebenen Daten vorgenommen wird, müssen offensichtlich fehlerhafte Eingaben abgefangen werden. Der Benutzer sollte dabei dahingehend unterstützt werden, daß er den Fehler leicht finden und in Zukunft vermeiden kann.

- *Individualisierbarkeit:* „Ein Dialog ist in dem Maße individualisierbar, wie er Anpassungen an individuelle Benutzerbelange und Benutzerfähigkeiten im Hinblick auf eine gegebene Arbeitsaufgabe zuläßt.“ [DIN/EN 29241, Individualisierbarkeit]

Diese Anforderung umfaßt die in jedem System zu beachtende Berücksichtigung von Leistungsminderungen des Benutzers, die z. B. eine unterschiedliche Farbdarstellung oder eine vergrößerte Schrift notwendig machen. Aber auch die Anpassung des Erläuterungsumfangs von Hilfetexten und Fehlermeldungen sowie die für Workflow Management-Systeme relevante Automatisierung eigener, wiederkehrender Aufgaben fallen in diesen Grundsatz.

- *Lernförderlichkeit:* „Ein Dialog ist in dem Maße lernförderlich, wie er dem Benutzer während des Erlernens Unterstützung und Anleitung gibt.“ [DIN/EN 29241, Lernförderlichkeit]

Im Hinblick auf Workflow Management-Systeme muß darauf geachtet werden, daß oft auch weniger erfahrene Benutzer im Rahmen gelegentlicher Tätigkeiten am System teilnehmen. Diese Benutzer können oftmals nicht geschult werden und erhalten daher nur eine kurze schriftliche Anleitung. Die restliche Schulung erfolgt direkt in der Praxis während der Arbeit im System.

Zusammenfassend kann eine hohe Relevanz der DIN/EN 29241 Teil 10 auch im Bereich der Workflow Management-Systeme festgestellt werden, da sie in all ihren Grundsätzen positive Gestaltungsansätze für die elektronische Umsetzung von Geschäftsprozessen bietet.

4 Konkretes Szenario der Deutschen Bundesbank

In diesem Kapitel sollen die Rahmenbedingungen für die prototypische Implementierung vorgestellt werden. Als erster Schritt wird die Eingliederung des Change Managements in den Systems Management-Ansatz der Deutschen Bundesbank dargestellt. Anschließend wird das Projektumfeld näher charakterisiert, bevor der von der IBM Unternehmensberatung erarbeitete theoretische Rahmen für die Projektimplementierung in den wichtigsten Punkten vorgestellt wird.

4.1 Systems Management-Ansatz

Die Deutsche Bundesbank hat sechs Prozeßgruppen als Bestandteile des Systems Managements identifiziert [vgl. IBM 1998b, S. 11]:

- *Problem Management*: alle Aktivitäten zur Erfassung, Bearbeitung, Lösung und Dokumentation von Problemen im IT-Bereich
- *Operations Management*: sämtliche Aktivitäten zur Planung, Steuerung und Überwachung des Betriebs der IT-Infrastruktur
- *Change Management*: alle Aktivitäten, die Änderungen in der IT-Infrastruktur hervorrufen
- *Performance Management*: sämtliche Aktivitäten zur Planung, Kontrolle, Analyse und Anpassung von Performance Parametern und Kapazitäten zur System- und Betriebsablaufoptimierung
- *Configurations Management*: alle Aktivitäten zur Planung, Erfassung und Bereitstellung von Konfigurationsdaten
- *Business Management*: vorwiegend betriebswirtschaftliche Aktivitäten, die zur Erfassung und Verwaltung der IT-Infrastruktur notwendig sind

Damit folgt der Systems Management-Ansatz dem in Kapitel 3.1 vorgestellten SystemView-Framework der Firma IBM.

Im Vorfeld des im weiteren Verlauf der Arbeit näher beschriebenen Projekts zur Realisierung des Change Managements mit Hilfe einer toolgestützten Lösung wurde bereits ein Problem Management-Prozeß definiert und ein entsprechendes Produkt eingeführt. Desweiteren wurde eine Studie zur Definition einer Prozeßgrobstruktur im Change und Configurations Management durchgeführt. Die praktische Umsetzung des Change

Managements stellt somit die zweite Ausbaustufe in der Realisierung der Systems Management-Prozesse dar.

Als dritte Ausbaustufe soll in naher Zukunft das Configurations Management angegangen werden, da hier eine besonders enge Verzahnung mit dem Problem und dem Change Management festgestellt werden muß. So ist es äußerst hilfreich, bei Problem-meldungen die Konfigurationsdaten des PCs zur Verfügung zu haben, um mögliche Fehlerquellen einzugrenzen. Bei Change-Anträgen kann wesentlich einfacher die Ver-träglichkeit der beantragten Hard- und Software mit der vorhandenen Konfiguration überprüft werden.

Die dann noch ausstehenden drei Prozeßgruppen des Operations, Performance und Business Managements sollen im Anschluß an das Configurations Management umge-setzt werden.

4.2 Projektumfeld

Ziel dieses Abschnitts ist es, das Projektumfeld vorzustellen, um insbesondere auf Basis der Projektziele die Umsetzung des Projekts beurteilen zu können.

4.2.1 Projekthintergrund/Motivation

Den gesteigerten Anforderungen an die Informationstechnik in Form der geplanten Währungsunion, der Jahr 2000-Problematik sowie neuer Techniken im Client-/Server-Umfeld versuchte die Deutsche Bundesbank in den letzten Jahren durch den Einsatz moderner Technologie und verbesserter Infrastruktur zu begegnen. „Durch die neuen Techniken und gesteigerten Anforderungen werden aber auch dringend neue, verbes-serte Abläufe (Prozesse), Strukturen (Organisation) und unterstützende Technologien (Tools) benötigt.“ [IBM 1998b, S. 4]

Ein zusätzlich motivierender Faktor ist die zunehmende Serviceausrichtung der DV-Abteilung, die allen Abteilungen als interner Dienstleister ein erhöhtes Service-angebot zur Verfügung stellen möchte. Die Richtlinien und Prinzipien des Systems Managements wurden entsprechend dieser Serviceziele ausgerichtet (vgl. Abbildung 4.2.1-1).

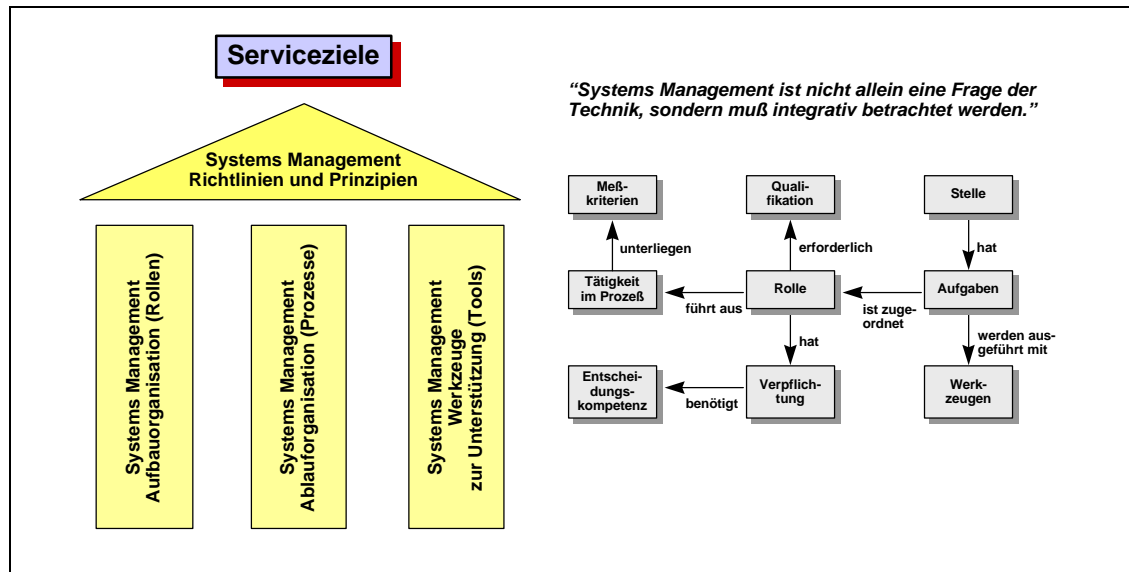


Abb. 4.2.1-1: *Integratives Systems Management mit klarer Serviceausrichtung [vgl. IBM 1998b, S. 12]*

Um die gewünschten Serviceziele zu erreichen und den gesteigerten Anforderungen gerecht zu werden, wurde daher „mit der Verbesserung des Managements des IT-Betriebs (Systems Management) begonnen“ [IBM 1998b, S. 4]. Dieser Prozeß wird mit der Implementierung des Change Management-Prototypen fortgesetzt.

4.2.2 Projektziele

„Ziel des Projektes ... ist die Implementierung des bereits definierten Change-Prozesses und die weitgehende Automatisierung des Prozeßablaufes mit Hilfe eines Lotus Notes-basierten Workflow-Systems.“ [IBM 1997a, S. 4]

Dies bedeutet, daß die Ergebnisse des Vorprojekts, in dem eine grobe Prozeßstruktur für das Change Management festgelegt wurde, insgesamt verfeinert (u. a. Feindesign der Aktivitätenabfolge, Erstellung eines konzeptionellen Datenmodells, Vertiefung des Rollenmodells) und konkret implementiert werden sollten (Modellierung und Umsetzung des Workflows).

Der Schwerpunkt der Umsetzung lag dabei auf der Unterstützung des anstehenden Windows NT-Rollouts, wobei aber auf eine einfache Erweiterbarkeit für die zukünftige Nutzung im UNIX- und Host-Umfeld zu achten war. Zudem sollte bei Projektbeginn lediglich der DV-interne Ablauf abgebildet werden, jedoch wurde im Verlauf des Projekts deutlich, daß eine elektronische Einbindung aller weiterer am Prozeßablauf beteiligten Abteilungen äußerst sinnvoll ist, und dies von diesen auch gewünscht wurde.

Als Prämissen wurden für das Projekt formuliert [vgl. IBM 1998b, S. 5]:

- schnelle Umsetzung
- Verringerung von Transport- und Liegezeiten
- erhöhte Transparenz (Statusabfragen sollten jederzeit möglich sein)
- modulares Konzept zur Gewährleistung zukünftiger Erweiterbarkeit
- Unterstützung der wichtigsten Geschäftsvorfälle

Als wichtigste Geschäftsvorfälle bei der NT-Einführung wurden dabei die *Einrichtung eines neuen Arbeitsplatzes*, die *Hardware-Neuanschaffung*, das *Hardware-Upgrade*, die *Hardware-Reparatur*, die *Software-Neueinführung* und die *Software-Fehlerbereinigung* identifiziert. Zusätzlich wurden in der Implementierung noch die Geschäftsvorfälle *Standard-Hardware* und *Standard-Software* berücksichtigt [vgl. IBM 1998b, S. 22].

4.2.3 Projektverlauf

Das Projekt wurde von vier Consultants der IBM Unternehmensberatung in enger Zusammenarbeit mit zwei Consultants der PAVONE Informationssysteme durchgeführt. Ein Teil der Implementierungsarbeit konnte im PAVONE-Hauptsitz in Paderborn vorgenommen werden, was eine enge Zusammenarbeit mit dem Entwicklungsteam des eingesetzten Workflow Management-Systems ermöglichte, so daß mehrere Funktionen aus dem Projekt in das Standardprodukt eingeflossen sind. Einen Überblick über den zeitlichen Verlauf des Projekts gibt die Abbildung 4.2.3-1, wobei allein der Implementierungsaufwand mit ca. 1.200 Mannstunden angesetzt werden muß.

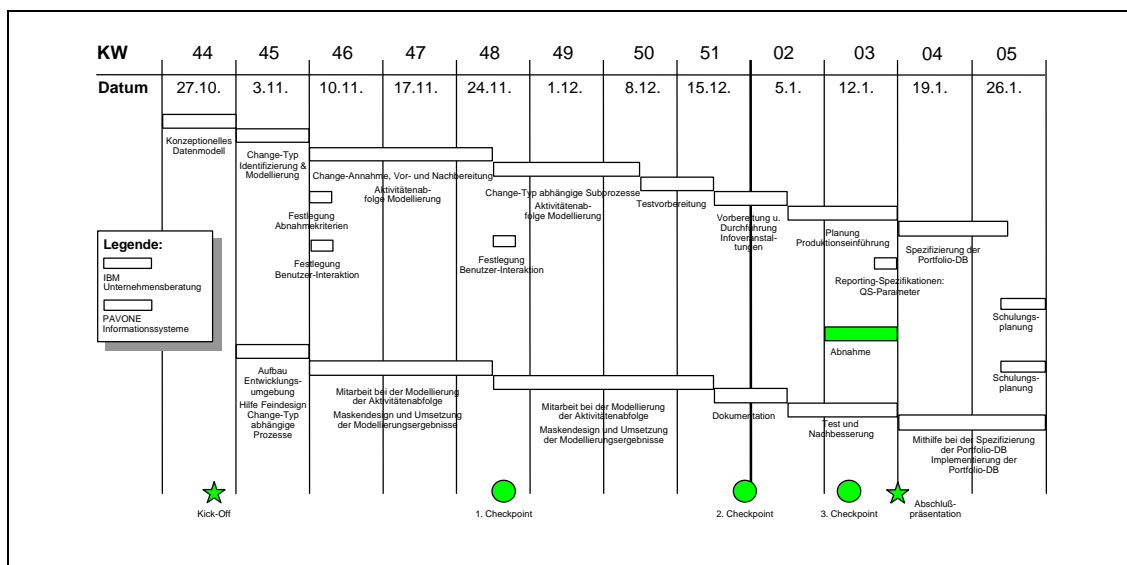


Abb. 4.2.3-1: Zeitlicher Projektverlauf [vgl. IBM 1998b, S. 6]

4.3 Umsetzung des Change Management-Prozesses

Dieser Abschnitt soll den theoretischen Rahmen für die konkrete Implementierung des Prototypen vorstellen. Ermittelt wurde das Konzept zur Realisierung des Prototypen von der IBM Unternehmensberatung, wobei die Firma PAVONE Informationssysteme zu Fragen der technischen Realisierbarkeit beratend mitwirkte.

4.3.1 Konzept der Geschäftsvorfälle und Change-Typen

Geschäftsvorfälle sind alltägliche Situationen wie z. B. das Einrichten eines neuen Arbeitsplatzes oder die Neuanschaffung von Hardware. Um nicht für alle denkbaren Geschäftsvorfälle je einen eigenständigen Prozeß definieren zu müssen, die sich dann in weiten Teilen überdecken würden, wurde ein Klassifikationsraster gebildet, das am Ende in die sogenannten Change-Typen mündet (siehe Abbildung 4.3.1-1). Damit sind „Change-Typen .. kombinierbare Bausteine, mit denen sämtliche Geschäftsvorfälle im Change Management abgebildet werden können.“ [IBM 1997a, S. 15] „Durch die Trennung von Geschäftsvorfällen und Change-Typen ist das Change Management modular aufgebaut und somit zukünftig leicht erweiterbar.“ [IBM 1997a, S. 16]

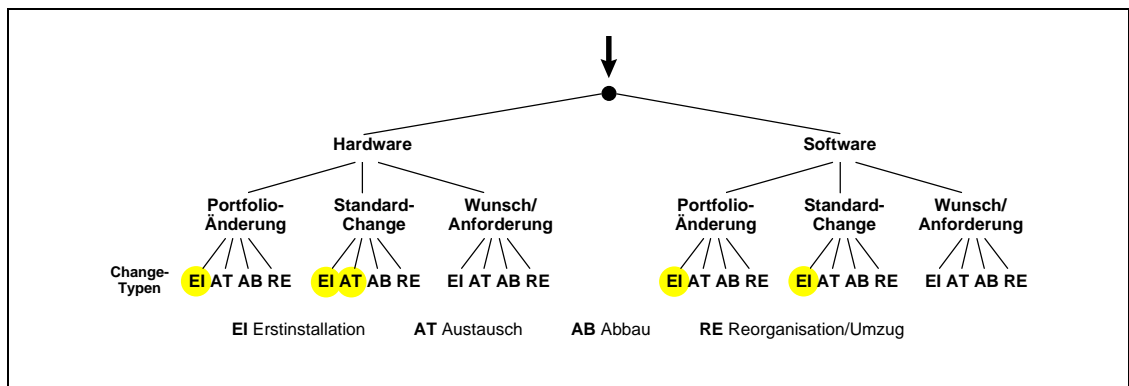


Abb. 4.3.1-1: Klassifikationsraster zur Change-Typen-Ermittlung [IBM 1997a, S. 15]

Zur Klassifikation wird zunächst zwischen Hard- und Software unterschieden. Der zweite Schritt differenziert zwischen einer Portfolio-Änderung (erstmalige Einführung eines Produkts, was vorbereitende und qualitätssichernde Maßnahmen voraussetzt), einem Standard-Change (Hard- oder Software, die zum wiederholten Male eingesetzt wird) und einem Wunsch (Anforderung einer völlig neuen, noch nicht implementierten, projektierten bzw. freigegebenen Komponente). In der dritten Klassifikationsebene wird schließlich zwischen Erstinstantion, Austausch, Abbau und Reorganisation/Umzug für den jeweiligen Anwender unterschieden.

Die fünf in Abbildung 4.3.1-1 markierten Change-Typen reichen dabei aus, um sowohl die sechs ursprünglich geplanten als auch die zwei später im Projektverlauf hinzugekommenen Geschäftsvorfälle abzudecken (vgl. Kapitel 4.2.2). Eine konkrete Zuordnung der implementierten Geschäftsvorfälle zu den benötigten Change-Typen kann dabei der Abbildung 4.3.1-2 entnommen werden.

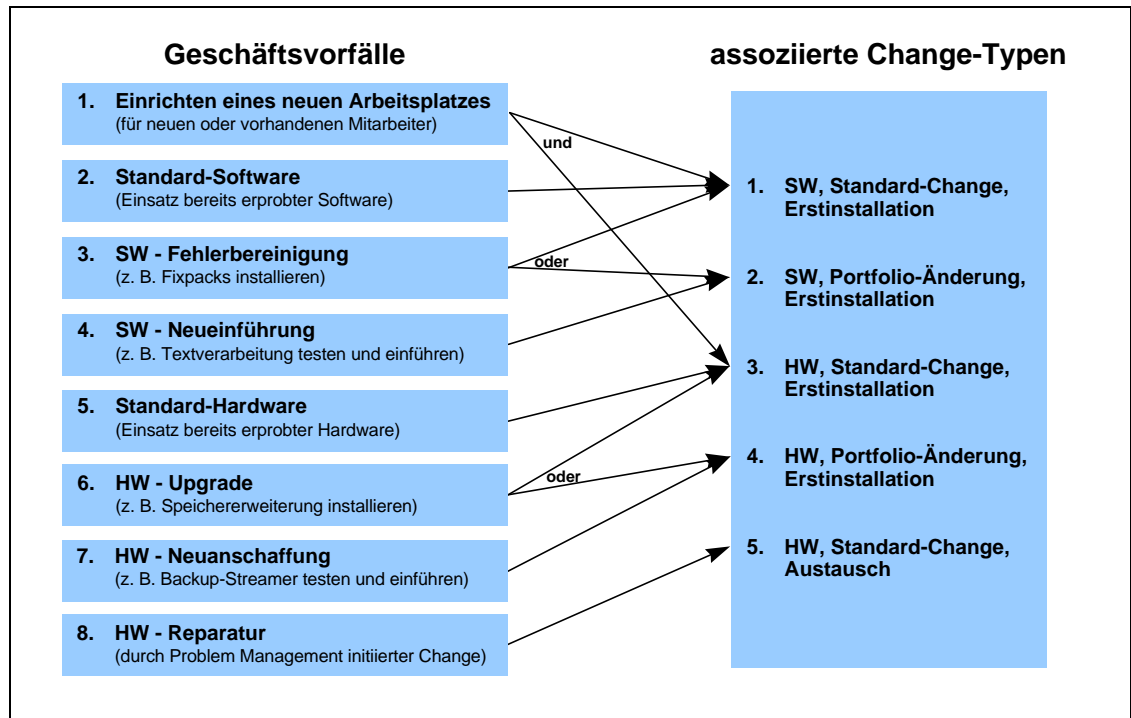


Abb. 4.3.1-2: Abbildung der ausgewählten Geschäftsvorfälle auf die Change-Typen
[vgl. IBM 1997a, S. 16]

4.3.2 Prozeßstruktur

Jeder Geschäftsvorfall im Change Management „kann in einen administrativen und in einen technischen Teil untergliedert werden. Während im technischen Teil die tatsächlichen Änderungen im IT-Umfeld ... vorgenommen werden, werden im administrativen Teil alle durchzuführenden Änderungen vorbereitet, die Durchführung unterstützt und dokumentiert.“ [IBM 1998b, S. 23] Der Change Administration Prozeß wurde daher in vier Teile untergliedert, die in Abbildung 4.3.2-1 dargestellt werden.

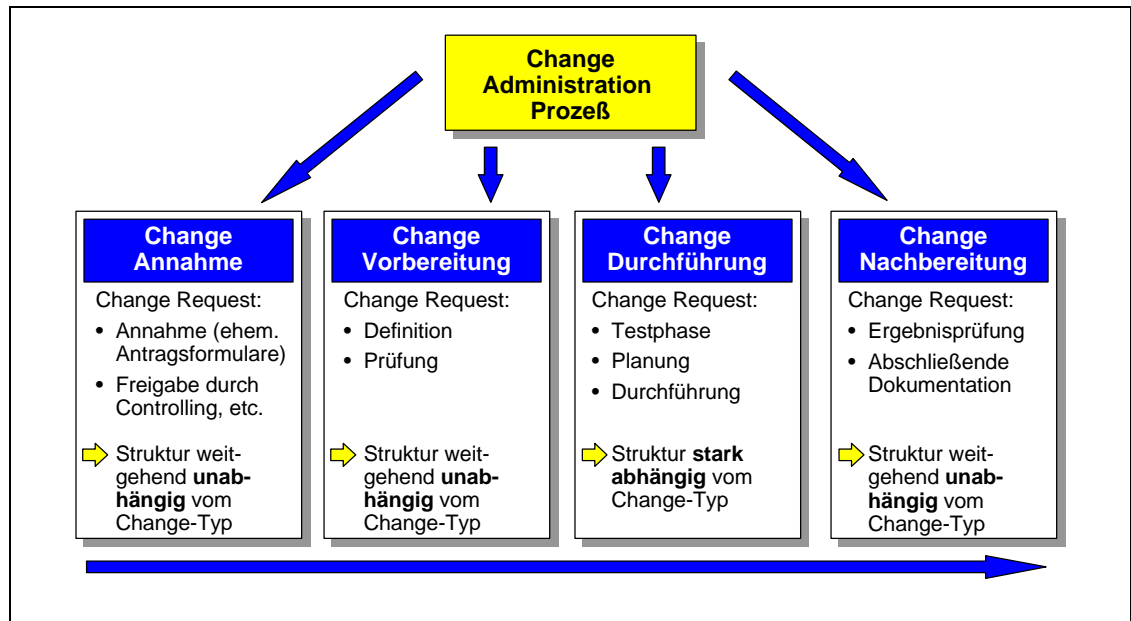


Abb. 4.3.2-1: Grobgliederung des Change Administration Prozesses [IBM 1997b, S. 2]

Somit ergeben sich „drei stark strukturierte, geringen Veränderungen unterworfenene Subprozesse“ (Change Annahme, Vorbereitung und Nachbereitung) sowie in der Change Durchführung eine „variable Anzahl von unterschiedlichst stark strukturierter, häufigen und oft großen Veränderungen unterworfenene und vom Change-Typ abhängige Subprozesse“ [IBM 1997b, S. 3].

Faßt man nun die Change Annahme und Vorbereitung zusammen, so ergibt sich die Struktur des Change Administration Prozesses, wie sie in Abbildung 4.3.2-2 dargestellt wird. Nach dem Durchlaufen der Change Annahme und Vorbereitung entscheidet sich auf Basis des Geschäftsvorfalles, welche Change-Typen in der Change Durchführung durchlaufen werden. Anschließend findet eine Prozeßresynchronisation statt, bevor abschließend die Change Nachbereitung durchlaufen wird.

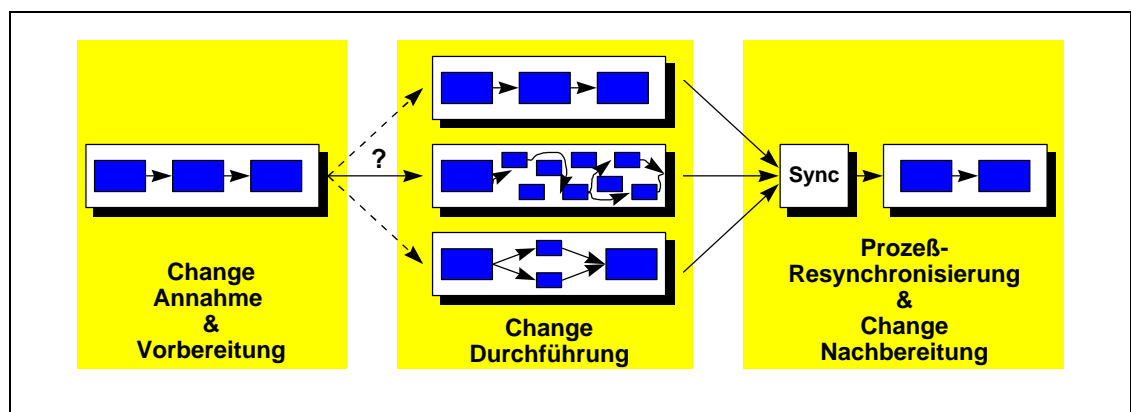


Abb. 4.3.2-2: Struktur des Change Administration Prozesses [IBM 1997b, S. 3]

4.3.3 Rollen und Aufgabenbereiche

„Eine Rolle kann von einer Person - aber auch von einer Gruppe von Personen - bekleidet werden und definiert für einen festgelegten Arbeitsbereich Aufgaben und Verantwortungen.“ [IBM 1998b, S. 16]

Damit kann eine Rolle als zusätzliche Abstraktionsebene bei der Zuordnung von Bearbeitern zu Aufgaben angesehen werden. Den Aufgaben der Workflow-Klasse kann eine funktionelle Rolle mit einem eingängigen Namen zugeordnet werden, die erst in einem zweiten Schritt zu konkreten Personen aufgelöst wird. Dabei können Personen auch mehrere funktionelle Rollen erfüllen, jedoch wird die Prozeßbeschreibung durch die Zuordnung des funktionellen Rollennamens wesentlich einfacher lesbar. Abbildung 4.3.3-1 gibt einen Überblick über die im Projekt benötigten Rollen.

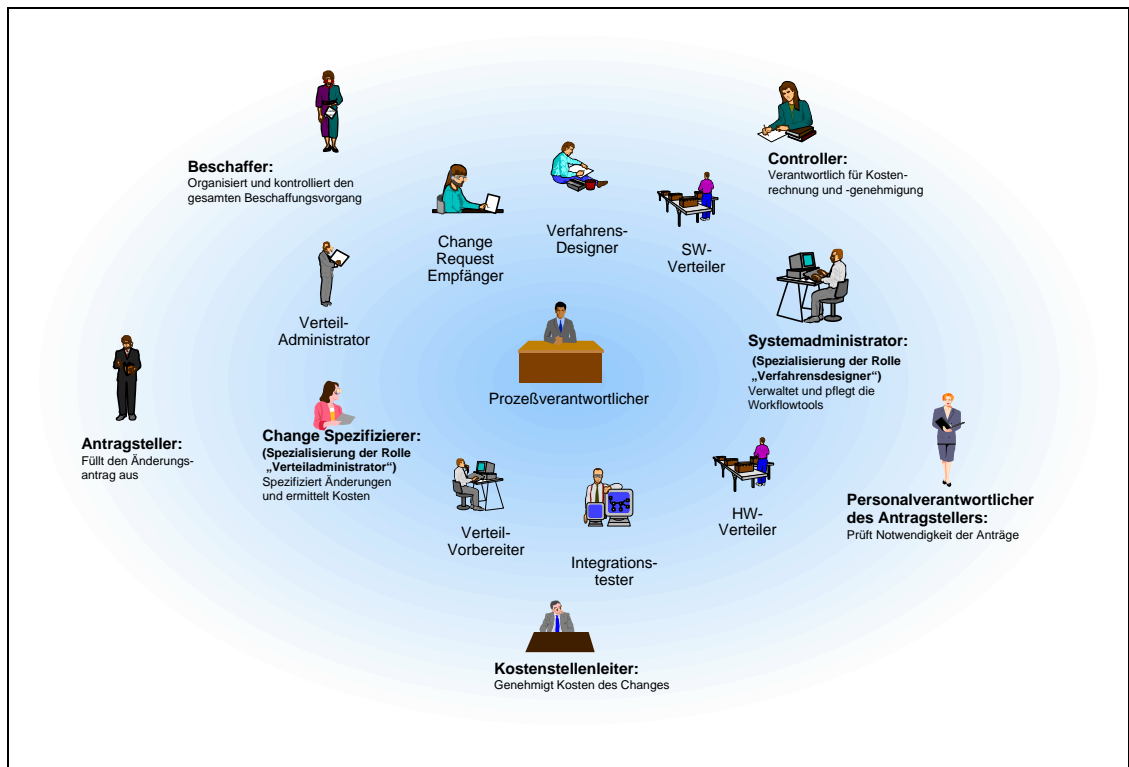


Abb. 4.3.3-1: Rollen im Change Management-Prozeß [vgl. IBM 1998c, S. 2]

Zu der Abbildung muß angemerkt werden, daß nicht alle Rollen aktiv im Prozeß teilnehmen. So pflegt der „Verfahrensdesigner“ die Vorgangstypen der konkreten Implementierung und erweitert ggf. das System, wobei er vom „Systemadministrator“ bei administrativen Tätigkeiten unterstützt wird. Allgemein kann angemerkt werden, daß die Häufigkeit der Teilnahme am Prozeß von innen nach außen abnimmt (so nehmen „Beschafter“ und „Controller“ nur im Rahmen einer Aufgabe am gesamten Prozeß teil,

während der „Prozeßverantwortliche“ permanent mit dem System arbeiten wird, um die darin verwalteten Changes zu überwachen - eine Funktionsbeschreibung jeder Rolle kann der Quelle [IBM 1998c, S. 3 ff.] entnommen werden).

In Abbildung 4.3.3-1 fehlende Tätigkeitsbeschreibungen werden in Abbildung 4.3.3-2 unter Verwendung des Prozeßgrobdesigns näher erläutert. Hierbei handelt es sich um die Kernrollen des Projekts, da alle zusätzlichen Rollen erst im Verlauf der Implementierung hinzugekommen sind, was teilweise mit der Einbindung zusätzlicher Abteilungen zusammenhängt.

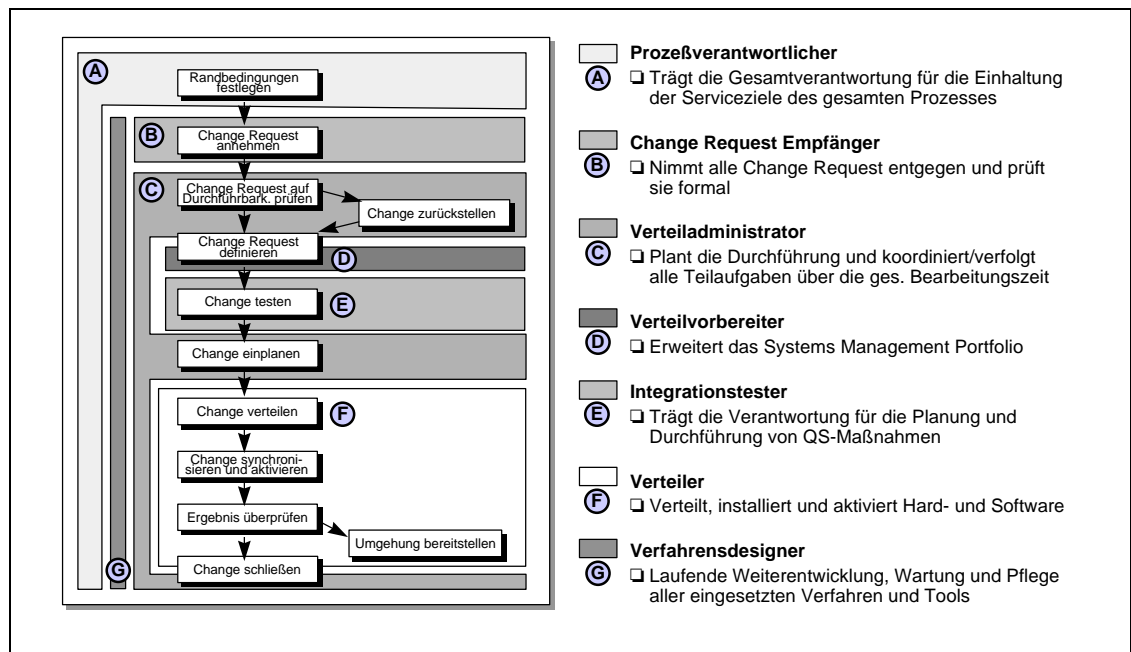


Abb. 4.3.3-2: Aufgaben und Verantwortungen der Kernrollen [vgl. IBM 1997b, S. 11]

4.3.4 Übersicht über den Prozeßfluß

Die Grundlage für die Beschreibung des Change Management-Prozesses bildete der in Abbildung 4.3.4-1 dargestellte Prozeßfluß. Die Abbildung spiegelt sehr gut die Komplexität des Prozesses wie auch die Verknüpfungen zwischen den einzelnen Prozeßgruppen des Systems Managements (dargestellt in den Ellipsen) wider. Da eine detaillierte Erläuterung des Prozeßflusses den Rahmen dieser Arbeit bei weitem sprengen würde, sei der Leser auf die Beschreibung im rechten Teil der Abbildung hingewiesen.

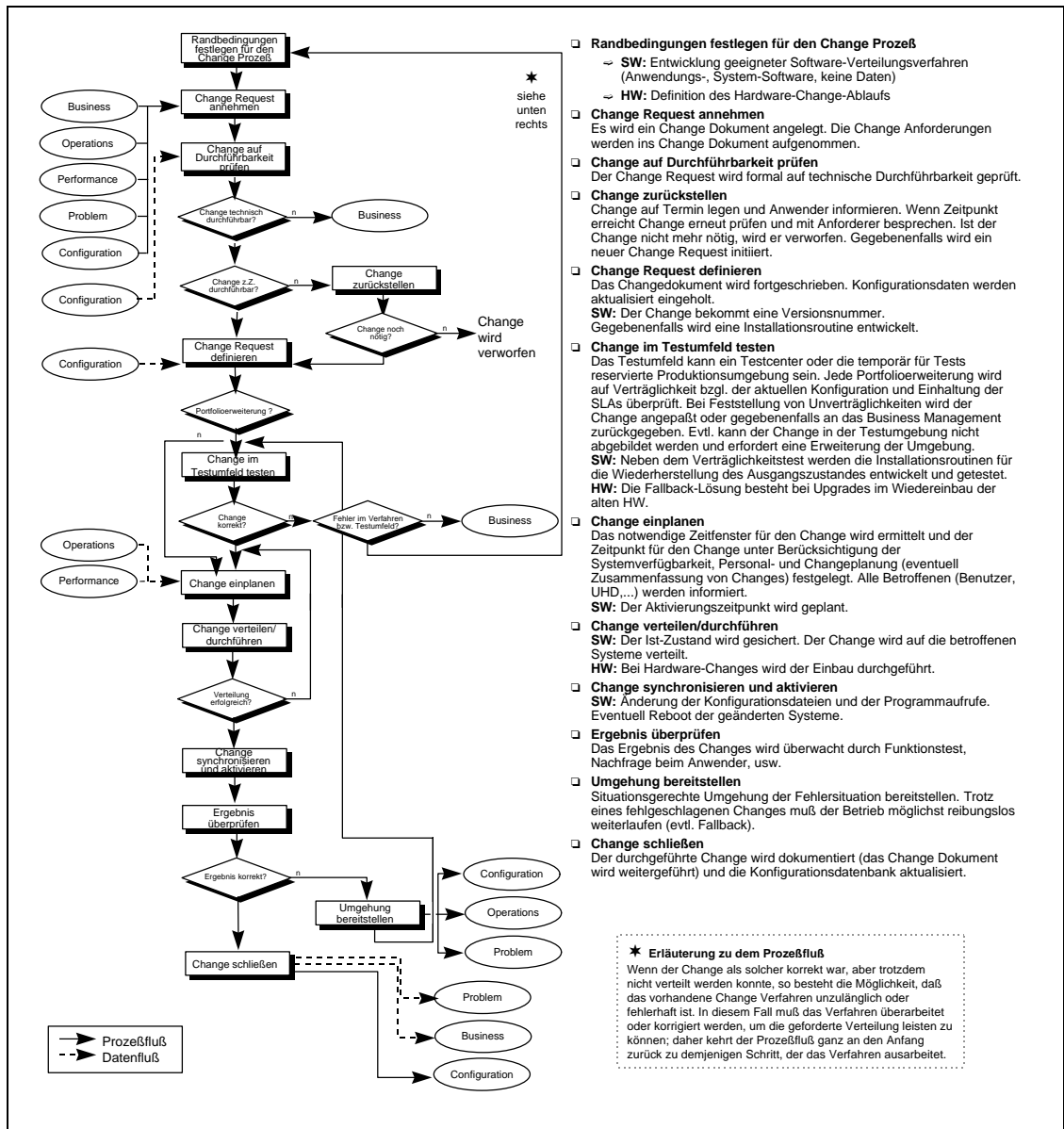


Abb. 4.3.4-1: Prozeßfluß im Change Management [vgl. IBM 1997b, S. 12 f.]

4.3.5 Qualitätsparameter

„Ziel des Change Management ist die termingetreue, qualitätsgesicherte und hoch-automatisierte Durchführung und Dokumentation von Änderungen.“ [IBM 1998d, S. 1]

Allgemein lassen sich Qualitätsparameter von kritischen Erfolgsfaktoren ableiten. Diese wurden für das Projekt wie in Abbildung 4.3.5-1 dargestellt und in [IBM 1998d, S. 1] konkretisiert: „Erfolgsfaktoren sind ein hoher Automatisierungsgrad, geringe Durchlaufzeiten, eindeutige Prozeßverantwortung, Aktualität der Konfigurationsdaten und, daß der Prozeß aktiv gelebt wird.“

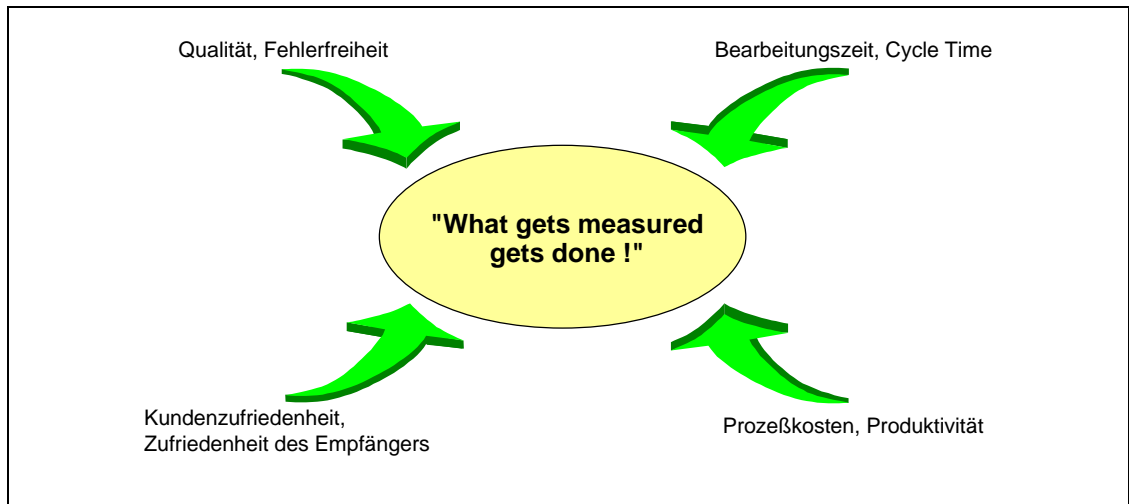


Abb. 4.3.5-1: Kritische Erfolgsfaktoren für Prozesse [IBM 1998b, S. 10]

Im Projekt wurden separate Ziele, Erfolgsfaktoren und Qualitätsparameter für die Change Vor- und Nachbereitung sowie für die Change Durchführung aufgestellt [vgl. IBM 1998d, S. 2 ff.]:

- Change Vor- und Nachbereitung:
 - Ziel der Vorbereitung ist die Bereitstellung aller Informationen, die für die Durchführung benötigt werden. Zusätzlich wird geprüft, ob eine Durchführung möglich ist oder der Change modifiziert oder abgebrochen werden muß.
 - Erfolgsfaktoren sind ein hoher Automatisierungsgrad, Aktualität und Vollständigkeit der Konfigurationsdaten und die Qualität der Prüfungen.
 - Qualitätsparameter: Anzahl der Änderungen (erfolgreich ggü. abgebrochen), Dauer erfolgreicher sowie Anzahl und Verweildauer noch offener Changes
- Change Durchführung:
 - Ziel ist die vollständige, fehlerfreie und termingerechte Durchführung der Änderung
 - Erfolgsfaktoren sind ein hoher Automatisierungsgrad, Fehlerfreiheit und Termintreue
 - Qualitätsparameter: Anzahl und Dauer der einzelnen Change-Typen wie auch die Dauer noch offener Change-Typen

Weitere Parameter zur Qualitätssicherung sind denkbar [vgl. IBM 1998d, S. 6], sind jedoch nicht im Projektumfang enthalten und sollen daher hier nicht weiter ausgeführt werden.

4.3.6 Eskalationsmanagement

In Workflow Management-Systemen läßt sich Eskalation wie folgt definieren: „Tritt bei einem Workflow-Dokument ein besonderes Ereignis auf, so soll dieses Ereignis einen Alarm auslösen und eine entsprechende Meldung soll erfolgen.“ [PAVONE 1997] Dies wirft die Frage auf, welche besonderen Ereignisse im Workflow-Umfeld existieren und wer im jeweils konkreten Fall benachrichtigt werden soll.

Als besondere Ereignisse wird in [PAVONE 1997] unterschieden:

- Überschreitung der maximalen Liegezeit für eine Aufgabe
- Gefährdung der rechtzeitigen Beendigung einer Workflow-Instanz
- Abbruch oder Delegation eines Vorgangsexemplars
- das Anschwellen der Bearbeitungsliste für eine bestimmte Person über einen kritischen Wert hinaus
- das Erreichen bestimmter Meilensteinaufgaben

Dabei kann insbesondere die Identifikation der Vorgangsexemplare, deren rechtzeitige Beendigung gefährdet ist, als schwer umsetzbar bezeichnet werden, da Workflow-Klassen mehrere Bearbeitungsenden, mehrere Möglichkeiten zum Erreichen dieser Enden sowie Schleifen aufweisen können.

Die Ermittlung des Empfängers einer Eskalation kann ebenfalls als problematisch angesehen werden. Hierbei kann es sich abhängig von der Aufgabe um verschiedene Personen handeln, da z. B. verschiedene Organisationseinheiten am Prozeßfluß teilnehmen. Auch ist die Art der Benachrichtigung festzulegen, wobei zwei generelle Konzepte unterschieden werden können: im Push-Konzept werden Statusnachrichten an die jeweiligen Empfänger versandt, während sich Vorgesetzte im Falle des Pull-Konzepts aktiv mit Hilfe von Report-Mechanismen über den Eskalationszustand informieren [vgl. PAVONE 1997].

Prinzipiell ist auch eine Status- und Prioritäten-orientierte Eskalation denkbar, in der dem Bearbeiter zeitüberschrittene Vorgangsexemplare in entsprechenden Überschreitungsklassen präsentiert werden. Ab einer gewissen Überschreitung kann dann die Priorität der Workflow-Instanz heraufgesetzt werden. Diese Maßnahmen würden implizit die Liegezeiten verkürzen und dafür sorgen, daß zuerst hochpriorisierte und zeitüberschrittene Vorgangsexemplare bearbeitet werden [vgl. PAVONE 1997].

Im vorliegenden Projekt wird die Informationsregelung „primär nicht als (vertikale) Kontrollfunktion verstanden, sondern soll den (horizontalen) Prozeßfluß unterstützen“ [IBM 1998c, S. 22]. „Kontrollmechanismen im herkömmlichen Sinne mit leistungsabhängiger, vertikaler Eskalation sind nicht vorgesehen.“ Vielmehr „wird die Unterstützung einer gleichmäßigeren Verteilung der Arbeitslast angestrebt“, und es soll eine „Kommunikation zwischen Rollen und Stellvertreterrollen“ sichergestellt werden [IBM 1998c, S. 23].

4.3.7 Systems Management Portfolio-Datenbank

Die Entwicklung einer Portfolio-Datenbank mit den entsprechenden Prozessen zur Pflege sowie Evaluierung und Aufnahme von Produkten stellt eine Erweiterung des ursprünglichen Projekts dar, da die Datenbank eigentlich parallel von der Deutschen Bundesbank entwickelt werden sollte, was aber mangels Kapazitäten für die Realisierung nicht möglich war.

Eine Einführung eines Change Management-Systems ohne eine Portfolio-Datenbank erschien aber nicht sinnvoll, da durch die Portfolio-Datenbank „in vielen Fällen der Prüfungs- und Beratungsaufwand innerhalb des Change Management-Prozesses erheblich verringert“ wird, da „diese Produkte .. bereits ausgetestet und vollständig spezifiziert“ sind [IBM 1998e, S. 2]. Damit kann die Portfolio-Datenbank selbst, wie auch die zu ihrem Management benötigten Prozesse dem weiteren Umfeld des Change Managements zugeordnet werden.

Aus diesem Grund wurde noch während des eigentlichen Projekts mit der Entwicklung einer entsprechenden Datenbank begonnen. Dieses neue Teilprojekt soll daher in den wesentlichen Teilen vorgestellt werden, da diese auch direkt das Change Management-System betreffen.

„Die Portfolio-Datenbank ... enthält Informationen über die wichtigsten Produkte im DV-Bereich ... So ist z. B. ersichtlich, welche Produkte getestet wurden, welchen Servicelevel der Anwender für Produkte erwarten kann und wie sich die Standard-Beschaffungspakete zusammensetzen.“ [IBM 1998f, S. 3] Außerdem wird die Datenbank als Schnittstelle zum Change Management-System eingesetzt und dient dort „der schnellen und einfachen Auswahl von Produkten für den elektronischen Beschaffungs-

antrag.“ [IBM 1998f, S. 3] Ferner wird die Datenbank in Zukunft als Arbeitsgrundlage für das Portfolio-Team im Rahmen des Portfolio-Erweiterungsprozesses dienen.

Unterschieden wird in erster Linie zwischen Produkten und Beschaffungspaketen. Diese werden dem Anwender auf Wunsch verschieden aufbereitet präsentiert, so daß eine einfache Auswahl möglich ist (z. B. nach Produktkategorie, Einsatzempfehlung, Servicekategorie usw.). Hervorzuheben ist dabei der benötigte hohe Informationsgehalt, so daß Benutzer Vor- und Nachteile verschiedener Produkte gegeneinander abwägen können. Dazu gehören exakte Produktbeschreibungen, Hinweise auf Abhängigkeiten, die Unterstützungskategorien der DV-Abteilung, Preise und ggf. zugehörige Produkte, die sinnvollerweise zusätzlich beschafft werden sollten [vgl. IBM 1998f, S. 3 ff.].

Zusätzlich ist auf eine einfache Übernahme der ausgewählten Produkte und Beschaffungspakete in den elektronischen Beschaffungsantrag zu achten. Dabei sollen wichtige Daten für die weitere Bearbeitung insbesondere im Hinblick auf die spätere Zuordnung zu den Change-Typen mit aus der Portfolio-Datenbank übernommen werden. Für die Bearbeiter im weiteren Prozeßfluß muß es möglich sein, zu erkennen, ob die Daten im elektronischen Beschaffungsantrag aus der Portfolio-Datenbank übernommen oder manuell eingetragen wurden. Bei einer Übernahme aus dem Portfolio sollte der komplette Produkteintrag für die weiteren Bearbeiter einfach wieder auffindbar sein, um dort ggf. nochmals spezielle Hinweise nachlesen zu können [vgl. IBM 1998f, S. 8].

5 Verwendete Arbeitsumgebung

Dieses Kapitel soll einen Überblick über die in der Projektimplementierung genutzten Plattform-Technologien geben (siehe Abbildung 5-1).

Als Basisplattform setzt die Deutsche Bundesbank unternehmensweit das Groupware-System Lotus Notes ein. Der in Kapitel 6 näher beschriebene, realisierte Prototyp wurde vollständig innerhalb der Applikation PAVONE Espresso umgesetzt, die wiederum auf Lotus Notes aufsetzt. Die benötigte Funktionalität innerhalb des Prototypen machte dabei eine Erweiterung der Funktionalität von Lotus Notes erforderlich, die durch das Genii Midas Rich Text LSX bereitgestellt wird.

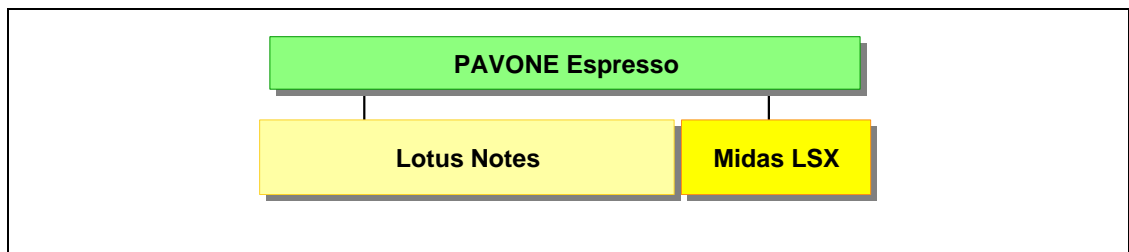


Abb. 5-1: Architekturübersicht über die genutzten Plattform-Technologien

5.1 Lotus Notes

Die Groupware-Plattform Lotus Notes ist nach einer aktuellen Studie der Gartner Group das dominierende System im traditionellen Groupware-Bereich der nicht-Web-basierten Zusammenarbeit [vgl. Gartner Group 1997]. Nach Angaben der Lotus Development Corporation setzen derzeit ca. 20 Millionen Benutzer diesen Groupware-Standard ein [vgl. Venetz 1998].

Aktuell in der Version 4.6.1 vorliegend, soll Lotus Notes im Rahmen dieser Arbeit in der Version 4.5 näher betrachtet werden, da die Deutsche Bundesbank die Version 4.5.2 in Deutsch einsetzt. Im Folgenden soll dabei lediglich ein Kurzüberblick über die für diese Arbeit wichtigsten Architekturmerkmale und Funktionalitäten gegeben werden, da eine vollständige Beschreibung den Rahmen dieser Arbeit bei weitem sprengen würde. Für eine genaue Darstellung sei auf die Quellen [IBM 1996], [Lotus 1996] und [Newbold/Lipton 1997] verwiesen.

Kurzdefinition von Lotus Notes

„Lotus Notes ist eine verteilte Client-/Server-Plattform, die es erlaubt, Applikationen zu entwickeln, deren enthaltene Daten von Benutzergruppen über ein Netzwerk gleichzeitig genutzt werden können. Notes besteht aus einer Sammlung von Dokumentendatenbanken auf Basis einer Kommunikationsinfrastruktur. Im Zusammenspiel mit der verteilten Speicherung und den Kommunikationseigenschaften erlaubt die integrierte Entwicklungsumgebung (Integrated Development Environment (IDE)) die Erstellung strategischer, unternehmensweiter Geschäftsapplikationen durch Rapid Prototyping (Rapid Application Development and Deployment (RADD)).“ [vgl. IBM 1996, S. 3 - Übersetzung durch den Autor]

Hauptfunktionalitäten und Architekturelemente von Lotus Notes

Wichtigstes Merkmal von Lotus Notes ist der Dokumenten-zentrierte Ansatz: ein *Dokument* wird in Notes als Basis-Informationsobjekt wahrgenommen, das sich aus verschiedenen Feldern und deren Inhalten zusammensetzt. Dabei werden Feldtypen unterschiedlichster Medialität unterstützt, so daß ein Dokument nicht nur Text sondern auch Graphiken, Videos, Dateianhänge und andere Datentypen enthalten kann. Somit handelt es sich um sogenannte Verbunddokumente [vgl. Nastansky et al. 1994, S. 285].

Über *Masken*, die statische Elemente wie z. B. Texte und Graphiken als auch die Felddefinitionen enthalten, werden die Dokumente dem Benutzer zur Ansicht gebracht und ermöglichen damit die Dokumentenerzeugung und -modifikation. Jedes Dokument kann über unterschiedliche Masken angezeigt werden, so daß je nach Kontext nur eine bestimmte Teilmenge der im Dokument enthaltenen Informationen angezeigt werden kann. *Teilmasken* geben dem Entwickler die Möglichkeit, bestimmte Maskenelemente konsistent über mehrere Masken hinweg zu benutzen, indem diese Teilmasken als Bestandteile der eigentlichen Hauptmasken eingefügt werden.

Mit Hilfe von *Ansichten* können Dokumente gezielt selektiert und aktiviert werden. Grundsätzlich sind Ansichten mit Inhaltsverzeichnissen vergleichbar, in denen eine unterschiedliche Auswahl an Dokumenten (z. B. „alle Dokumente“ oder nur „meine Aufgaben“) unterschiedlich sortiert (z. B. „nach Datum“ oder „nach Priorität“) angezeigt werden. Die Ansichten können zwischen allen Benutzern geteilt werden oder aber benutzerspezifisch sein (sog. Private Ansichten). Um gelegentliche Benutzer durch die oft hohe Anzahl von Ansichten nicht zu verwirren, stellen *Navigatoren* eine graphische

Oberfläche zur Verfügung, in der die Hauptansichten ausgewählt oder Aktionen wie z. B. das Erstellen von Dokumenten ausgeführt werden können.

Zur Automatisierung bestimmter Aktionen werden *Agenten* eingesetzt, die zeit- oder ereignisgesteuert aufgerufen werden können. So können beispielsweise einmal wöchentlich alte Dokumente von einem Agenten archiviert oder einem Benutzer eine Benachrichtigung zugeschickt werden, sobald ein Dokument neu erzeugt oder verändert wurde.

Diese Architekturelemente, die intern auch als (Architektur-)Dokumente verwaltet werden, und die (Daten-)Dokumente werden in *Datenbanken* zusammengefaßt. Somit enthalten Lotus Notes-Datenbanken nicht nur Daten, sondern gleichzeitig auch die Funktionalität zur Bearbeitung und Nutzung dieser Daten.

Die Datenbanken können dabei auf Notes-Servern im LAN- oder WAN-Umfeld zur Verfügung gestellt werden, so daß mehrere Benutzer gleichzeitig auf diese Datenbanken zugreifen können, oder aber lokal beim Benutzer vorliegen, was den mobilen Endanwender ohne ständige Serveranbindung unterstützt. Der Abgleich zwischen den *verteilten Datenbanken* wird durch *Replikation* sichergestellt, die automatisch alle Änderungen an den Dokumenten zwischen den Datenbanken abgleicht.

Leistungsstarke und hochgradig skalierbare *Sicherheitsmechanismen* regeln den Zugriff bis hinunter zu einzelnen Feldern in Dokumenten. Basis des Sicherheitskonzepts ist dabei die Authentifizierung des Benutzers über dessen Nutzerkennung und Paßwort. Der Zugriff auf Datenbanken kann dann für jeden Benutzer individuell eingestellt werden, wobei die einzelnen Zugriffsstufen durch unterschiedliche Rechte gekennzeichnet sind. Durch Verschlüsselung können einzelne Felder oder gesamte Dokumente gegen unbefugte Zugriffe geschützt werden. Ein im Rahmen des Projekts wichtiger Sicherheitsmechanismus ist die digitale Unterschrift, mit deren Hilfe einzelne Feldinhalte signiert werden können. So wird sichergestellt, daß die Information tatsächlich vom angegebenen Benutzer stammt und nachträglich nicht verändert wurde.

Die *integrierte Gruppenkommunikation* umfaßt ein leistungsstarkes E-Mail-System mit einer Adreßbuch-Verwaltung (Directory Services) ebenso wie die Nutzung gemeinsamer Datenbanken (Information Sharing).

Möglichkeiten der Programmierung in Lotus Notes

Entwicklern bietet Lotus Notes mehrere Möglichkeiten, die benötigte Funktionalität in Anwendungen zu realisieren. Die leistungsstarke Entwicklungsumgebung ist dabei vollständig in die Vollversion von Lotus Notes integriert.

Die einfachste Form der Programmierung ist die Nutzung vordefinierter Funktionen, den sog. *Simple Actions*. Diese können in einem geführten Dialog gesetzt werden und verlangen daher kaum Entwicklungskennntnisse. Mögliche Aktionen sind z. B. das Ändern von Feldinhalten oder das Versenden von Dokumenten.

Die älteste Form der Programmierung in Lotus Notes ist eine komplexe Formelsprache, die sog. *@-Funktionen*, mit denen sich kurze, aber leistungsstarke Formeln erstellen lassen. Allerdings reicht das Leistungsspektrum nicht an eine richtige Programmiersprache heran, da z. B. keine Schleifen programmiert werden können.

Aus diesem Grund wurde Lotus Notes ab der Version 4.0 mit *LotusScript* eine BASIC-kompatible, objektorientierte und multi-plattformfähige Programmiersprache hinzugefügt. Die integrierte Entwicklungsumgebung unterstützt dabei den kompletten Entwicklungsprozeß ähnlich wie herkömmliche Entwicklungsumgebungen anderer Programmiersprachen. Erweitert werden kann das Klassenmodell von LotusScript durch die Einbindung eigener Klassendefinitionen, den sog. *LotusScript Extensions (LSX)*, die dann wie alle anderen LotusScript-Klassen genutzt werden können. Abgelegt wird der Script-Code an den jeweiligen Objekten oder in sog. *Script-Bibliotheken*, so daß der dort abgelegte Code von mehreren Objekten genutzt werden kann.

Ferner können über die *Notes API* (API: Application Programming Interface, dt. Anwendungsprogrammchnittstelle) externe Anwendungen auf Notes-Datenbanken in ähnlicher Art und Weise zugreifen, wie dies Benutzer über Lotus Notes tun. Dadurch können externe Applikationen Daten in Notes-Datenbanken verändern sowie im- und exportieren.

5.2 Genii Midas Rich Text LSX

Das Midas Rich Text LSX der amerikanischen Firma Genii Software erweitert die Programmiersprache LotusScript um Klassendefinitionen, mit deren Hilfe Feldinhalte vom Datentyp Rich Text besser bearbeitet werden können (siehe Abbildung 5.2-1). Der

Datentyp Rich Text zeichnet sich dadurch aus, daß in einem solchen Feld nicht nur Text sondern auch formatierter Text, Tabellen, Graphiken, Videos und alle Arten von Objekten wie z. B. OLE-Objekte und Dokumentenverknüpfungen enthalten sein können.

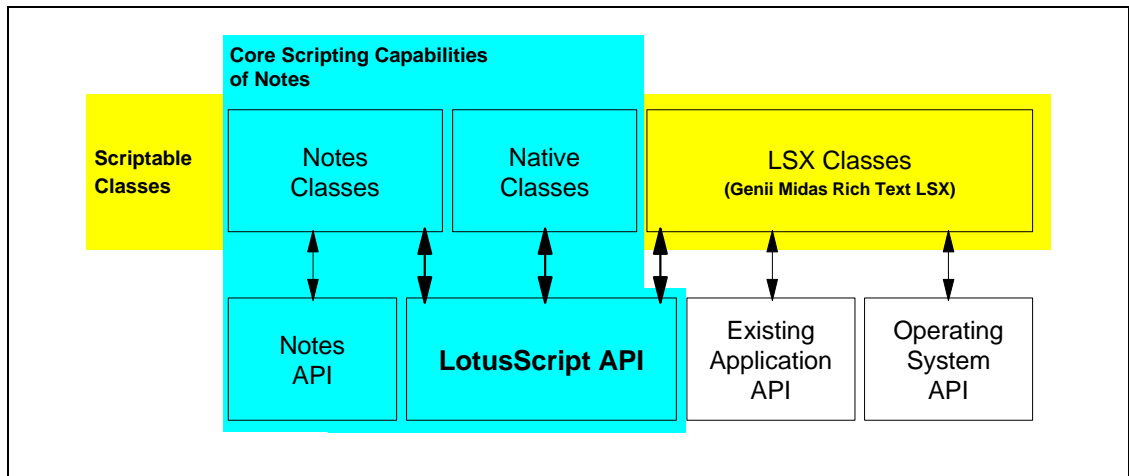


Abb. 5.2-1: Erweiterung von LotusScript um das Midas Rich Text LSX [vgl. Lotus 1996, S. 172]

Zwar bietet LotusScript eine eigene Klasse „NotesRichTextItem“, jedoch ist deren Funktionalität für die Umsetzung der Projektanforderungen nicht ausreichend. So ermöglicht Notes, bestehende Rich Text-Elemente auszulesen, jedoch können diese nicht in ihrem Aussehen verändert werden. Die Firma Genii spricht daher von einer statischen Funktionalität [vgl. Genii 1998].

Midas dagegen bietet die Möglichkeit der Änderung bestehender Rich Text-Elemente. Insbesondere umfaßt dies die dynamische Erzeugung von Tabellen wie auch die dynamische Formatierung einzelner Tabelleninhalte und -eigenschaften. Genau diese Funktionalität wurde im Projekt benötigt, kann aber mit LotusScript allein nicht erreicht werden.

5.3 PAVONE Espresso

„PAVONE Espresso ist ein gruppenbasiertes Workflow- und Office-Management-System auf Basis von Lotus Notes.“ [PAVONE 1998a, S. 4]

Da auf Wunsch der Deutschen Bundesbank auf die Office Management-Komponente im Projekt verzichtet werden sollte, wurde diese Komponente in enger Zusammenarbeit zwischen der Produktentwicklung und dem Projektteam entfernt, so daß eine eigenständige Workflow Management-Lösung auf Basis von PAVONE Espresso entstand.

Diese ist dadurch charakterisiert, daß spezielle Office Management-Funktionalitäten des Hauptprodukts entfernt wurden, die Konzepte und Basisfunktionen (z. B. persönliche Wiedervorlage oder zur Kenntnisnahme) jedoch nach wie vor auch in der Workflow-Version enthalten sind und deshalb auch genutzt werden können. Im Rahmen dieser Arbeit soll daher die Workflow-Version näher vorgestellt werden, ohne die speziellen Merkmale der Office Management-Komponente zu berücksichtigen oder weiter auf die Integrationspotentiale der beiden Komponenten einzugehen.

Abbildung 5.3-1 gibt einen Überblick über die grundlegende Architektur von PAVONE Espresso, die im weiteren Verlauf näher erläutert werden soll. Bei den als „PAVONE Espresso-Komponenten“ bezeichneten Merkmalen handelt es sich um Lotus Notes-Datenbanken, während die „PAVONE Espresso externe Anwendungen“ eigenständige Applikationen sind, die über die Notes API auf die Datenbanken zugreifen.

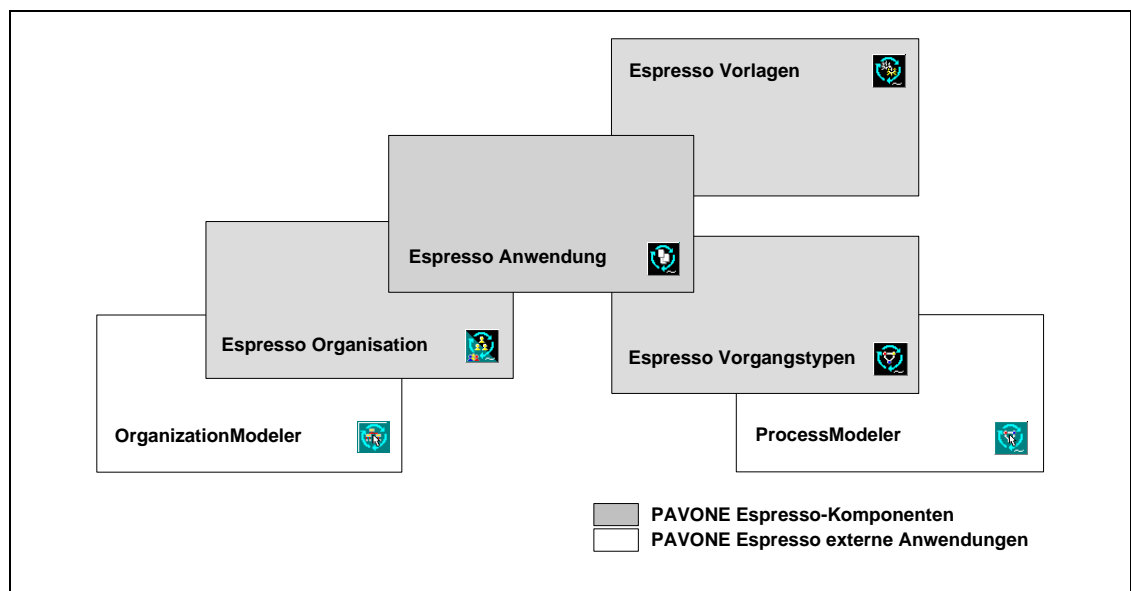


Abb. 5.3-1: PAVONE Espresso Architekturüberblick [vgl. PAVONE 1998a, S. 3]

Im Hinblick auf die in Kapitel 3.2 formulierten Anforderungen an Workflow Management-Systeme kann festgestellt werden, daß diese von PAVONE Espresso nahezu vollständig abgedeckt werden:

- Für das Workflow-Referenzmodell stellt der ProcessModeler ein Process Definition Tool dar, während die Espresso-Komponenten den Workflow Enactment Service sowie die Interfaces 2, 3 und 5 beinhalten. Das Interface 4 kann als generelles Problem heutiger Workflow Management-Systeme angesehen werden, da noch kein allgemein anerkanntes Austauschformat existiert. Daher wird dieses Interface auch von PAVONE Espresso nicht speziell unterstützt.

- Komplette abgedeckt wird der Workflow Life Cycle: Modellierung und Simulation übernimmt der ProcessModeler, bereitgestellt wird die spezifizierte Workflow-Klasse in der Espresso Vorgangstypendatenbank und die Ausführung zur Laufzeit sowie die nachträgliche Analyse wird durch die Espresso Anwendung abgedeckt.
- Vollständig unterstützt wird auch das Workflow-Kontinuum im Zusammenspiel der Espresso Anwendung und des ProcessModelers. Die notwendige Funktionalität für Ad-hoc-Workflows stellt Lotus Notes als Basisplattform bereit.

5.3.1 Espresso Datenbanken

Die Lotus Notes-Datenbanken enthalten alle Informationen, die für die Benutzung von PAVONE Espresso notwendig sind. Damit unterstützt Espresso alle Hard- und Software-Plattformen, die auch von Lotus Notes unterstützt werden. Außerdem kann die Anwendung hierdurch über den Replikationsmechanismus von Lotus Notes auf verschiedenen Servern dem Benutzer bereitgestellt oder von diesem lokal an der eigenen Arbeitsstation genutzt werden.

Espresso Anwendung

„Die Anwendungsdatenbank dient der Aufzeichnung und Verwaltung sämtlicher Geschäftsvorfälle und Dokumente, der Kommunikation zwischen den Organisationsmitgliedern, dem Management von einzelnen Vorgängen und der Kontrolle über die Ausführung von einzelnen Aufgaben.“ [PAVONE 1998a, S. 20]

Damit ist die Anwendungsdatenbank die einzige Datenbank, in der der Endbenutzer aktiv arbeitet, während die anderen Datenbanken lediglich als Repositories dienen, auf die zur Laufzeit zugegriffen wird. Aus diesem Grund soll nun die Funktionalität im Hinblick auf die Workflow-Komponente näher erläutert werden [vgl. PAVONE 1998a, S. 111 ff.].

Jeder Benutzer besitzt die Möglichkeit, neue Instanzen von Vorgangstypen zu bilden, für deren erste Aufgabe er als Bearbeiter eingetragen ist. Allgemein muß festgestellt werden, daß Espresso die Sicherheitsmechanismen von Lotus Notes ausgesprochen stark nutzt, so daß nur die jeweils als Bearbeiter zugewiesenen Personen ein Dokument verändern dürfen. Alle anderen Personen können diese Dokumente nur lesen.

Zu bearbeitende Dokumente findet der Benutzer in persönlichen Aufgabenlisten vor, die unterschiedlich sortiert dargestellt werden können. Jedes Vorgangsexemplar enthält den

aktuellen Aufgabennamen sowie die zugehörige Aktivitätenliste und Aufgabenbeschreibung. Nach dem Abschließen der Aufgabe überprüft die Workflow Engine die Eingaben und weist auf fehlende Daten oder andere Unstimmigkeiten hin. Werden keine Probleme festgestellt, so leitet die Workflow Engine das Dokument auf Basis der eingegebenen Daten an die korrekte(n) Folgeaufgabe(n) und deren Bearbeiter weiter.

Häufig ist die Nutzung dieses starren Konzepts standardisierter Workflow-Klassen in der Praxis jedoch nicht einsetzbar. Daher wird die Espresso Anwendung Endbenutzergetriebenen Flexibilitätsanforderungen gerecht. Dies beginnt mit der Bereitstellung einer Auswahl möglicher Weiterleitungsoptionen an den Benutzer, beinhaltet die Möglichkeit, Ausnahmebehandlungen vorzunehmen (u. a. Aufgabe delegieren oder andere Aufgabe für die Workflow-Instanz auswählen), und reicht bis zur Definition einmaliger Ad-hoc-Vorgänge.

Espresso Organisation

„Die Organisationsdatenbank dient der Verwaltung aller Ressourcen, die im PAVONE Espresso-Umfeld eingesetzt werden. Eine Ressource ist eine Person, die durch einen Namen repräsentiert wird.“ [PAVONE 1998a, S. 62]

Organisiert werden die Ressourcen dabei in Abteilungen, Arbeitsgruppen und Rollen (vgl. Abbildung 5.3.1-1). Diese werden primär in Vorgangstypen den einzelnen Aufgaben zugeordnet und werden dann zur Laufzeit zu den einzelnen Personen aufgelöst. Dadurch können häufig auftretende Änderungen in der Organisationsstruktur flexibel gehandhabt werden, ohne daß alle Workflow-Klassen jedesmal entsprechend geändert werden müssen.

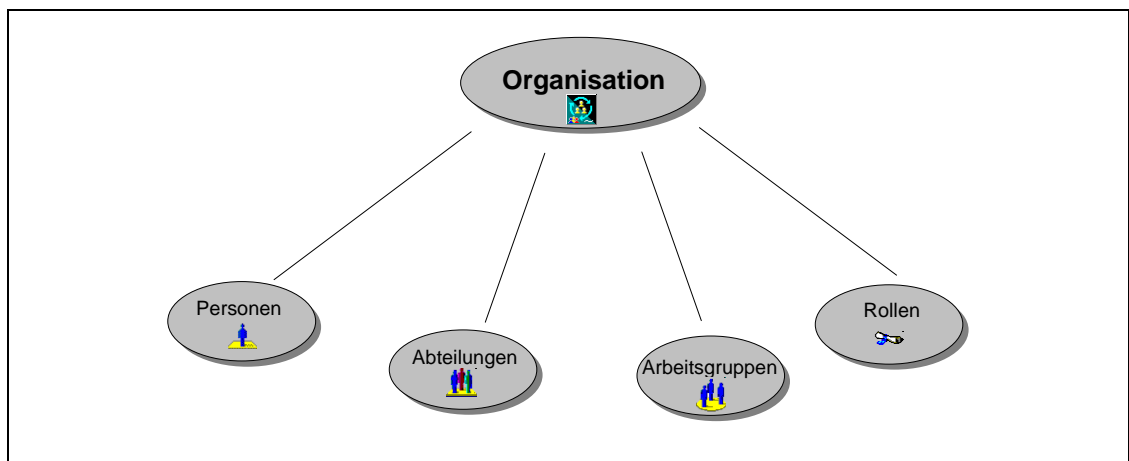


Abb. 5.3.1-1: Organisationseinheiten von PAVONE Espresso [PAVONE 1998a, S. 63]

Espresso Vorgangstypen

In der Vorgangstypendatenbank werden die erstellten Workflow-Klassen gespeichert. Zur Laufzeit werden diese Daten dann von der Anwendungsdatenbank ausgelesen, um mögliche Nachfolgeraufgaben zu ermitteln.

Die einzelnen Vorgangsobjekte (Aufgaben) werden als je ein Dokument abgespeichert und sind durch verschiedene Attribute charakterisiert. Als wichtigste Attribute sind hier Aufgabenname, Bearbeiter, Aktivitäten der Aufgabe, Aufgabenbeschreibung und Weiterleitungsbedingungen anzuführen. Ferner werden interne Felder für die Workflow Engine sowie zur Präsentation im Espresso ProcessModeler gespeichert.

Espresso Vorlagen

Die Vorlagendatenbank stellt spezielle Office Management-Funktionen in Form von Briefköpfen und Formeln zur Adressierung von Briefen zur Verfügung. Vorlagen dienen zur Anwendung als Standardformulare, in denen auch auf Endbenutzerapplikationen wie z. B. Microsoft Word oder Lotus 1-2-3 zurückgegriffen werden kann. Desweiteren können Schlüsselwortlisten für die Verwendung in Masken innerhalb der Anwendungsdatenbank definiert werden. Als Beispiele seien hier eine Staatenliste oder eine Liste mit möglichen Briefanreden angeführt. Ferner kann in dieser Datenbank eine äußerst flexible, Masken-basierte Archivierungslogik hinterlegt werden.

5.3.2 Espresso ProcessModeler

„Der ProcessModeler ist ein grafisches Vorgangstypen-Modellierungswerkzeug.“
[PAVONE 1998a, S. 76]

Abbildung 5.3.2-1 zeigt die Oberfläche anhand eines implementierten Beispielprozesses. Aufgaben werden als Objekte in der Modellierungsoberfläche abgelegt und können über eine Dialogbox mit Attributen (Aufgabenname, Bearbeiter usw.) belegt werden. Verbindungen zwischen den Objekten, die durch einfaches Anklicken der zu verbindenden Objekte realisiert werden, stellen Weiterleitungsmöglichkeiten dar. Auch diese können mit Attributen versehen werden, wobei die Weiterleitungsoptionen „Immer“, „Mehrfachauswahl“, „Exklusivauswahl“, „Bedingung“ (Formel kann eingegeben werden) und „Sonst“ unterstützt werden. Zusätzliche Textobjekte erlauben eine Kommentierung der einzelnen Workflow-Phasen. Alle Objekte können dabei jederzeit über Drag&Drop-Mechanismen in der Oberfläche verschoben werden.

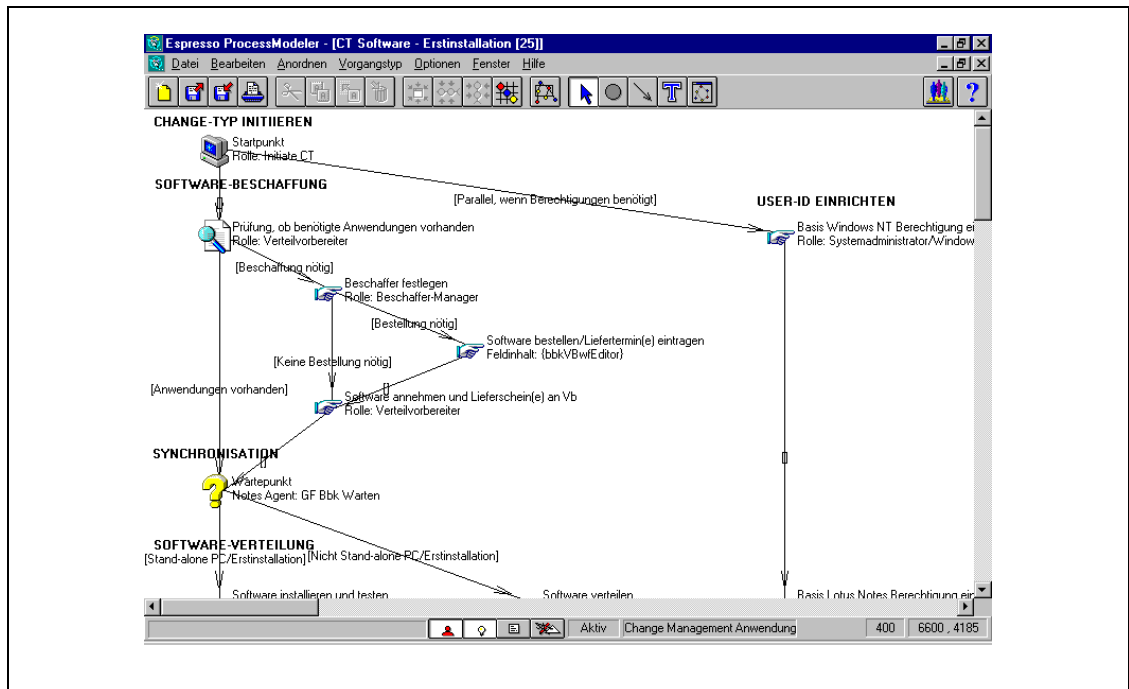


Abb. 5.3.2-1: Graphische Modellierungsoberfläche des Espresso ProcessModelers

Zudem erlaubt der ProcessModeler, über eine Verifikation die Korrektheit des Vorgangstypen zu überprüfen, während eine Simulation Problembereiche wie z. B. Ressourcenüberlastungen aufdeckt. Die so modellierten Workflow-Klassen werden dann in der Vorgangstypendatenbank abgelegt.

5.3.3 Espresso OrganizationModeler

Wie in Kapitel 5.3.1 beschrieben, dient die Organisationsdatenbank dazu, alle Ressourcen zu verwalten, die im PAVONE Espresso-Umfeld eingesetzt werden. Dies kann vollständig über die Masken innerhalb der Datenbank oder einfacher mit dem Espresso OrganizationModeler geschehen, der die Aufbauorganisation mit graphischen Elementen visualisiert. Personen und Arbeitsgruppen können dann mit Hilfe einfacher Drag&Drop-Interaktion aus dem Lotus Notes-Adreßbuch übernommen werden. Auch die Rollen-, Abteilungs- und Arbeitsgruppenzuordnung kann auf diese Weise vorgenommen werden.

Auf eine weitergehende Beschreibung wird an dieser Stelle aus Relevanzgründen verzichtet. Eine Anleitung zur Benutzung des OrganizationModelers kann [PAVONE 1998a, S. 62 ff.] entnommen werden.

6 Ausgewählte Implementierungsaspekte des Prototypen

Innerhalb dieses Kapitels sollen einige zentrale Aspekte der prototypischen Implementierung ausführlich vorgestellt werden.

6.1 Überblick

Zur besseren Orientierung soll zunächst ein Überblick über das entstandene Architekturmodell und den realisierten Prozeßverlauf gegeben werden, bevor die konkreten Aspekte der Implementierung im Einzelnen vorgestellt werden.

6.1.1 Architekturmodell

Basierend auf den Ausführungen zur verwendeten Arbeitsumgebung in Kapitel 5 kann das Architekturmodell der prototypischen Implementierung wie in Abbildung 6.1.1-1 dargestellt werden. Neben den bereits erläuterten Komponenten werden drei weitere Datenbanken als Repositories eingesetzt, um die Bedienung des Prototypen zu vereinfachen.

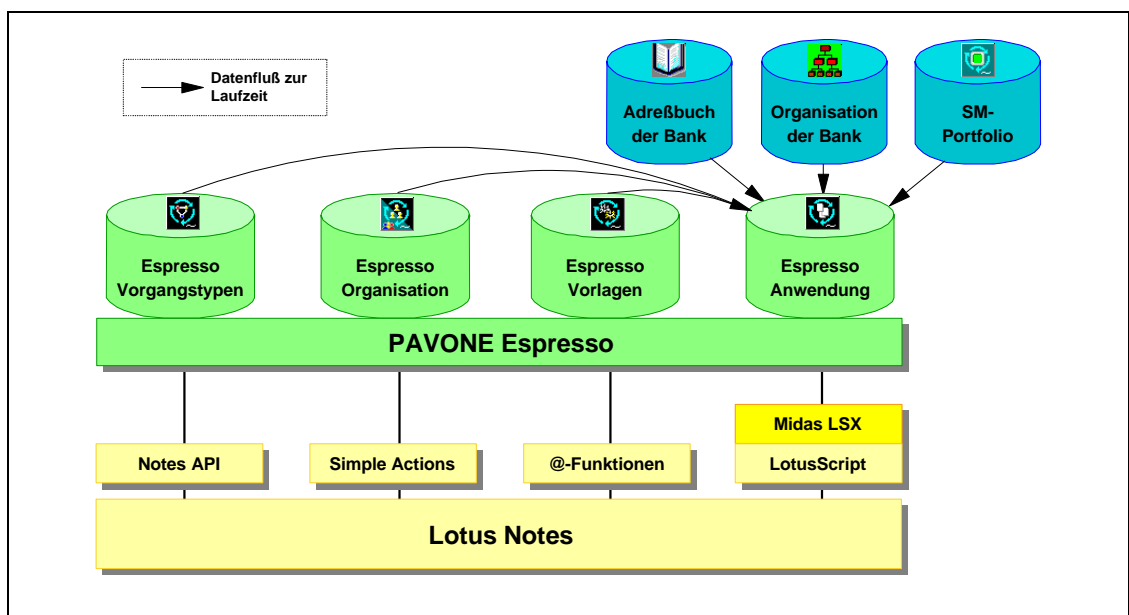


Abb. 6.1.1-1: Architekturmodell der prototypischen Implementierung

Wann immer personenbezogene Daten (Name, Telefonnummer usw.) benötigt werden, können die einzelnen Felder von Hand ausgefüllt werden. Eine einfachere Möglichkeit besteht darin, über eine Schaltfläche die Person(en) aus dem *Lotus Notes Namens-* und

Adreßbuch der Deutschen Bundesbank herauszusuchen. Die assoziierten Felder werden dann automatisch gefüllt, indem Name und E-Mail-Adresse dem Namens- und Adreßbuch entnommen werden, während die restlichen Daten (Telefon, Fax, Raum, Organisationsmerkmal und Kostenstelle) in der bereits vor Projektbeginn vorhandenen *Organisationsdatenbank der Deutschen Bundesbank* enthalten sind und von dort in die Maske übertragen werden. Somit werden im Sinne einer effizienten Nutzung der vorhandenen Datenbasis die Benutzereingaben minimiert.

Die *Systems Management Portfolio-Datenbank* wurde bereits in Kapitel 4.3.7 konzeptionell vorgestellt. Die enthaltenen Hard- und Software-Produkte können einfach in den elektronischen Beschaffungsantrag übernommen werden. Genauer wird die Benutzerinteraktion der Portfolio-Datenbank in Kapitel 6.4.3 beschrieben.

6.1.2 Realisierter Prozeßverlauf

Die Initiierung eines Change-Workflows kann, wie in Abbildung 6.1.2-1 dargestellt, auf drei unterschiedliche Arten geschehen.

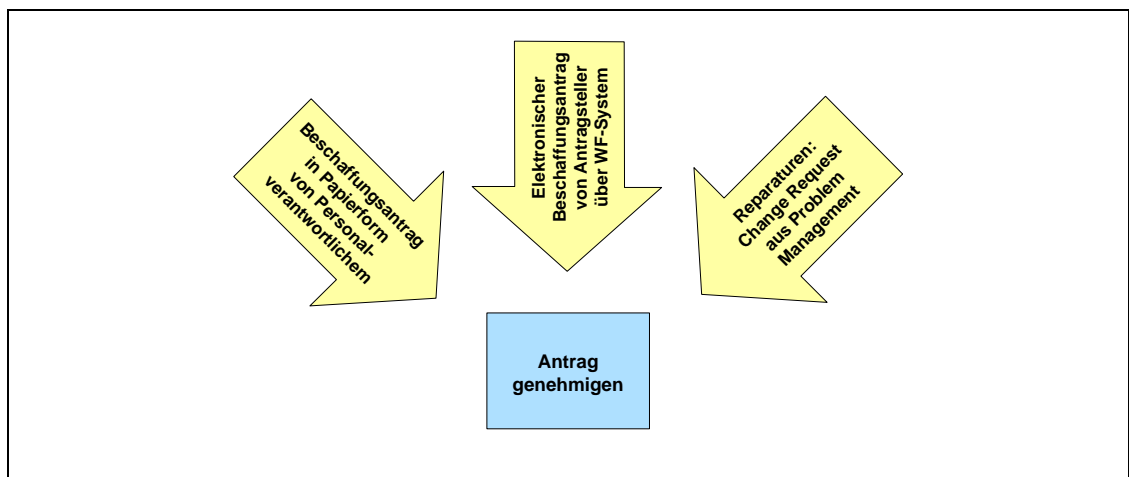


Abb. 6.1.2-1: Drei Möglichkeiten der Initiierung eines Change-Workflows [vgl.

IBM 1998b, S. 24]

Nutzt der Kostenstellenleiter noch nicht den Prototypen, so muß der Beschaffungsantrag nach wie vor papierbasiert gestellt werden. Der Antrag wird dann in das Change Management-System übernommen und im Folgenden elektronisch bearbeitet. Für die Kostengenehmigung wird ein ergänzender Ausdruck vorgenommen, der dem ursprünglichen Papierantrag angehängt wird. Der Kostenstellenleiter entscheidet auf Basis des ergänzten Papierantrags über eine Freigabe der Kosten. Diese Entscheidung wird in das System übernommen, und der Antrag läuft bis zum Ende elektronisch weiter.

Ist der Kostenstellenleiter dagegen aktiv in das Change Management-System eingebunden, so kann der Beschaffungsantrag direkt elektronisch gestellt und weiterverarbeitet werden. Medienbrüche werden komplett vermieden.

Kommt es im Rahmen einer Problemmeldung zu einem Hard- oder Software-Austausch (z. B. defekter Bildschirm erfordert Austauschgerät und Reparatur), so muß zunächst kein Beschaffungsantrag gestellt werden. Statt dessen werden die Daten aus dem Problem Management-System übernommen und der Austausch so dokumentiert (Aufstellung des Ersatzgerätes). Ist eine Ersatzbeschaffung nötig (Reparatur nicht möglich), muß der Anwender einen papierbasierten oder elektronischen Beschaffungsantrag mit Bezug auf die bestehende Problem- und Change-ID stellen. Nach erfolgreicher Reparatur oder Eintreffen des Ersatzgerätes wird dann das Austauschgerät wieder abgeholt und das eigentliche Gerät angeschlossen, was wiederum in dem vom Change Management initiierten Vorgang dokumentiert wird.

Mit den Ausführungen zur möglichen Initiierung eines Change-Workflows wurde bereits ein kleiner Einblick in den realisierten Prozeßfluß gegeben. Genauer wird dieser in Abbildung 6.1.2-2 dargestellt, wobei eine weiterreichende Erläuterung auf Aufgaben- oder gar Aktivitätenebene den Rahmen dieser Diplomarbeit sprengen würde. Eine detaillierte Beschreibung der realisierten Prozesse auf Aufgabenebene kann dem Benutzerhandbuch der Projektdokumentation [PAVONE 1998b, S. 62 ff.] entnommen werden.

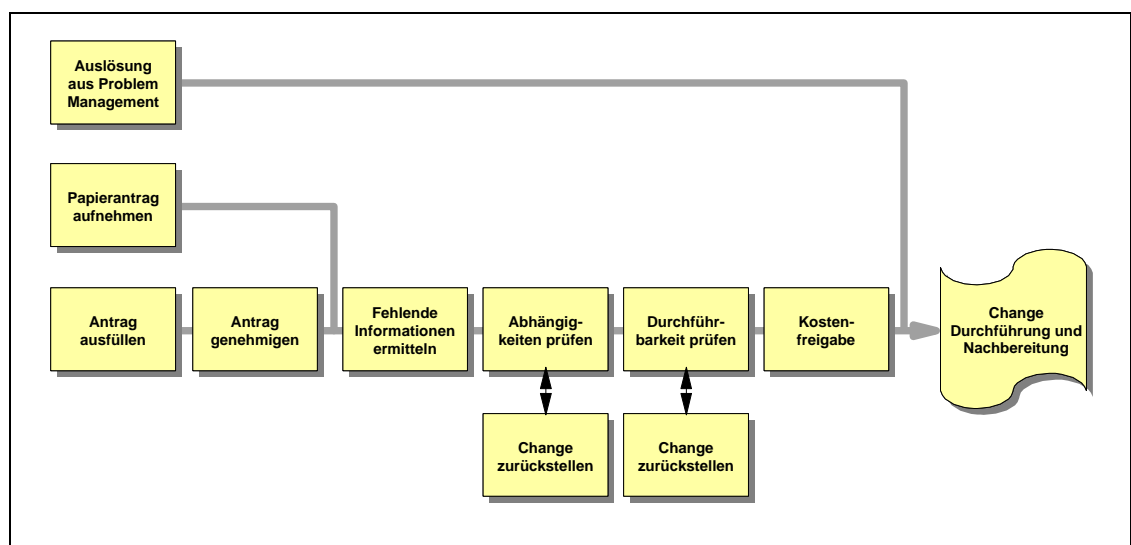


Abb. 6.1.2-2: Überblick über den realisierten Prozeßverlauf [vgl. IBM 1998b, S. 28]

6.2 Modularer Lösungsansatz

Das in Kapitel 4.3.1 vorgestellte Konzept der Geschäftsvorfälle und Change-Typen sowie der geforderte modulare Ansatz (vgl. Kapitel 4.2.2) machten es notwendig, zur Bearbeitung eines Change Requests mehrere Vorgangstypen heranzuziehen. Dieser Abschnitt soll zuerst die Notwendigkeit hierfür durch die Diskussion möglicher Prozeßarchitekturen verdeutlichen, um dann die konkreten Auswirkungen auf eine Beispielanforderung deutlich zu machen. Abschließend wird erläutert werden, wie die benötigte Funktionalität in der Hintergrundverarbeitung umgesetzt wurde.

6.2.1 Vergleich möglicher Prozeßarchitekturen

Betrachtet man die Abbildung 4.3.2-2 unter Berücksichtigung der in dem zugehörigen Kapitel getroffenen Feststellungen, so bietet sich an, die drei stark strukturierten Prozeßteile der Change Annahme, Vorbereitung und Nachbereitung innerhalb einer Workflow-Klasse gemeinsam abzubilden. Für die Change Durchführung stehen dann zwei Alternativen zur Verfügung (siehe Abbildung 6.2.1-1): alle möglichen Change-Typen als parallele Äste im gleichen Vorgangstypen zu modellieren oder aber eigenständige Prozesse zu bilden, die sich mit dem Hauptprozeß synchronisieren müssen.

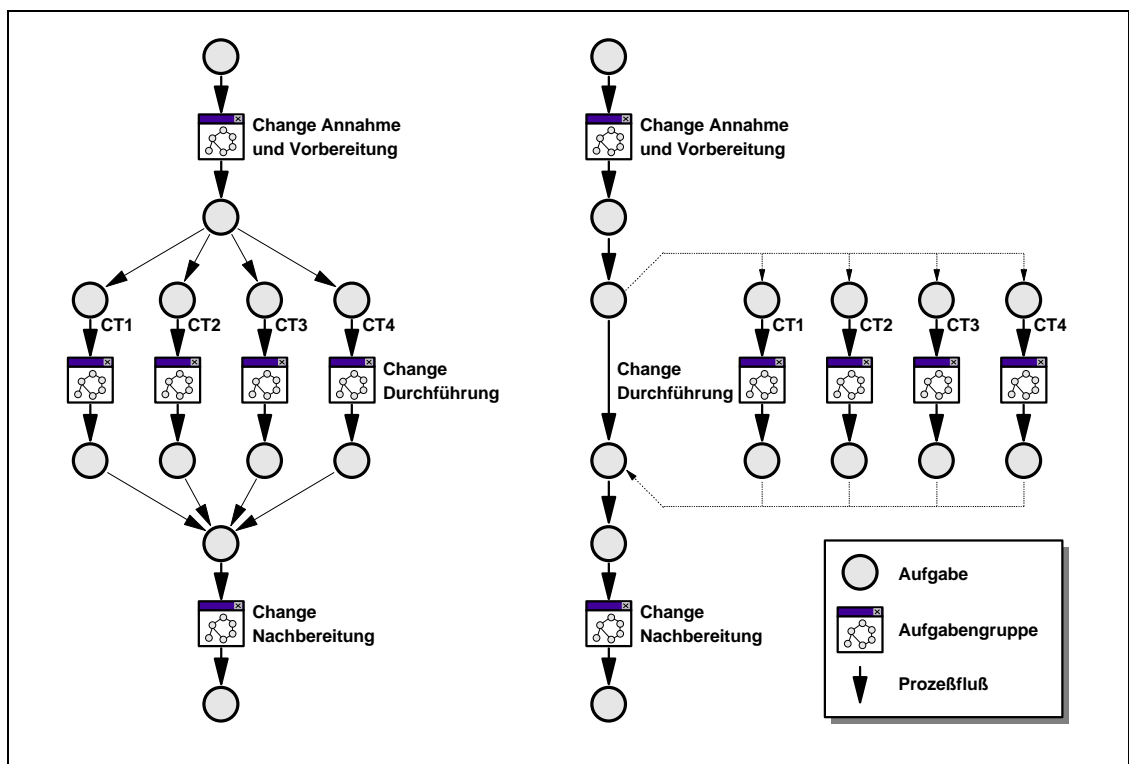


Abb. 6.2.1-1: Mögliche Realisierungsformen der Prozeßarchitektur

Die Einbeziehung in die gleiche Workflow-Klasse bedeutet, daß die Lösung dem gewünschten modularen Ansatz nicht gerecht wird, da selbst bei einer Änderung nur eines Change-Typen in der Change Durchführung (einem der parallelen Äste) bzw. bei einer Erweiterung um einen weiteren Change-Typen (weiteren parallelen Ast hinzufügen) der gesamte Vorgangstyp verändert wird.

Die zweite Möglichkeit der Prozeßarchitektur wird dem modularen Gedanken gerecht, jedoch muß eine Prozeßsynchronisation im verteilten Umfeld geschaffen werden, die das Standardprodukt PAVONE Espresso nicht abdeckt. Dadurch erhöht sich der Verwaltungs- wie durch die nötigen Synchronisationsaufgaben auch der Modellierungsaufwand erheblich.

Entscheidend für die Auswahl der Prozeßarchitektur war der bereits in Papierform existierende Beschaffungsantrag der Deutschen Bundesbank (vgl. Anlage C.1), über den bislang alle Hard- und Software-Anforderungen beantragt wurden, so daß dessen Struktur und Daten in den elektronischen Beschaffungsantrag überführt werden mußten. Bringt man diesen Antrag mit dem Modell der Geschäftsvorfälle und Change-Typen in Verbindung, so wird deutlich, daß auf einem Beschaffungsantrag mehrere Geschäftsvorfälle beantragt werden können. Dieser Fall liegt beispielsweise vor, wenn ein „neuer Arbeitsplatz“ und ein CD-ROM für einen vorhandenen PC bestellt werden. Folgt man für dieses Beispiel der Abbildung 4.3.1-2 in der Zuordnung der Change-Typen, so ergibt sich, daß der Change-Typ „HW, Standard-Change, Erstinstallation“ zweimal durchlaufen werden muß (einmal für den Hardware-Teil des neuen Arbeitsplatzes und einmal für das Hardware-Upgrade).

Dieser Umstand ist bei der Modellierung als parallele Äste jedoch nur umsetzbar, wenn die Abarbeitung sequentiell erfolgt (per Rücksprung an den Anfang des Astes) oder wenn die angeforderten Komponenten beider Geschäftsvorfälle zusammengefaßt durch diesen Ast geleitet werden. Beide Alternativen scheiden jedoch aus, da sie zu inflexibel sind. Andere Optionen wie das dynamische Duplizieren eines Astes zur Laufzeit oder das Bereitstellen mehrerer paralleler Äste des gleichen Change-Typs sind entweder technisch nicht realisierbar oder erscheinen nicht sinnvoll, da der Modellierungsaufwand insbesondere bei Änderungen als extrem hoch angesehen werden muß.

Somit verbleibt als Lösung nur die Umsetzung als eigenständige Vorgangstypen, bei der die nötige Anzahl an Vorgangsexemplaren je Change-Typ erzeugt wird. Allein diese

Prozeßarchitektur bietet auch die nötige Flexibilität zur Synchronisation von Abhängigkeiten zwischen den Workflow-Instanzen, die nun näher erläutert werden soll.

6.2.2 Synchronisation von Abhängigkeiten an einem Praxis-Beispiel

Durch die Realisierung der Change-Typen als eigenständige Workflow-Klassen ergibt sich eine erste Abhängigkeit, in der der Hauptprozeß nach dem Initiieren der Change-Typ-Prozesse auf deren Beendigung warten muß, bevor er selbst in die Change Nachbereitung eintreten darf.

Ferner müssen Interdependenzen berücksichtigt werden, die sich in drei Kategorien einteilen lassen:

- Abhängigkeiten zwischen Change-Anträgen (Antrag A muß vor Antrag B durchgeführt werden)
- Abhängigkeiten zwischen Geschäftsvorfällen eines Antrages (Geschäftsvorfall I muß vor Geschäftsvorfall II durchgeführt werden)
- Abhängigkeiten zwischen Change-Typen eines Geschäftsvorfalles (im Falle eines neuen Arbeitsplatzes muß erst die Hardware bereitstehen, bevor die Software darauf installiert werden kann)

Gelöst wurden die Interdependenzen durch Synchronisationsaufgaben in den Workflow-Klassen, die automatisiert durch Lotus Notes-Agenten bearbeitet werden. Das Konzept soll im Folgenden am Beispiel eines Geschäftsvorfalles „Neuer Arbeitsplatz“ erläutert werden, der die beiden Change-Typen „HW, Standard-Change, Erstinstallation“ und „SW, Standard-Change, Erstinstallation“ umfaßt (vgl. Abbildung 4.3.1-2).

In dem Beispiel wird nun der Change Request initiiert, durchläuft die Change Annahme und Vorbereitung und gelangt nach der Kostenfreigabe in die Change Durchführung. Hier werden im Hauptprozeß die einzelnen Change-Typ-Prozesse gestartet, und der Hauptprozeß wartet auf deren Abarbeitung. Abbildung 6.2.2-1 verdeutlicht diesen wie auch die folgenden Schritte im Rahmen der betrachteten Change-Typen. Die Initiierung der Change-Typen wurde dabei als Schritt ① gekennzeichnet.

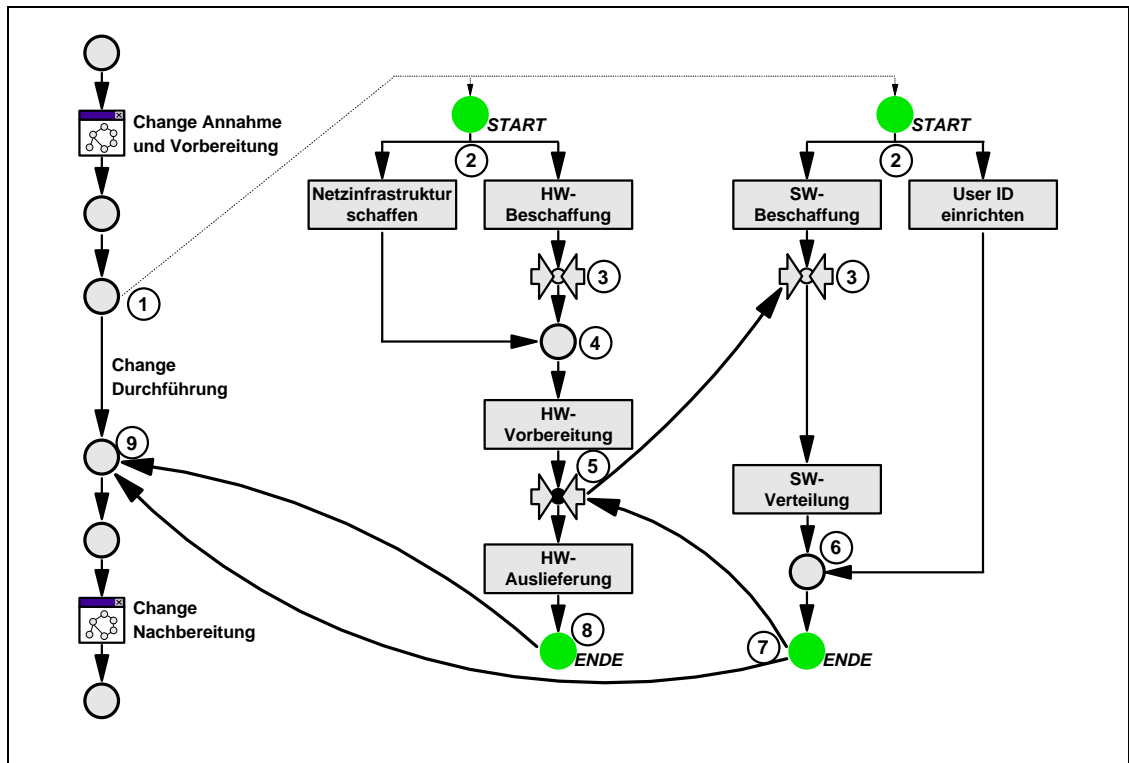


Abb. 6.2.2-1: Synchronisation der Change-Typ-Prozesse im betrachteten Beispiel

Über die Startaufgabe zur Initiierung des Change-Typs werden bestimmte Arbeitsschritte angestoßen (②), die parallel ablaufen können, da der Antrag bereits in der Change Vorbereitung sowohl technisch als auch betriebswirtschaftlich geprüft und genehmigt wurde.

Nach der HW- bzw. SW-Beschaffung ist jeweils ein Synchronisationspunkt vorgesehen, an dem auf Abhängigkeiten gewartet wird (③): der Hardware-Change-Typ (HW-CT) wartet auf Interdependenzen zu anderen Geschäftsvorfällen oder Change Requests, während der Software-Change-Typ (SW-CT) auf die gleichen Abhängigkeiten sowie auf den Abschluß der Hardware-Vorbereitung im HW-CT wartet.

Sind die Interdependenzen für die Hardware nun aufgelöst, so kann die Bearbeitung fortgesetzt werden. Es wird ein Zusammenführungspunkt erreicht, an dem auf die Bereitstellung der Netzinfrastruktur gewartet wird (④). Währenddessen wartet der SW-CT weiterhin an seinem Synchronisationspunkt darauf, daß der HW-CT die Hardware-Vorbereitung abgeschlossen hat.

Wurde die Netzinfrastruktur bereitgestellt oder mußte keine Netzinfrastruktur geschaffen werden, so kann die Hardware vorbereitet werden. Anschließend wird ein zweiter, Geschäftsvorfall-interner Synchronisationspunkt erreicht (⑤), an dem der SW-CT frei-

geschaltet wird. Gleichzeitig wird im HW-CT eine Interdependenz zum SW-CT eingetragen, so daß hier auf dessen Beendigung gewartet wird.

Der SW-CT verliert durch die Freischaltung die letzte Abhängigkeit (weitere müssen mit denen des HW-CT in Schritt ③ identisch gewesen und somit auch aufgelöst sein), so daß die Software-Verteilung vorgenommen werden kann. Es wird ein Zusammenführungspunkt erreicht (⑥), der wie Schritt ④ gehandhabt wird.

Im nächsten Schritt erreicht der SW-CT die Endaufgabe, in der von diesem Vorgangsexemplar abhängige Workflow-Instanzen freigeschaltet werden (⑦). Dazu gehört in jedem Fall der zugehörige Hauptprozeß sowie der betrachtete HW-CT, kann aber auch noch weitere Change-Typ-Prozesse beinhalten.

Der HW-CT durchläuft nun die Hardware-Auslieferung, bevor er seine Endaufgabe erreicht, an der wiederum bestehende Interdependenzen freigeschaltet werden (⑧).

Damit kann nun auch der Hauptprozeß in die Change Nachbereitung eintreten, sofern nicht auf die Beendigung weiterer Kindprozesse gewartet werden muß (⑨).

6.2.3 Hintergrundverarbeitung zur Realisierung der benötigten Funktionalität

Um die in Kapitel 6.2.1 und 6.2.2 vorgestellte Funktionalität bereitzustellen, wurden mehrere Lotus Notes-Agenten in LotusScript realisiert, die die Bearbeitung automatisieren. Abschließend sollen die Komponenten dieser Hintergrundverarbeitung vorgestellt werden. Für technische Details sei auf das Administratorhandbuch der Projektdokumentation verwiesen [vgl. PAVONE 1998c, S. 15 ff.].

Starten von Prozessen

Auf Basis der im Beschaffungsantrag involvierten Geschäftsvorfälle wird ermittelt, welche Change-Typ-Prozesse zu starten sind. Die entsprechenden Vorgangsexemplare werden erzeugt und als Abhängigkeiten, auf die gewartet werden muß, in einem Feld vermerkt. Abschließend wird das Dokument zur nächsten Aufgabe weitergeleitet.

Bei der Erzeugung der Vorgangsexemplare müssen gewisse Daten aus dem Hauptprozeß an die Workflow-Instanzen übergeben werden, so daß diese Daten für die Durchführung zur Verfügung stehen (Content Management). Unterschieden werden muß hier zwischen Kerndaten, die der Agent automatisch in jeden Kindprozeß transferiert (z. B.

die Interdependenzen dieses Vorgangsexemplars), und Change-Typ-spezifischen Daten. Letztere können in der Espresso Vorlagendatenbank je Change-Typ in Einstellungsdocumenten spezifiziert werden, so daß für einen Transport zusätzlicher oder anderer Felder nicht der Agent, sondern nur dieses Einstellungsdocument geändert werden muß.

Interprozeßkommunikation und Synchronisation

Der Hauptprozeß übergibt den Change-Typen alle für sie relevanten Abhängigkeiten. Erreicht ein solcher Kindprozeß nun einen Synchronisationspunkt, so wird dort solange gewartet, bis die Workflow-Instanzen, von denen das betrachtete Vorgangsexemplar abhängig ist, dieses freischalten.

Dies geschieht, indem eine abhängige Workflow-Instanz in einem ihrer Felder eine Nachricht für das wartende Vorgangsexemplar einträgt, sobald ein Freischaltpunkt erreicht wird. Somit steht diese Nachricht für das wartende Vorgangsexemplar zur Verfügung, die freischaltende Workflow-Instanz kann aber ohne Beeinträchtigung weiterlaufen. Als Nachrichten werden zur Zeit „Completed“ für den Beendigung einer Workflow-Instanz und „HW-Vorbereitung abgeschlossen“ zur internen Synchronisation des Geschäftsvorfalles „Neuer Arbeitsplatz“ verwendet.

Das wartende Vorgangsexemplar prüft nun pro eingetragener Interdependenz, ob die abhängige Workflow-Instanz die vereinbarte Nachricht in deren Feld gesetzt hat. Ist dies der Fall, so wird die im wartenden Vorgangsexemplar gespeicherte Abhängigkeit in ein Feld übernommen, in dem erledigte Interdependenzen protokolliert werden. Die Abhängigkeit selbst wird aus dem Feld, das die aktuellen Interdependenzen enthält, entfernt. Somit kann das am Synchronisationspunkt wartende Vorgangsexemplar zur nächsten Aufgabe weitergeleitet werden, sobald dessen Feld mit den aktuellen Interdependenzen leer ist. Ist das Feld dagegen nicht leer, so wird gewartet, bis der bearbeitende Agent das nächste Mal die noch bestehenden Abhängigkeiten überprüft.

Fehlerbehandlung

Von besonderer Relevanz ist die Fehlerbehandlung der automatisierten Aufgaben, da Fehler am Server auftreten und somit keinem Benutzer zur weiteren Behandlung übergeben werden können. Pro Aufgabengebiet der Hintergrundverarbeitung (u. a. Starten von Prozessen, Benachrichtigung unter den Prozessen) werden daher jeweils drei Felder verwaltet: *ErrorCounter*, *Errors* und *Resets*.

Tritt ein Fehler auf, so wird der Wert in *ErrorCounter* inkrementiert und die zugehörige Fehlermeldung im Feld *Errors* angehängt. Kann die Aufgabenbearbeitung bei einem erneuten Versuch erfolgreich abgeschlossen werden, so wird der *ErrorCounter* gelöscht, um den Mechanismus später erneut anwenden zu können. Wird dagegen eine festgelegte maximale Fehleranzahl in einem Aufgabengebiet überschritten, so wird das Vorgangsexemplar suspendiert.

Eine spezielle Ansicht erlaubt dem Administrator, alle Workflow-Instanzen einzusehen, während deren Bearbeitung ein Fehler aufgetreten ist. Eine besondere Maske zeigt die aufgetretenen Fehler und ermöglicht es, das Feld *ErrorCounter* auf den Wert 0 zurückzusetzen, wodurch die Hintergrundverarbeitung erneut angestoßen wird. Das Zurücksetzen wird dabei im Feld *Resets* protokolliert, damit nachvollzogen werden kann, welche Person wann einen Reset angestoßen hat.

6.3 Die Anforderungsliste als zentrale Komponente

Zur Umsetzung eines elektronischen Change Management-Systems gehört die Berücksichtigung des bereits vorhandenen Formularwesens. In diesem Projekt handelt es sich dabei um den in Anlage C.1 abgebildeten Beschaffungsantrag. Weitere Formulare wurden in der Prototypimplementierung nicht berücksichtigt.

Die zentrale Komponente des Beschaffungsantrags bildet die Anforderungsliste, weshalb bei der elektronischen Umsetzung auf eine möglichst einfache Bereitstellung der benötigten Funktionalität geachtet wurde. Dieser Abschnitt soll zuerst die nötigen Architekturmerkmale vorstellen, um den Leser in die Problematik einzuführen. Anschließend werden die im Prototypen genutzten Implementierungskonzepte vorgestellt [vgl. PAVONE 1998b, S. 14 ff.].

6.3.1 Benötigte Architekturmerkmale

Die Anforderungsliste besteht primär aus zwei Spalten, in denen einer Menge eine Bezeichnung zugeordnet wird. Bei der Preisermittlung werden dann zwei Spalten mit einem Einzel- und einem Gesamtpreis ergänzt. Im weiteren Verlauf der Bearbeitung werden zusätzlich pro Position aufgenommen: der Geschäftsvorfall, der Change-Typ, eine Bemerkung, der voraussichtliche Beschaffungstermin und die für die Beschaffung zuständige Person. Durch diese sukzessive Anreicherung mit Informationen ist es zur

besseren Übersichtlichkeit des jeweiligen Bearbeiters notwendig, diesem nur die jeweils benötigten Daten aufgabenselektiv verfügbar zu machen.

Ferner setzt die Angabe des Geschäftsvorfalles und des daraus resultierenden Change-Typs eine Verwaltung aller in der Anforderung vorhandenen Geschäftsvorfälle in einer separaten Liste voraus. Diese Liste umfaßt pro Geschäftsvorfall die folgenden Daten: Name des Geschäftsvorfalles, Priorität, Empfängerinformationen (Namen, Aufstellungsorte, Bemerkung) sowie Abhängigkeiten von anderen Geschäftsvorfällen oder kompletten Change Requests.

Ein sehr wichtiges Architekturmerkmal ist der möglichst einfache Zugriff auf die Systems Management Portfolio-Datenbank. Produkte müssen inklusive aller wichtiger Daten für die weitere Bearbeitung insbesondere im Hinblick auf die spätere Zuordnung zu den Change-Typen aus der Portfolio-Datenbank in die Anforderungsliste übernehmbar und später aufgrund der Einträge in der Anforderungsliste in der Portfolio-Datenbank wieder auffindbar sein (vgl. Kapitel 4.3.7). Diese Anforderung ist nur durch eine strukturierte Ablage der Daten möglich, bei der es zudem notwendig ist, daß aus der Portfolio-Datenbank übernommene Bezeichnungen nicht mehr editiert werden können, um eine eindeutige Zuordnung zu gewährleisten.

Weiterhin macht die gewählte Prozeßarchitektur eine Separierung der Anforderungsliste notwendig, denn jedem Change-Typ-Prozeß soll nur der für ihn relevante Teil in Form einer eigenen Anforderungsliste übergeben werden. Dies vereinfacht die Bearbeitung in der Change Durchführung, da nicht in jeder Aufgabe erneut die zu bearbeitenden Positionen in der Anforderungsliste identifiziert werden müssen. Letzteres Vorgehen würde auch dazu führen, daß gewisse Positionen doppelt und andere Positionen gar nicht bearbeitet werden, da die Zuordnung zu den Change-Typen äußerst flexibel gehandhabt werden kann und somit nicht eindeutig ist.

Eine weitere zentrale Funktionalität ist die Erstellung einer Historie zur Sicherstellung der Revisionsfähigkeit. Zu unterschiedlichen Zeitpunkten sind in der Bearbeitung von gewissen Personen Unterschriften zu leisten (Antragsteller, Personalverantwortlicher, Kostenstellenleiter sowie vom Controlling). Diese müssen auch im elektronischen Beschaffungsantrag in Form elektronischer Unterschriften abgebildet werden. Da sich die Unterschriften in erster Linie auf die zu beschaffenden Komponenten beziehen, muß die Anforderungsliste daher zu diesen Zeitpunkten „eingefroren“ und digital signiert

werden. Ferner muß eine Übersicht bereitgestellt werden, die aufzeigt, wer wann welchen Stand der Anforderungsliste genehmigt hat.

6.3.2 Überblick über die technische Realisierung

Zur Implementierung der geschilderten Architekturmerkmale wurde eine dedizierte Maske in Lotus Notes bereitgestellt, bei der in einer Tabelle strukturiert die benötigten Felder zur Verfügung gestellt werden. Im Gegensatz zu anderen Implementierungsalternativen, wie z. B. als OLE-Objekt eingebettete Tabellenelemente, ermöglicht dies eine aufgabenselektive und zugriffskontrollierte Datenpräsentation, was eine übersichtliche Bearbeitung sowie eine spätere Wiederauffindbarkeit der aus der Systems Management Portfolio-Datenbank übernommenen Positionen gewährleistet.

Die zu lösende Problematik innerhalb der Maske bestand darin, daß die Anforderungsliste eine variable Anzahl an Positionen besitzt, in Lotus Notes jedoch eine vordefinierte Anzahl an Feldern bereitgestellt werden muß, da eine dynamische Felderzeugung zur Laufzeit nicht möglich ist. Als Lösung wurde ein Schichtenmodell eingeführt, das in Abbildung 6.3.2-1 dargestellt wird.

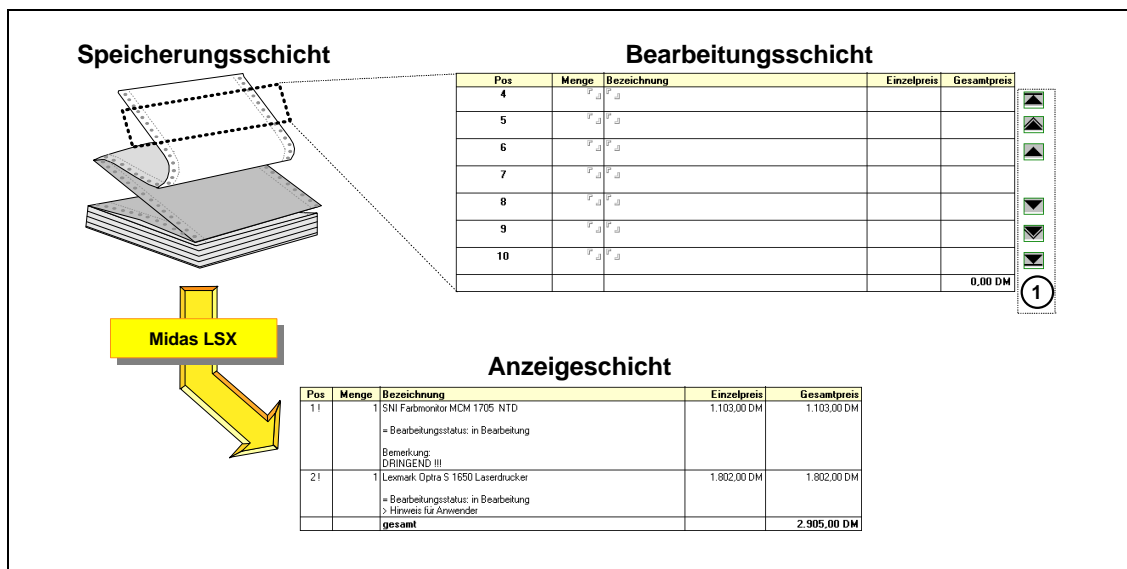


Abb. 6.3.2-1: Schichtenmodell zur Realisierung der Anforderungsliste

In der Speicherungsschicht wird die Anforderung in einer vom Entwickler frei wählbaren, nativen Lotus Notes-Datenstruktur verwaltet, was die Handhabung der Daten z. B. im Hinblick auf die Datenübernahme aus der Portfolio-Datenbank gegenüber der Verwendung eines externen Tabellenobjekts stark vereinfacht und beschleunigt.

Die Bearbeitungsschicht ermöglicht das Bearbeiten eines Teilausschnitts der Anforderungsliste in strukturierter Form. Der Teilausschnitt kann dabei durch entsprechende Navigationselemente (Bildlaufleiste - vgl. ① in Abbildung) selektiert werden. Damit wird die Darstellung von der internen Verwaltung entkoppelt, so daß die Speicherstruktur bei Bedarf geändert werden kann, ohne daß sich dies auf die Benutzerdarstellung auswirkt.

Da die Bearbeitungsschicht nur einen Teilausschnitt der Anforderungsliste bietet und sich zudem noch in einer dedizierten Maske befindet, wurde eine weitere Schicht, die Anzeigeschicht, eingeführt. In dieser wird die Anforderungsliste in eine Rich Text-Tabelle der benötigten Länge überführt und in der eigentlichen Maske zur Aufgabebearbeitung zur Anzeige gebracht. Zur Realisierung wurde an dieser Stelle das Midas Rich Text LSX genutzt.

Die Separierung der Anforderungsliste bei der Initiierung der Change-Typ-Prozesse geschieht automatisiert auf Basis der je Position zugeordneten Geschäftsvorfälle und Change-Typen. Für jeden Kindprozeß werden die zugeordneten Positionen aus der Anforderungsliste ermittelt und in eine neue Anforderungsliste für diesen Change-Typ übernommen. Die in der Portfolio-Datenbank definierten Beschaffungspakete (z. B. Beschaffungspaket „Standard-PC“) werden dabei in den einzelnen Bestandteilen in die Anforderungslisten der Change-Typen übertragen, während sie im Hauptprozeß noch als eine Position geführt werden. Abbildung 6.3.2-2 verdeutlicht diesen Vorgang.

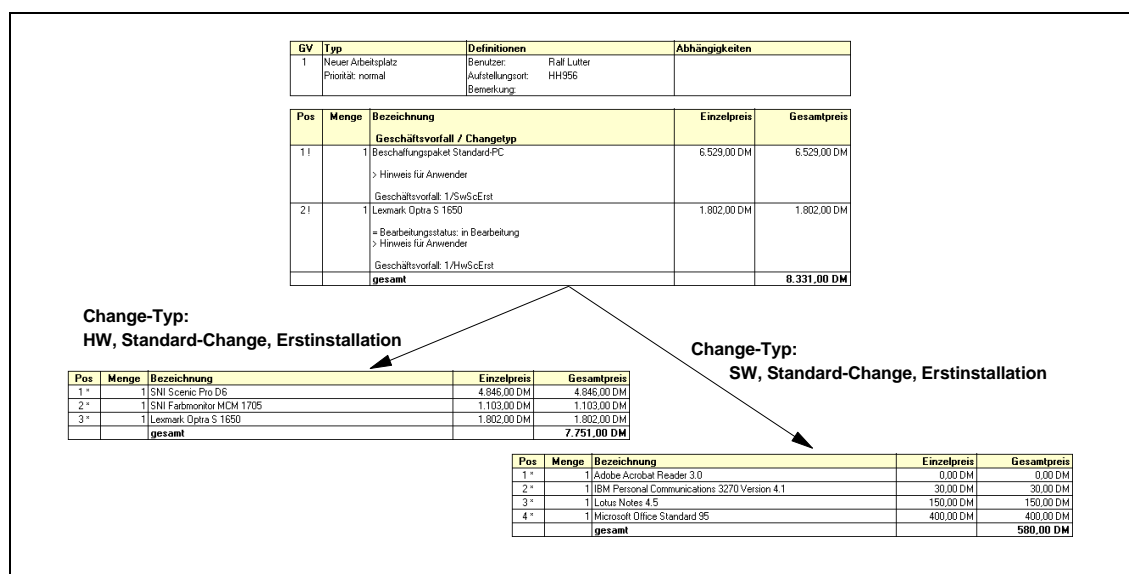


Abb. 6.3.2-2: Separierung der Anforderungsliste für die Change-Typ-Prozesse

Somit werden die Beschaffungspakete nicht von den Herstellern bestimmt, sondern können von der Deutschen Bundesbank selbst festgelegt werden. Um die Separierung zu ermöglichen, müssen die dafür nötigen Informationen bereits in der Portfolio-Datenbank spezifiziert und bei der Bestellung eines Beschaffungspaketes mit in die Speicherungsschicht der Anforderungsliste übernommen werden.

Die benötigte Historie wurde wiederum in einer dedizierten Maske realisiert, um die eigentliche Maske zur Aufgabebearbeitung in ihrer Darstellungskomplexität zu entlasten. Die einzelnen „eingefrorenen“ und signierten Anforderungslisten werden dabei im Vergleich zur aktuellen Anforderungsliste übersichtlich dargestellt. Sind zwischen den einzelnen zu protokollierenden Zeitpunkten keine Änderungen an der Anforderung vorgenommen worden, so wird dies durch die Bemerkung „keine Änderung durchgeführt“ kenntlich gemacht, um eine Suche nach nicht vorhandenen Unterschieden in den Listen zu vermeiden (siehe Abbildung 6.3.2-3).

▼ aktuell				
Pos	Menge	Bezeichnung	Einzelpreis	Gesamtpreis
1!	1	SNI Scenic Pro D6 = Bearbeitungsstatus: in Bearbeitung	4.900,00 DM	4.900,00 DM
2!	1	SNI Farbmonitor MCM 1705 = Bearbeitungsstatus: in Bearbeitung	1.599,00 DM	1.599,00 DM
gesamt				6.499,00 DM
▼ Original - Signed by Ingo Huebner/DV3/DV/504/BBK/DE on 03.02.98 13:26:54, according to /DV/504/BBK/DE				
Pos	Menge	Bezeichnung	Einzelpreis	Gesamtpreis
1!	1	SNI Scenic Pro D6 = Bearbeitungsstatus: in Bearbeitung	4.900,00 DM	4.900,00 DM
2!	1	Sony Multiscan 400PS = Bearbeitungsstatus: in Bearbeitung	2.499,00 DM	2.499,00 DM
gesamt				7.399,00 DM
▼ Personalverantwortlicher - Signed by Franz Weis/DV1/DV/504/BBK/DE on 03.02.98 14:32:50, according to /504/BBK/DE				
keine Änderung durchgeführt				
▶ <u>Kostenstellenleiter</u>				
▶ <u>Controlling</u>				

Abb. 6.3.2-3: Historie eines Beispiel-Beschaffungsantrages

6.4 Benutzerführung und -interaktion

„Ein Software-Produkt ist nur so gut, wie seine Benutzer damit umgehen können.“ [Lauter 1987, S. 1] Dies gilt auch für die vorgestellte prototypische Implementierung, denn die durch den Projektumfang bedingte Komplexität und Interdependenzen des Gesamtsystems sollen dem Anwender verborgen bleiben. Aus diesem Grund wurde bei der Implementierung auf eine möglichst einfache Bearbeitung der Vorgangstypen durch

die Anwender geachtet, um die Aufgabenbearbeitung zu beschleunigen, die Ergebnisqualität zu verbessern und die Akzeptanz des Systems sicherzustellen.

Im Rahmen dieses Unterkapitels soll zuerst das konzeptionelle Maskendesign als Strukturierung der Schnittstelle zur Informationsrepräsentation vorgestellt werden. Anschließend wird anhand ausgewählter Aspekte der allgemeinen Benutzerführung die einfache Bedienbarkeit des Systems belegt, bevor die Benutzerinteraktion in der Systems Management Portfolio-Datenbank vorgestellt wird. Abschließend wird eine Bewertung auf Basis der in Kapitel 3.3 vorgestellten DIN/EN-Kriterien vorgenommen.

6.4.1 Konzeptionelles Maskendesign

Innerhalb des Prototypen wurden insgesamt 7 Masken und 46 Teilmasken umgesetzt. Für die eigentliche Aufgabenbearbeitung werden dabei lediglich zwei Masken benötigt: eine Maske für den Hauptprozeß und eine Maske für alle Change-Typ-Prozesse. In diesen beiden Masken wird dann jeweils eine aufgabenspezifische Teilmaske eingebunden, die alle für diese Aufgabe benötigten Felder enthält (siehe Abbildung 6.4.1-1).

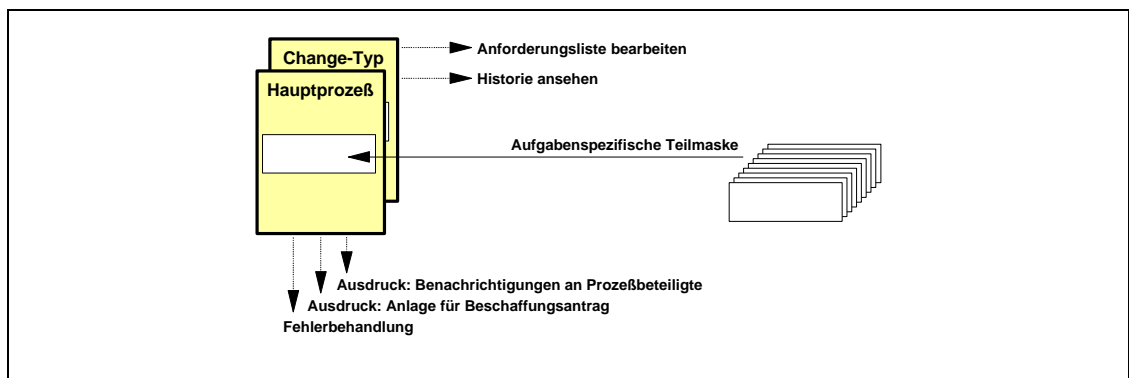


Abb. 6.4.1-1: Zusammenspiel der implementierten Masken und Teilmasken

Bestimmt wird die Teilmaske auf Basis eines im ProcessModeler festgelegten Alternativnamens für jede Aufgabe, so daß der eigentliche Name der Aufgabe verändert werden kann, ohne daß die Auswahl der richtigen Teilmaske gefährdet wird. Diese Vorgehensweise wurde im Projektrahmen neu spezifiziert und ist in das Standardprodukt PAVONE Espresso eingeflossen.

Die konsistente Benutzung nur einer Maske über alle Aufgaben eines Vorgangstypen hinweg hat den Vorteil, daß aufgabenübergreifende Funktionalitäten in der Maske bereitgestellt und sehr einfach geändert werden können, ohne daß pro Aufgabe je eine Maske angepaßt werden muß.

Die weiteren Masken stellen Hilfsmasken zur Bearbeitung der Anforderungsliste, zur Anzeige der Historie, zur Fehlerbehandlung, zum Ausdruck eines Anhangs zum Beschaffungsantrag für die Genehmigung durch den Kostenstellenleiter oder zum Ausdruck von Benachrichtigungen an Prozeßbeteiligte dar.

Im Folgenden soll das konzeptionelle Maskendesign des Hauptprozesses beispielhaft vorgestellt werden. Abbildung 6.4.1-2 zeigt die Maskenstruktur.

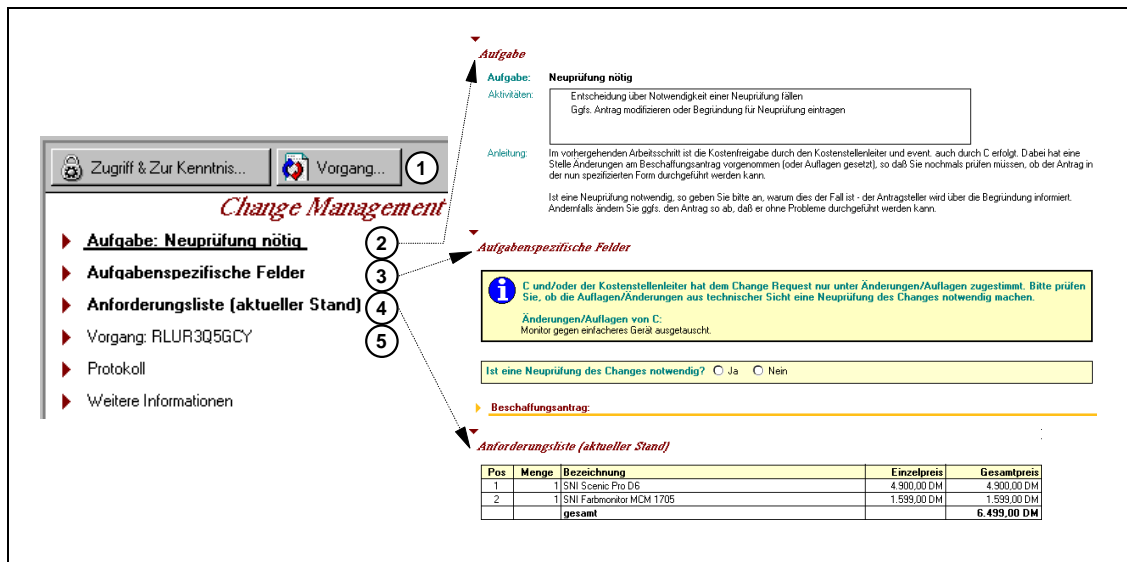


Abb. 6.4.1-2: Konzeptionelles Maskendesign für den Hauptprozeß

Die Aktionsleiste (①) beinhaltet Schaltflächen für die jeweils zu diesem Zeitpunkt möglichen Aktionen. Dies bedeutet, daß sich die Zusammensetzung je nach Bearbeitungsstatus (z. B. Anforderungsliste änderbar oder nicht) und Bearbeitungsmodus (Lese- oder Schreibmodus) ändern kann.

Die Aufgabensektion (②) gibt an, welche Aufgabe gerade bearbeitet wird (eine Sektion in Lotus Notes ermöglicht es, einen Bereich eines Dokuments normal anzuzeigen oder lediglich als Sektionsüberschriftenzeile darzustellen). Standardmäßig ist die Sektion geschlossen, jedoch kann sie bei Bedarf vom Bearbeiter geöffnet werden, woraufhin der Aufgabenname, die Aktivitäten der Aufgabe sowie eine Anleitung zur Bearbeitung angezeigt werden.

Die standardmäßig geöffnete Sektion „Aufgabenspezifische Felder“ (③) enthält die aufgabenspezifische Teilmaske wie auch den Beschaffungsantrag am Ende der Sektion. Dieser wird aus Gründen der Übersichtlichkeit in einer eigenständigen, als Vorgabe geschlossenen Sektion innerhalb der beschriebenen Sektion verwaltet.

Der aktuelle Stand der Anforderungsliste wird wegen seiner Bedeutung im Rahmen der Bearbeitung des Change Requests in einer separaten, standardmäßig geöffneten Sektion dargestellt (④). Je nach Bearbeitungsfortschritt innerhalb des Hauptprozesses wird dort auch die Aufteilung in Geschäftsvorfälle und Change-Typen angezeigt.

Am Schluß der Maske werden die weniger relevanten Sektionen mit Vorgangseinstellungen, dem Protokoll und weiteren Informationen zum Vorgang aufgeführt (⑤).

Die Maske für alle Change-Typen besitzt vergleichbare Merkmale, jedoch entfällt die separate Sektion für die Anforderungsliste, die statt dessen innerhalb der Sektion „Aufgabenspezifische Felder“ angezeigt wird, da dort der Beschaffungsantrag nicht mehr enthalten ist. Dieser befindet sich nur im Elterndokument des Change Management-Hauptprozesses, so daß eine Schaltfläche „Beschaffungsdokument“ in der Aktionsleiste zur Verfügung gestellt wird, die die nicht mehr benötigte Aktion „Historie“ ersetzt, um das Elterndokument so einfach erreichen zu können.

6.4.2 Ausgewählte Aspekte der allgemeinen Benutzerführung

Ein zentrales Ziel der Prototypenumsetzung ist die einfache Aufgabenbearbeitung durch die Prozeßbeteiligten, weshalb zunächst ein möglichst einfaches Navigationskonzept benötigt wird. Dieses basiert auf der Standardnavigation von PAVONE Espresso, wurde jedoch projektspezifisch angepaßt und wird in Abbildung 6.4.2-1 dargestellt.

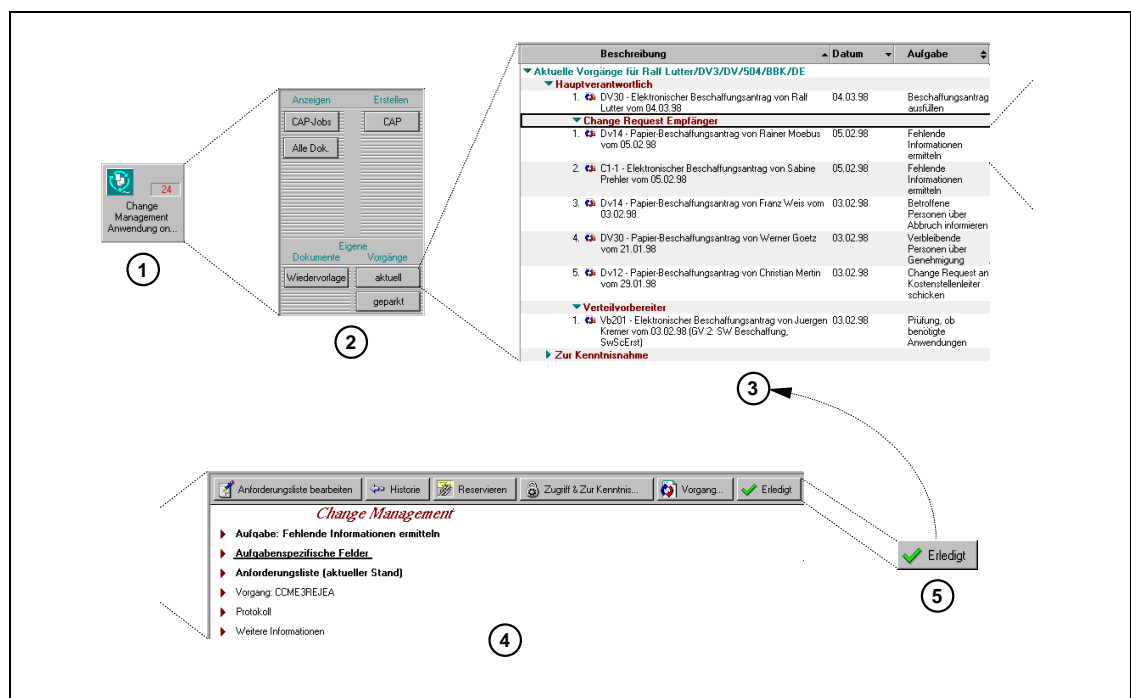


Abb. 6.4.2-1: Navigationskonzept des implementierten Prototypen

In Schritt ① wählt der Benutzer die Anwendungsdatenbank in der Lotus Notes-Arbeitsoberfläche aus und öffnet die Datenbank. Der erscheinende Navigator (②) gibt einen Überblick über die wichtigsten Ansichten der Datenbank und ermöglicht das Anlegen neuer Change Requests.

Als zentrale Ansicht steht dem Endanwender die private Ansicht „Eigene Vorgänge - aktuell“ zur Verfügung (③), die rechts im Fenster neben dem Navigator alle zu bearbeitenden Vorgangsexemplare rollenbasiert aufführt, so daß der Bearbeiter sofort erkennen kann, in welcher Funktion die jeweilige Aufgabe auszuführen ist.

Wählt der Benutzer ein Dokument aus, so kann er dieses bearbeiten (④). Ist die Aufgabenbearbeitung abgeschlossen, so betätigt der Anwender die Schaltfläche „Erledigt“ in der Aktionsleiste (⑤): das Dokument wird nach einer Eingabvalidierung automatisch an die nächste(n) Aufgabe(n) weitergeleitet, und der Benutzer kehrt zu seiner gewählten Ansicht zurück (③) und kann dort die nächste Workflow-Instanz auswählen.

Ergänzt wird das bewußt einfach gehaltene Navigationskonzept durch ein intuitives, aufgabenbezogenes Maskendesign, welches grundlegende Konzepte über alle Aufgaben hinweg bereitstellt, so daß die Endanwender keine aufgabenspezifische Einarbeitung in die Anwendung benötigen. Das in Kapitel 6.4.1 vorgestellte konzeptionelle Maskendesign stellt dabei die Grundlage für die eigentliche Maskengestaltung dar, deren zentrale Aspekte im Folgenden vorgestellt werden sollen.

Die Aufgabensektion enthält eine genaue Anleitung darüber, was in der jeweiligen Aufgabe zu tun ist (vgl. Abbildung 6.4.1-2). Die Sektion gibt auch einen Überblick darüber, weshalb das Vorgangsexemplar zu bearbeiten ist (woher es kommt) und was nach der Bearbeitung geschieht (Weiterleitungsmöglichkeiten werden beschrieben). Die Aktivitätenliste unterstützt den Bearbeiter dahingehend, daß er bereits ausgeführte Aktivitäten als solche kennzeichnen kann (Aktivitäten können abgehakt werden).

Da die Aufgabensektion standardmäßig geschlossen ist und die Anleitung nur statische Informationen zur Aufgabenbearbeitung enthält, wurde am Anfang jeder Aufgabe eine Informationsbox eingefügt, die alle nötigen Hinweise zum Bearbeitungsstand der zu erledigenden Aufgabe enthält und deren Inhalt sich dynamisch an die jeweilige Situation anpaßt (vgl. Schritt ① in Abbildung 6.4.2-2). Der erste Schritt zur Bearbeitung einer Aufgabe besteht somit in der Information über den Bearbeitungsstand.

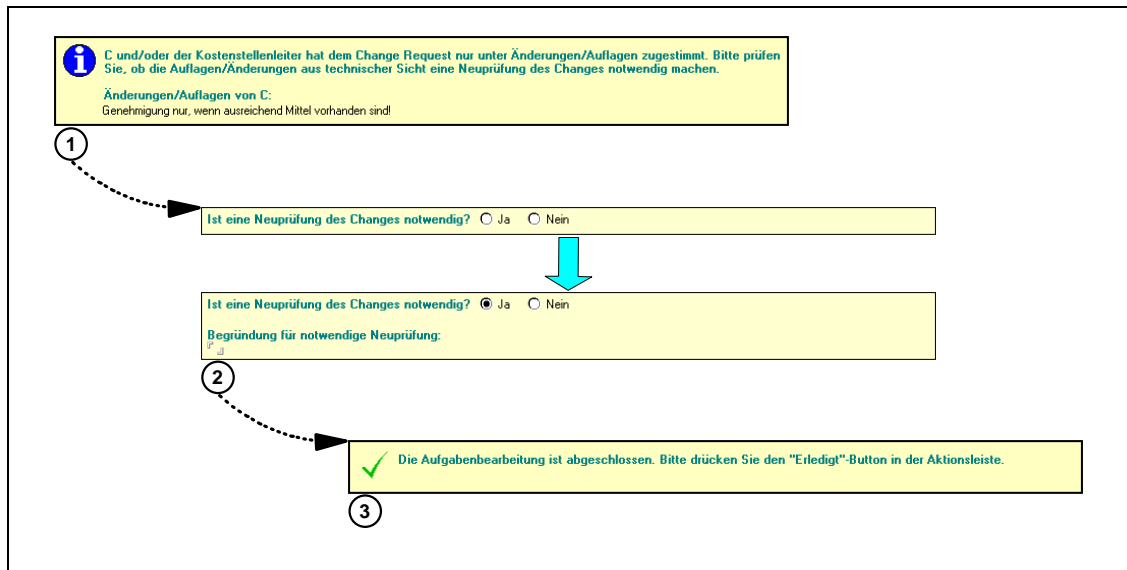


Abb. 6.4.2-2: Bearbeitungskonzept des implementierten Prototypen

Schritt ② im Bearbeitungskonzept ist dann die Bearbeitung der Aufgabe auf Basis der angezeigten Felder und Informationstexte. Diese wurden dem Sinn nach in Tabellen und ggf. Sektionen gruppiert, um dem Bearbeiter aus Gründen der Übersichtlichkeit ein Schließen der Sektionen zu ermöglichen. Dadurch soll dem Endanwender ein klar gegliedertes und übersichtliches Dokument zur Verfügung gestellt werden, was selbst gelegentlichen Bearbeitern die Nutzung des Systems ohne großen Einarbeitungsaufwand erlaubt.

Um interdependente Aktivitätenabfolgen innerhalb einer Aufgabe zu realisieren, werden die entsprechenden Felder nicht nur gruppiert, sondern diese werden auch erst nach Abschluß der vorhergehenden Aktivität eingeblendet. Ermöglicht wird dies durch die Nutzung einer Standardfunktionalität der Lotus Notes-Entwicklungsumgebung zum differenzierten Ausblenden von Gestaltungselementen und anderen Attributen. Die gleiche Funktionalität wird auch genutzt, um Masken kontextsensitiv anzupassen (z. B. situationsbedingte Informationstexte in Schritt ①, Anzeige der Felder je nach Bearbeitungsstatus auch innerhalb einer Aktivität).

Durch die Abhängigkeiten in der Aktivitätenabfolge kann auch der Zeitpunkt ermittelt werden, an dem die Aufgabenbearbeitung abgeschlossen ist. Der Bearbeiter wird dann durch eine weitere Hinweisbox über diesen Umstand informiert und aufgefordert, das Vorgangsexemplar durch das Drücken der Schaltfläche „Erledigt“ in der Aktionsleiste an die nächste(n) Aufgabe(n) weiterzuleiten (③). Kann aufgrund fehlender Interdependenzen zwischen den Aktivitäten der Aufgabenabschluß nicht ermittelt werden, so wird

anderweitig auf die Schaltfläche hingewiesen (zumeist bereits innerhalb der ersten Informationsbox).

Um den Eingabeaufwand in den Masken zu reduzieren, werden Felder weitestgehend mit sinnvollen Vorgaben vorbelegt und bestehende Datenpools zur Ermittlung von Daten herangezogen. So können über Feldern zugeordneten Schaltflächen Namen aus dem Lotus Notes Namens- und Adreßbuch ausgewählt werden, wobei alle zugehörigen Merkmale wie Telefonnummer etc. aus den vorhandenen Repositories extrahiert und in die zugeordneten Felder eingetragen werden.

6.4.3 Benutzerinteraktion in der Portfolio-Datenbank

Die Systems Management Portfolio-Datenbank folgt als eigenständige Datenbank-anwendung in ihrer Interaktion dem Paradigma des Electronic Shopping, bei dem Waren in einem elektronischen Warenkorb abgelegt werden können, um sie später in ihrer Gesamtheit bezahlen zu können. Abbildung 6.4.3-1 und die folgende Ausführung verdeutlichen dieses Vorgehen am konkreten Beispiel der Portfolio-Datenbank (vgl. [PAVONE 1998d]; [PAVONE 1998e]).

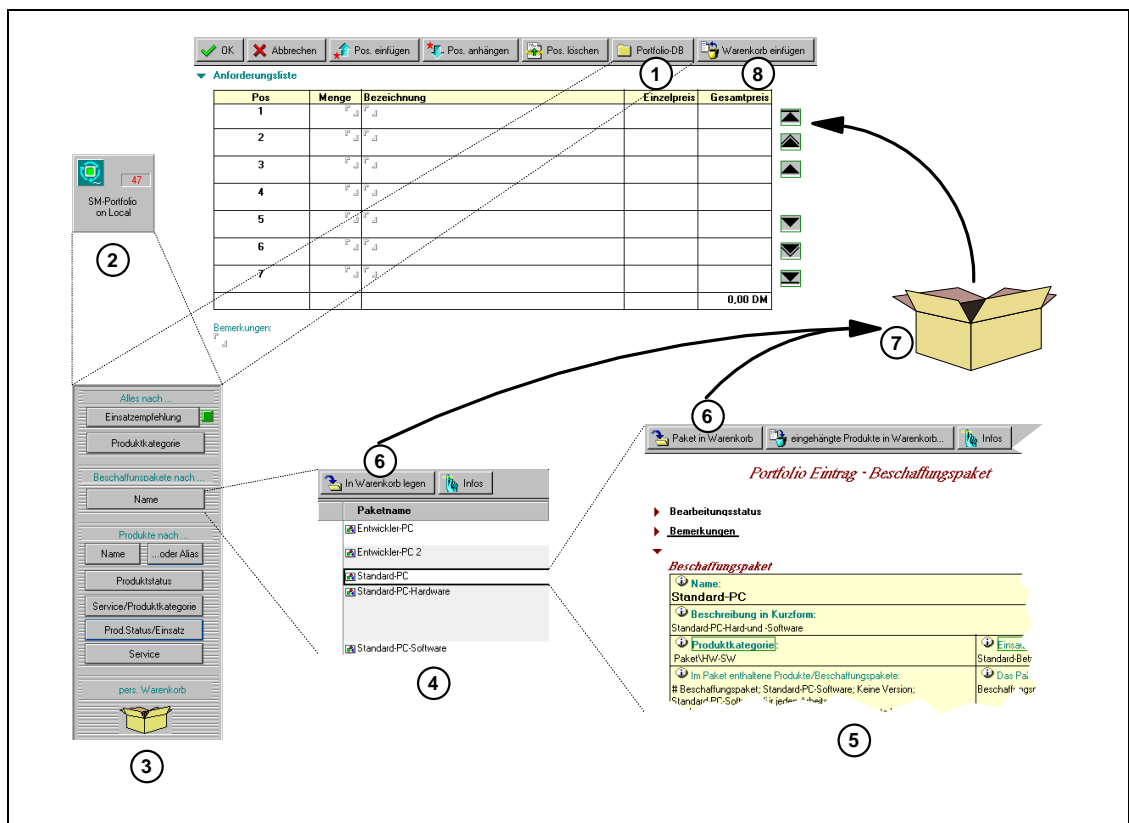


Abb. 6.4.3-1: Paradigma des Electronic Shopping in der Portfolio-Datenbank

Nach dem Öffnen der Datenbank über eine Schaltfläche in der Maske zur Bearbeitung der Anforderungsliste (①) oder von der Lotus Notes-Arbeitsoberfläche aus (②), können Produkte und Beschaffungspakete mit Hilfe des Navigators (③) und vorhandener Ansichten (④) unterschiedlich sortiert angezeigt, selektiert und betrachtet werden (⑤). Bei Interesse werden die ausgewählten Elemente über eine Schaltfläche in der Aktionsleiste des Dokuments oder der gewählten Ansicht (⑥) in einen persönlichen Warenkorb (⑦) gestellt, nachdem eine kurze Zusammenfassung angezeigt wurde (Daten des Produkts, Servicekategorie etc.). Im Warenkorb enthaltenen Produkte und Beschaffungspakete verbleiben dort, bis sie aktiv entfernt werden oder bis der Warenkorb über eine weitere Schaltfläche in der Maske zur Bearbeitung der Anforderungsliste (⑧) in eben diese übernommen wird.

Somit verbleiben die Dokumente auch im Warenkorb, wenn die Datenbank verlassen wird. Dies ermöglicht dem Antragsteller, frühzeitig nach den benötigten Komponenten zu suchen und diese im privaten Warenkorb einzustellen. Der eigentliche Antrag kann dann zu einem beliebigen, späteren Zeitpunkt gestellt werden.

6.4.4 Bewertung auf Basis der Grundsätze zur Dialoggestaltung

Abschließend soll eine Untersuchung der Benutzerinteraktion in Bezug auf die in Kapitel 3.3 vorgestellten Grundsätze der Dialoggestaltung vorgenommen werden:

- *Aufgabenangemessenheit:* Der Beachtung dieses Grundsatzes wurde insgesamt ein sehr hoher Stellenwert zugemessen, so daß hier nur eine Auswahl an beispielhaften Merkmalen angegeben werden kann. Diese umfassen u. a. die einfache Navigation, die selektive Anzeige nur arbeitsrelevanter Sachverhalte, die kontext-sensitive Hilfe zur Arbeitsaufgabe, die Automatisierung der für den Prototypen notwendigen Aufgaben ohne Bezug zur Arbeitsaufgabe und die Vorgabe ersetzbarer Standardwerte.
- *Selbstbeschreibungsfähigkeit:* Für dieses Kriterium wurde auf eine intuitive Oberflächengestaltung in Verbindung mit ausreichenden, auf den jeweiligen Benutzer und die Aufgabe zugeschnittenen und den Gesamtzusammenhang verdeutlichenden Hilfetexten geachtet. Dies beginnt beim Feldtitel und reicht über die Feldhilfe und den beschriebenen Informationsboxen bis hin zur Aufgabenanleitung.
- *Steuerbarkeit:* Dieser Grundsatz wird hauptsächlich von der Groupware-Plattform Lotus Notes bzw. von PAVONE Espresso abgedeckt. So bietet Notes die Mög-

lichkeit, die letzte Aktion rückgängig zu machen. Espresso ermöglicht die Arbeitsunterbrechung/-wiederaufnahme an einem beliebigen Punkt und leitet das Dokument erst an den nächsten Bearbeiter weiter, wenn dieser diese Aktion aufruft. Im Projekt wurde durch die Bildung von Sektionen darauf geachtet, daß der Benutzer die Menge der angezeigten Daten steuern kann.

- *Erwartungskonformität:* Die einheitliche Informationsdarstellung wird auf Basis des konzeptionellen Maskendesigns und das einheitliche Dialogverhalten durch die allgemeine Benutzerführung sichergestellt und im Benutzerhandbuch beschrieben. Da die vorhandenen Arbeitsabläufe lediglich elektronisch nachgebildet wurden und Lotus Notes als Unternehmensstandard eingesetzt wird, entspricht auch das neue System den bisherigen Kenntnissen der Anwender.
- *Fehlerrobustheit:* Bei der Gestaltung des Prototypen wurde Wert darauf gelegt, daß fehlerhafte Eingaben auf Basis umfangreicher Validierungsformeln vermieden werden und daß entsprechende Fehlermeldungen verständlich und sachlich konstruktiv formuliert sowie einheitlich strukturiert sind.
- *Individualisierbarkeit:* Hierbei handelt es sich wohl um die am schwersten umsetzbare Forderung im Rahmen dieses Projekts. Grundsätzliche Funktionen werden daher vom Betriebssystem bereitgestellt (Mauseinstellungen, Schriftgröße, Farbkombinationen usw.). Lotus Notes bietet die Möglichkeit, eigene Funktionen in Form von privaten Agenten zu definieren, was jedoch entsprechende Rechte und technisches Wissen voraussetzt. Bei der prototypischen Implementierung wurde auf die Bereitstellung von Hilfetexten in unterschiedlichem Detaillierungsgrad geachtet.
- *Lernförderlichkeit:* Diese Anforderung wurde dahingehend beachtet, daß die allgemeine Benutzerführung konsistent durch die gesamte Anwendung durchgehalten wird und die Hilfe- und Hinweistexte auch dem ungeübten Benutzer ausreichende Hinweise zur korrekten Bearbeitung der jeweiligen Aufgabe geben. Ein entsprechender Schulungsmodus wurde nicht implementiert.

Die vorgenommene Einschätzung wird gestützt durch die nahezu durchgängig positiven Bewertungen zur Systembedienbarkeit im Rahmen der vier eintägigen Endanwenderschulungen. Eine statistische Auswertung der Beurteilungen kann der Anlage C.2 entnommen werden.

7 Fazit und Perspektiven

Im Rahmen dieses Kapitels soll eine abschließende Betrachtung der Arbeit durchgeführt sowie konzeptionelle Weiterentwicklungsmöglichkeiten aufgezeigt werden.

Ziel dieser Arbeit war die Einordnung der Themenstellung in einen theoretischen und praktischen Bezugs- und Relevanzrahmen sowie die Vorstellung der zentralen Implementierungsaspekte des Prototypen zur Abbildung der administrativen Geschäftsprozesse im Change Management.

Fazit

Im vorliegenden schriftlichen Teil wurde die Bedeutung des Systems Management durch die erhöhte Komplexität im Client-/Server-Umfeld herausgestellt. Die Einordnung des Change Managements in die verschiedenen Systems Management-Ansätze zeigt dabei eine starke Interdependenz zwischen den einzelnen Disziplinen. Als zentrale Komponente zur Steuerung von Veränderungen in der IT-Infrastruktur wird das Change Management daher im Verbund mit der geforderten Geschäftsprozeßorientierung als Erfolgsfaktor der betrieblichen Datenverarbeitung aufgefaßt.

Zusätzlich zum theoretischen Rahmen der Projektimplementierung wurden Anforderungen an Workflow Management-Systeme vorgestellt und das eingesetzte System auf die Umsetzung hin untersucht. Ferner wurden Ergonomie-Anforderungen im Rahmen von Workflow Management-Systemen diskutiert und der realisierte Prototyp auf dieser Basis bewertet.

Der implementierte Prototyp stellt eine innovative Lösung im Bereich des Systems Management dar, die allen Projektanforderungen gerecht wird. Das modulare Konzept ermöglicht eine beliebige Skalierbarkeit im Einsatzbereich der IT-Infrastruktur, und der geforderte Geschäftsprozeß-orientierte Ansatz bietet eine systematische Verzahnung der kompletten Wertschöpfungskette zur Erschließung von Effizienzpotentialen.

Perspektiven: Allgemeine Trends

Den hohen Stellenwert des Systems Management bestätigen die Markteinschätzungen, die von einem jährlichen Wachstum von 30 Prozent ausgehen. So erwartet die International Data Corporation (IDC) für das Jahr 2000 ein weltweites Marktvolumen von 12,7 Milliarden Dollar [vgl. Beuthner 1998, S. 17].

Immer größere Bedeutung kommt dabei der Einbindung von Internet-Technologien zu [vgl. Born 1998, S. 13; o. V. 1998b, S. 10]. „Vielleicht helfen solche Technologien bei der dramatischen Vereinfachung des unternehmensweiten Systemmanagements sowie der Verbindung dessen mit den Geschäftsprozessen.“ [Born 1998, S. 13] Lotus Notes bietet mit seiner Internet-Ausrichtung daher die ideale Plattform für den implementierten Prototypen.

Perspektiven: Weiterentwicklungspotentiale der Lösung

Durch den modularen Aufbau des Prototypen ist dieser in der Lage, zukünftige Erweiterungen in Form weiterer Geschäftsvorfälle wie z. B. Umzug oder Außerbetriebnahme sowie der Unterstützung des Mainframe-Bereichs zu integrieren. Auch kann der Portfolio-Erweiterungsprozeß eigenständig modelliert und automatisch mit dem Change-Prozeß über die vorhandene Interprozeßkommunikation synchronisiert werden.

Die Architektur des Prototypen kann durch eine Öffnung hin zu bestehenden Systemen wie Tivoli, SAP und dem Bestellwesen OMEGA einfach erweitert werden, um den Datenaustausch mit anderen Disziplinen des Systems Managements und den Geschäftsprozeß-Anwendungen zu gewährleisten. Insgesamt sollte eine Integration in ein Systems Management-Framework in Erwägung gezogen werden. Die Einbindung eines Projekt Management-Werkzeugs kann den Anwendungsbereich von Endbenutzer-getriebenen Veränderungen hin zu geplanten Veränderungen erweitern, was auch eine Ressourcenplanung für die DV-Abteilung ermöglichen würde. Aus praktischer Sicht ist eine Verschmelzung der beiden eingesetzten Organisationsdatenbanken anzustreben.

Eine Erweiterung des Funktionsumfangs kann aus administrativer Sicht in einer graphischen Auswertung bearbeiteter Changes mit dem Ziel der Prozeßverbesserung sowie in einer besseren Unterstützung des Prozeßverantwortlichen im Rahmen der Interprozeßkommunikation angestrebt werden (z. B. müssen bei einem Abbruch in der Change Durchführung alle betroffenen Prozesse von Hand abgebrochen werden). Auch sollte eine vertikale Eskalationsregelung vorgesehen werden. Im operativen Umfeld könnten die Bearbeiter in Zukunft durch „Best Practise“-Studien unterstützt werden, um die Prozeßqualität zu erhöhen.

A Verzeichnisse

A.1 Literatur

[Arnold 1993]

Arnold, U.: Anforderungen an das Netzmanagement ändern sich immer schneller, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 48/1993, S. 14

[Becker/Vossen 1996]

Becker, J.; Vossen, G.: Geschäftsprozeßmodellierung und Workflow Management: Eine Einführung, in: Vossen, G.; Becker, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow Management: Modelle, Methoden, Werkzeuge, International Thomson Publishing, Bonn usw., 1996, S. 17-26

[Beuthner 1998]

Beuthner, A.: Systemmanagement wird zu einem Bombengeschäft, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 1+2/1998, S. 17

[Born 1998]

Born, A.: Das Systemmanagement erfüllt Erwartungen nicht, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 10/1998, S. 13

[Dehn 1995]

Dehn, D.: (Ohne Titel), in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 11/1995, Sonderteil 25 Jahre Computer Zeitung, S. 17

[DIN/EN 29241]

Deutsches Institut für Normung e.V.: DIN/EN 29241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten Teil 10: Grundsätze der Dialoggestaltung - Entwurf Januar 1994, Beuth Verlag, Berlin usw., 1994, aus: http://hyperg.uni-paderborn.de/0x83ea6001_0x00020703 am: 13.03.98

[Duden 1996]

Wermke, M. [Red.]: Duden Deutsches Universalwörterbuch, 3. Auflage, Dudenverlag, Mannheim usw., 1996

[Elgass/Krcmar 1994]

Elgass, P.; Krcmar, H.: Computerunterstützung in Geschäftsprozessen, in: Hasenkamp, U.; Kirn, S.; Syring, M. (Hrsg.): CSCW - Computer Supported Cooperative Work, Addison-Wesley, Bonn usw., 1994, S. 67-83

[Gartner Group 1997]

Gartner Group: Workgroup and Intranet Computing: Cost of Ownership Study - Management Summary, San Jose, 1997, aus:
<http://www.gartnerweb.com/public/static/home/tco13098.html> am: 18.03.98

[Genii 1998]

Genii Software: Overview of the Midas Rich Text LSX, Shaker Heights, 1998, aus: <http://members.aol.com/GeniiSoft/midas.htm> am: 17.02.98

[Gulla 1996]

Gulla, J.G.: C/S Disciplines and Tools, o. O., 1996, aus:
<http://www.scis.nova.edu/~gullaj/dsb.html> am: 26.02.98

[Gursky/Thole 1996]

Gursky, S.; Thole, S.: Die Kostensenkung um ein Drittel ist kein Hexenwerk, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 20/1996, S. 21

[Hammerschmidt 1998]

Hammerschmidt, C.: Server-Konsolidierung räumt mit dem IT-Dschungel im Unternehmen auf, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 11/1998, S. 32

[Hasenkamp 1987]

Hasenkamp, U.: Konzipierung eines Bürovorgangssystems, Habilitationsschrift, Köln, 1987

[Heilmann 1994]

Heilmann, H.: Workflow Management: Integration von Organisation und Informationsverarbeitung, in: HMD, Theorie und Praxis der Wirtschaftsinformatik, Schwerpunktthema „Workflow Management“, Forkel Verlag, o. O., Heft 176, März, 1994, S. 8-21

[Hilpert 1994]

Hilpert, W.: Effiziente und wirtschaftliche Einsatzplanung von Workflow Management-Systemen auf Groupware-Basis - vom Pilotprojekt zur unternehmensweiten Lösung, Arbeitspapier, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2 und PAVONE Informationssysteme GmbH, Paderborn, 1994

[Holzmann 1998]

Holzmann, E.: Billig kann teuer werden, in: Office Management, Verlagsgruppe Handelsblatt, Düsseldorf, Heft 1, Januar, 1998, S. 52-53

[IBM 1996]

IBM Corporation: Lotus Notes Release 4.5: A Developer's Handbook, IBM Corporation - International Technical Support Organization, Research Triangle Park, 1996

[IBM 1997a]

IBM Unternehmensberatung GmbH: Bereitstellung eines Change-Administration Prototyps für den Client-/Server-Bereich - Kick-Off, Ehningen, 1997

[IBM 1997b]

IBM Unternehmensberatung GmbH: Arbeitspapier zur Angebotseinholung bei der PAVONE Informationssysteme GmbH, o. O., 1997

[IBM 1998a]

IBM Unternehmensberatung GmbH: Bereitstellung eines Change-Administration Prototyps für den Client-/Server-Bereich - Abschlußreview, Ehningen, 1998

[IBM 1998b]

IBM Unternehmensberatung GmbH: Anwenderschulung Change Administration Prototyp, Frankfurt am Main, 1998

[IBM 1998c]

IBM Unternehmensberatung GmbH: Rollen und Schulungsplanung, Stellvertreter- und Informationsregelung, Ehningen, 1998

[IBM 1998d]

IBM Unternehmensberatung GmbH: Arbeitspapier zur Bewertung des Change Management Prozesses, o. O., 1998

[IBM 1998e]

IBM Unternehmensberatung GmbH: Angebot zur Erstellung einer Systems Management Portfolio-Datenbank (SM-DB) und Begleitung der Produktionseinführung des Change Management-Systems, Ehningen, 1998

[IBM 1998f]

IBM Unternehmensberatung GmbH: Erstellung einer Systems Management Portfolio-Datenbank - Pflichtenheft zur Implementierung, Ehningen, 1998

[Kauffels 1995]

Kauffels, F.-J.: Netzwerk- und System-Management: Probleme-Standards-Strategien, DATACOM-Buchverlag, Bergheim, 1995

[Keil-Slawik 1995]

Keil-Slawik, R.: Foliensatz zur Vorlesung "Grundlagen der Systemgestaltung" Wintersemester 1995/96, Paderborn, 1995

[Kickuth 1996]

Kickuth, R.H.: E-mail und Workflow Management-Systeme für Top-Manager wichtig, in: Informatik Spektrum, Springer Verlag, Berlin/Heidelberg, Band 19, Heft 3, Juni, 1996, S. 161-163

[Koll 1998]

Koll, S.: Auch ohne NC lassen sich die Kosten der Desktop-Landschaft im Zaum halten, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 11/1998, S. 34

[Langsford/Moffett 1993]

Langsford, A.; Moffett, J.D.: Distributed Systems Management, Addison-Wesley Publishing Company, Wokingham usw., 1993

[Lauter 1987]

Lauter, B.: Software-Ergonomie in der Praxis: Software anwenderfreundlich schreiben, Oldenbourg Verlag, München/Wien, 1987

[Lotus 1996]

Lotus Development Corporation: White Paper - Lotus Notes: An Overview, o. O., 1996, aus:
<http://www.lotus.com/core/content.nsf/060af40aab61ec2852564f2005f4ff4/b1ab39322df48dea852564f2005f7dc7?OpenDocument> am: 27.02.98

[Magura 1995]

Magura, S.: Helpdesk-Outsourcing sollte nur unter Anwenderaufsicht stattfinden, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 23/1995, S. 20

[Matzer 1997]

Matzer, M.: Integrierte Suiten ebnen den Weg durch komplexe Rechnerumgebungen, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 36/1997, S. 16

[Nastansky et al. 1994]

Nastansky, L.; Schicker, T.; Behrens, O. M.; Ehlers, P.: Büroinformationssysteme, in: Fischer, J.; Herold W.; Dangelmaier, W.; Nastansky, L.; Wolff, R.: Bausteine der Wirtschaftsinformatik, S + W Steuer- und Wirtschaftsverlag, Hamburg, 1994, S. 273-373

[Nastansky/Hilpert 1994]

Nastansky, L.; Hilpert, W.: The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation, in: Wolfinger, Bernd (Ed.): Innovationen bei Rechen- und Kommunikationssystemen - Eine Herausforderung an die Informatik, Proceedings of 24th Annual Conference of the German Computer Society during 13th World Computer Congress, IFIP '94, Springer Verlag, Berlin usw., 1994, S. 473-479

[Newbold/Lipton 1997]

Newbold, D.; Lipton, R.: Notes: A sustainable platform architecture, o. O., 1997, aus:
<http://notes.net/today.nsf/cbb328e5c12843a9852563dc006721c7/7ab2e0cdf9755d9685256522005a576f?OpenDocument> am: 27.02.98

[Ott 1996]

Ott, M.: Groupware - Charakterisierung und technologische Perspektive, Workgroup Computing Competence Center, Paderborn, 1996

[Ottomeier 1996]

Ottomeier, M.: Neue Konzepte reduzieren die Kosten verteilter Client-Server-Infrastrukturen, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 49/1996, S. 11

[o. V. 1997]

o. V.: Zentrale Speicherung gilt als NC-Highlight, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 46/1997, S. 1

[o. V. 1998a]

o. V.: Disziplinen wachsen nur langsam zusammen, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 10/1998, S. 13

[o. V. 1998b]

o. V.: Softwareverteilung über das Internet, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 5/1998, S. 10

[PAVONE 1997]

PAVONE Informationssysteme GmbH: Internes Arbeitspapier zur Umsetzung eines Eskalationsmodells in PAVONE Espresso, Paderborn, 1997

[PAVONE 1998a]

PAVONE Informationssysteme GmbH: PAVONE Espresso - Die ersten Schritte, Paderborn, 1998

[PAVONE 1998b]

PAVONE Informationssysteme GmbH: Projekt-Dokumentation - Benutzerhandbuch, Paderborn, 1998

[PAVONE 1998c]

PAVONE Informationssysteme GmbH: Projekt-Dokumentation - Administratorhandbuch, Paderborn, 1998

[PAVONE 1998d]

PAVONE Informationssysteme GmbH: Projekt-Dokumentation - Portfolio DB Benutzerhandbuch, Paderborn, 1998

[PAVONE 1998e]

PAVONE Informationssysteme GmbH: Projekt-Dokumentation - Portfolio DB Administratorhandbuch, Paderborn, 1998

[Rademacher 1997]

Rademacher, R.: Die Abkapselung von Clients reduziert die IT-Kosten schnell, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 46/1997, S. 27

[Reiß 1997]

Reiß, M.: Change Management: Programme, Projekte, Prozesse, Schäffer-Poeschel Verlag, Stuttgart, 1997

[Reiterer 1994]

Reiterer, H.: User Interface Evaluation and Design, R. Oldenbourg Verlag, München/Wien, 1994

[Riemensperger 1993]

Riemensperger, F.: Die Entwicklungsarchitektur ist der Knackpunkt des Client/Server-Computing, in: Computer Zeitung, Konradin Verlag, Leinfelden-Echterdingen, Nr. 7/1993, S. 14

[Rose 1996]

Rose, T.: Vorgangsmanagementsysteme: Modellierungs- und Implementierungskonzepte, in: Vossen, G.; Becker, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow Management: Modelle, Methoden, Werkzeuge, International Thomson Publishing, Bonn usw., 1996, S. 319-334

[Scheer et al. 1997]

Scheer, A.-W.; Hagemeyer, J.; Schmidt, Y.: Workflow Management: Von der Idee zur Sollprozeßgestaltung, in: Office Management, Verlagsgruppe Handelsblatt, Düsseldorf, Heft 10, Dezember, 1997, S. 29-34

[Schneider 1997]

Schneider, H.-J.: Lexikon der Informatik und Datenverarbeitung, 4. Auflage, Oldenbourg Verlag, München/Wien, 1997

[Tivoli 1997]

Tivoli Systems Inc.: Deployment Management, o. O., 1997, aus:
http://www.tivoli.com/n_sol/depmgt.html am: 22.10.97

[Venetz 1998]

Venetz, L.A.: Lotusphere '98 Orlando: Knowledge Management als Rückgrat, in: Notes Magazin, IWT Magazin Verlags-GmbH, Vaterstetten, Nr. 2, April/Mai 1998, S. 6

[Weiß/Krcmar 1996]

Weiß, D.; Krcmar, H.: Workflow-Management: Herkunft und Klassifikation, in: Wirtschaftsinformatik, Friedr. Vieweg & Sohn Verlagsgesellschaft, Braunschweig/Wiesbaden, Nr. 5, Oktober, 1996, S. 503-513

[Wess 1996]

Wess, S.: Intelligente Systeme für den Customer Support, in: Wirtschaftsinformatik, Friedr. Vieweg & Sohn Verlagsgesellschaft, Braunschweig/Wiesbaden, Nr. 1, Februar, 1996, S. 23-31

[Wessler 1995]

Wessler, B.: Die hohen Integrationskosten sind der Tribut für neugewonnene Freiheiten, in: Computer Zeitung Extra: Client-Server-Architekturen, Konradin Verlag, Leinfelden-Echterdingen, Sondernummer, September, 1995, S. 21-23

[WfMC 1994]

WfMC: Workflow Management Coalition: The Workflow Reference Model, Document Number TC00-1003 Document Status - Issue 1.1, by David Hollingworth, Nov 29, 1994

[WfMC 1996]

WfMC: Workflow Management Coalition: Terminology & Glossary, Document Number WMFC-TC-1011 Document Status - Issue 2.0, June, 1996

[Wimmer 1997]

Wimmer, K.H.: PCs durch automatisiertes Management rentabler, in: Office Management, Verlagsgruppe Handelsblatt, Düsseldorf, Heft 10, Dezember, 1997, S. 43-45

A.2 Abbildungen

Abb. 1.1-1: Vergleich der DV-Kostenverteilung	2
Abb. 2.1-1: Klassifikationsschema für Groupware	5
Abb. 2.3-1: Begriffsdefinitionen zum Workflow Management im Projekt	8
Abb. 3.1-1: Systemmanagement-Modell der Meta Group in schematischer Übersicht	12
Abb. 3.1-2: Interdependenzen zwischen den Framework-Disziplinen.....	14
Abb. 3.1-3: Verzahnung technischer und betriebswirtschaftlicher Anwendungen	14
Abb. 3.2-1: Workflow-Referenzmodell - Komponenten und Schnittstellen.....	15
Abb. 3.2-2: Workflow Life Cycle.....	16
Abb. 3.2-3: Workflow-Kontinuum.....	17

Abb. 3.3-1: ABC-Modell der Software-Ergonomie	19
Abb. 4.2.1-1: Integratives Systems Management mit klarer Serviceausrichtung.....	24
Abb. 4.2.3-1: Zeitlicher Projektverlauf	25
Abb. 4.3.1-1: Klassifikationsraster zur Change-Typen-Ermittlung.....	26
Abb. 4.3.1-2: Abbildung der ausgewählten Geschäftsvorfälle auf die Change-Typen	27
Abb. 4.3.2-1: Grobgliederung des Change Administration Prozesses	28
Abb. 4.3.2-2: Struktur des Change Administration Prozesses	28
Abb. 4.3.3-1: Rollen im Change Management-Prozeß	29
Abb. 4.3.3-2: Aufgaben und Verantwortungen der Kernrollen.....	30
Abb. 4.3.4-1: Prozeßfluß im Change Management.....	31
Abb. 4.3.5-1: Kritische Erfolgsfaktoren für Prozesse	32
Abb. 5-1: Architekturüberblick über die genutzten Plattform-Technologien	36
Abb. 5.2-1: Erweiterung von LotusScript um das Midas Rich Text LSX.....	40
Abb. 5.3-1: PAVONE Espresso Architekturüberblick.....	41
Abb. 5.3.1-1: Organisationseinheiten von PAVONE Espresso.....	43
Abb. 5.3.2-1: Graphische Modellierungsoberfläche des Espresso ProcessModelers....	45
Abb. 6.1.1-1: Architekturmodell der prototypischen Implementierung	46
Abb. 6.1.2-1: Drei Möglichkeiten der Initiierung eines Change-Workflows.....	47
Abb. 6.1.2-2: Überblick über den realisierten Prozeßverlauf.....	48
Abb. 6.2.1-1: Mögliche Realisierungsformen der Prozeßarchitektur.....	49
Abb. 6.2.2-1: Synchronisation der Change-Typ-Prozesse im betrachteten Beispiel.....	52
Abb. 6.3.2-1: Schichtenmodell zur Realisierung der Anforderungsliste.....	57
Abb. 6.3.2-2: Separierung der Anforderungsliste für die Change-Typ-Prozesse.....	58
Abb. 6.3.2-3: Historie eines Beispiel-Beschaffungsantrages	59
Abb. 6.4.1-1: Zusammenspiel der implementierten Masken und Teilmasken.....	60

Abb. 6.4.1-2: Konzeptionelles Maskendesign für den Hauptprozeß.....	61
Abb. 6.4.2-1: Navigationskonzept des implementierten Prototypen.....	62
Abb. 6.4.2-2: Bearbeitungskonzept des implementierten Prototypen	64
Abb. 6.4.3-1: Paradigma des Electronic Shopping in der Portfolio-Datenbank.....	65
Abb. C.1-1: Beschaffungsantrag der Deutschen Bundesbank.....	83

A.3 Tabellen

Tab. 1.1-1: Verteilte Umgebung erhöht die Komplexität.....	1
Tab. C.2.3-1: Antwortverteilung zur korrekten Umsetzung der Geschäftsprozesse	87
Tab. C.2.3-2: Antwortverteilung zur komfortablen Bedienbarkeit des Prototypen	87
Tab. C.2.3-3: Antwortverteilung zur Fehlerfreiheit des Prototypen.....	88
Tab. C.2.3-4: Antwortverteilung zur Sinnhaftigkeit des Prototypeneinsatzes	88

A.4 Abkürzungen

API	Application Programming Interface
BBK	Deutsche Bundesbank
CT	Change-Typ
DIN/EN	Deutsche Industrie Norm/Europäische Norm
DME	Distributed Management Environment
EVADIS	Evaluation of Dialogue Systems
GMD	Gesellschaft für Mathematik und Datenverarbeitung
HW	Hardware
HW-CT	Hardware-Change-Typ
IBM	International Business Machines
ID	Identifizier
IDC	International Data Corporation
IDE	Integrated Development Environment

ISO	International Standards Organization
LAN	Local Area Network
LSX	LotusScript Extension
NMS	Network Management Station
OLE	Object Linking and Embedding
OSF	Open Software Foundation
OSI	Open Software Interconnection
QS	Qualitätssicherung
RADD	Rapid Application Development and Deployment
SM	Systems Management
SNMP	Simple Network Management Protocol
SW	Software
SW-CT	Software-Change-Typ
TCO	Total Cost of Ownership
UHD	User Help Desk
VDI	Verein Deutscher Ingenieure
WAN	Wide Area Network
WfMC	Workflow Management Coalition

B Eidesstattliche Erklärung**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 nicht veröffentlicht.

Paderborn, den
(Datum) (Unterschrift)

C Anlagen

C.1 Beschaffungsantrag in Papierform

Abbildung C.1-1 zeigt den bereits existierenden Beschaffungsantrag der Deutschen Bundesbank. Als Bestandteil des Formularwesens war dieser Antrag in elektronische Form zu überführen.

Ordnungsmerkmal	Kostenstelle	Frankfurt am Main, _____
_____	_____	Hausruf _____

A. Beschaffungsantrag
für Büroeinrichtungsgegenstände, Maschinen aller Art, Büro-, Text- und Kommunikationssysteme, DV-Hard- und Software

Wir bitten um Beschaffung von _____ Zutreffendes ankreuzen oder ausfüllen

Menge	Bezeichnung und Begründung (ggf. auf einem besonderen Blatt)

Bei DV-Nachbestellungen/Ersatzbeschaffungen bitte den Typ, Modell/Feature, die Maschinen-Nr., BBK-Inventar-Nr. oder PC-Nr. des bereits installierten Gerätes bzw. die Version der Software angeben. Bei Ersatzbeschaffungen zusätzlich die Hard- und Softwareteile angeben, die zurückgegeben werden.

Anfordernder Bereich
Notwendigkeit bescheinigt

Unterschrift des Kostenstellenleiters	Name des Antragstellers	Aufzustellen in (Gebäude/Raum-Nr.), Nutzer, Ordnungsmerkmal
_____	_____	_____

Weiterzuleiten an **Vb 20** Beschaffungen (außer Hard- und Software sowie Telefonausstattung)
 Dv 33 Hard- und Software einschließlich Zubehör
 Vb 320-1 Telefonausstattung

B. Bearbeitungsvermerke Eingang _____

1. Prüfung der Zweckmäßigkeit der beantragten Gegenstände und Festlegung der Bestellmerkmale einschließlich Preisermittlung

Menge	Bestellmerkmale	Einzelpreis	Gesamtpreis

Unterschrift, Datum, Ordnungsmerkmal

2. Kostenstelle
In der Plankostenrechnung/in dem Investitionsplan 19 _____ sind für diese Beschaffung Mittel

genehmigt (Weiterleitung an Vb 20/Vb 320-1)
 eingestellt, aber noch nicht genehmigt (mit eingehender Begründung Weiterleitung auf dem Dienstweg an C 3)
 noch nicht eingestellt (mit eingehender Begründung Weiterleitung auf dem Dienstweg an C 1) worden.

Unterschrift, Datum

3. Ggf. C 3 zur Genehmigung/Stellungnahme

Unterschrift gemäß Ermächtigungsregelung, Datum

4. Ggf. C 1 zur Aufnahme in die Plankostenrechnung 19 _____ 5. Eingang bei Vb 20/Vb 320-1
 Unterschrift, Datum NZ, Datum

_____ _____

BBK-Vordr. 1211a I 12.93 - 6 5

Abb. C.1-1: Beschaffungsantrag der Deutschen Bundesbank

C.2 Endanwenderschulung

Die Endanwender wurden in einem eintägigen Seminar im realisierten Prototypen geschult. Insgesamt fanden vier Schulungen im Zeitraum vom 3.-6. Februar 1998 bei der Deutschen Bundesbank in Frankfurt am Main statt, wobei der Autor selbst die Schulungen durchgeführt hat.

C.2.1 Agenda



AGENDA

9:00 - 10:15	Begrüßung, Einführung, Ablauf Hintergrund und Ziele des CAP-Projektes Theoretische Grundlagen des Workflow-Managements Der Change Prozeß bei der DdD Rollen und Aufgabenbereiche
10:15 - 10:30	Pause
10:30 - 12:00	Architektur: Grundlagen der Espresso-Umgebung Konfiguration der Espresso-Datenbanken Aktionen \ weitere Einstellungen Espresso Einführung in die CAP-Anwendung Navigator Bedienung der Aufgabenliste Anlegen eines neuen Vorgangs Aufbau des CAP-Dokumentes Grundlegende Bedienung der CAP-Dokumente Ausfüllen eines neu erstellten Vorgangs Speichern
12:00 - 13:00	Pause
13:00 - 14:45	Praktische Übung zur Wiederholung der bisherigen Lerninhalte Einführung in die CAP-Anwendung (Fortsetzung) Kopplung zur Portfolio-DB Abschließen von Aufgaben Wiederfinden von Dokumenten Weiterführung des Beispiels eines neuen Vorgangs Allgemeine Espresso-Vorgehensweisen Reservieren Zur Kenntnisnahme
14:45 - 15:00	Pause
15:00 - 16:30	Praktische Übung

C.2.2 Auswertungsbogen

Vorderseite

IBM Unternehmensberatung	Change Administration Prototyp Anwenderschulung	Deutsche Bundesbank					
<h2 style="margin: 0;">Auswertungsbogen</h2> <p style="margin: 10px 0;">Ihre Meinung zum Change Administration Prototyp ist uns wichtig. Bitte helfen Sie uns, das CAP-System und den zugrundeliegenden Change-Prozeß weiter zu verbessern, indem Sie folgende Aussagen durch Ankreuzen bewerten und die Fragen auf der Rückseite des Blattes beantworten. Vielen Dank für Ihre Unterstützung!</p>							
<p><u>Name:</u></p> <p><u>Ordnungsmerkmal:</u></p> <p><u>Datum:</u></p>	<p><u>Legende:</u> ++ vollständig zutreffend + größtenteils zutreffend 0 teilweise zutreffend - größtenteils nicht zutreffend -- gar nicht zutreffend</p>						
<p>Die in der Schulung vorgestellten Geschäftsprozesse wurden meines Erachtens im CAP-System korrekt umgesetzt.</p> <p>Bemerkungen:</p>	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">++</td> <td style="width: 15%; text-align: center;">+</td> <td style="width: 15%; text-align: center;">0</td> <td style="width: 15%; text-align: center;">-</td> <td style="width: 15%; text-align: center;">--</td> </tr> </table>	++	+	0	-	--	
++	+	0	-	--			
<p>Das CAP-System ist leicht und komfortabel bedienbar.</p> <p>Bemerkungen:</p>	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">++</td> <td style="width: 15%; text-align: center;">+</td> <td style="width: 15%; text-align: center;">0</td> <td style="width: 15%; text-align: center;">-</td> <td style="width: 15%; text-align: center;">--</td> </tr> </table>	++	+	0	-	--	
++	+	0	-	--			
<p>Mir sind keine gravierenden Fehler bei der Ausführung des CAP-Systems aufgefallen.</p> <p>Bemerkungen:</p>	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">++</td> <td style="width: 15%; text-align: center;">+</td> <td style="width: 15%; text-align: center;">0</td> <td style="width: 15%; text-align: center;">-</td> <td style="width: 15%; text-align: center;">--</td> </tr> </table>	++	+	0	-	--	
++	+	0	-	--			
<p>Ich halte die toolgestützte Bearbeitung von Änderungswünschen im IT-Umfeld für eine sinnvolle Erweiterung der bestehenden DV-Dienstleistungen.</p> <p>Bemerkungen:</p>	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">++</td> <td style="width: 15%; text-align: center;">+</td> <td style="width: 15%; text-align: center;">0</td> <td style="width: 15%; text-align: center;">-</td> <td style="width: 15%; text-align: center;">--</td> </tr> </table>	++	+	0	-	--	
++	+	0	-	--			

Rückseite

IBM Unternehmensberatung	Change Administration Prototyp Anwenderschulung	Deutsche Bundesbank
--------------------------	--	---------------------

Folgende **Änderungen am Change Prozeß** sind aus meiner Sicht zum reibungslosen Ablauf unbedingt erforderlich (z. B. Änderungen oder Ergänzungen in der Abfolge der Aufgaben, Abhängigkeiten, Informationsfluß):

Folgende **Änderungen am Change Prozeß** wären wünschenswert:

Folgende **Änderungen am CAP-System** sind aus meiner Sicht zum reibungslosen Ablauf unbedingt erforderlich (z. B. Änderung der Masken, Ansichten, zusätzliche Funktionalität):

Folgende **Änderungen am CAP-System** wären wünschenswert:

Sonstige Anmerkungen (ggf. zusätzliches Blatt benutzen):

C.2.3 Ergebnis der Befragung

Insgesamt liegen dem Autor 29 Auswertungsbögen vor.

Auf eine Wiedergabe der teilweise umfangreichen Bemerkungen soll an dieser Stelle verzichtet werden, da eine Beschränkung auf den zu betrachtenden Teil der Systembedienbarkeit für den Leser nicht nachvollziehbar wäre und die Bemerkungen oftmals extrem detailliert sind (z. B. „Angabe der Leisten-Bodentanknummer“ als unbedingt

erforderliche Änderung am System). Zudem kann das Bewertungsraster der Vorderseite als ausreichend angesehen werden, da es alle Bemerkungen zu einer quantifizierten Aussage summiert.

Die folgenden Tabellen geben die Antwortverteilung zu den vier zu wertenden Aussagen wieder. Summendifferenzen im Vergleich zur Grundgesamtheit ergeben sich durch fehlende Wertungen auf den Auswertungsbögen zu der betrachteten Aussage.

Angeschlossen an die jeweilige Antwortverteilung wird ein Durchschnittswert angegeben, der sich aus der Gewichtung von +5 für „vollständig zutreffend“ bis +1 für „gar nicht zutreffend“ ermittelt.

Aussage 1: „Die in der Schulung vorgestellten Geschäftsprozesse wurden meines Erachtens im CAP-System korrekt umgesetzt.“

Tendenzielle Aussage	Anzahl Nennungen
vollständig zutreffend	2
größtenteils zutreffend	17
teilweise zutreffend	6
größtenteils nicht zutreffend	0
gar nicht zutreffend	0

Tab. C.2.3-1: Antwortverteilung zur korrekten Umsetzung der Geschäftsprozesse

Ermittelter Durchschnittswert: +3,84

Aussage 2: „Das CAP-System ist leicht und komfortabel bedienbar.“

Tendenzielle Aussage	Anzahl Nennungen
vollständig zutreffend	5
größtenteils zutreffend	16
teilweise zutreffend	7
größtenteils nicht zutreffend	0
gar nicht zutreffend	0

Tab. C.2.3-2: Antwortverteilung zur komfortablen Bedienbarkeit des Prototypen

Ermittelter Durchschnittswert: +3,93

Aussage 3: „Mir sind keine gravierenden Fehler bei der Ausführung des CAP-Systems aufgefallen.“

Tendenzielle Aussage	Anzahl Nennungen
vollständig zutreffend	4
größtenteils zutreffend	16
teilweise zutreffend	7
größtenteils nicht zutreffend	0
gar nicht zutreffend	0

Tab. C.2.3-3: Antwortverteilung zur Fehlerfreiheit des Prototypen

Ermittelter Durchschnittswert: +3,89

Aussage 4: „Ich halte die toolgestützte Bearbeitung von Änderungswünschen im IT-Umfeld für eine sinnvolle Erweiterung der bestehenden DV-Dienstleistungen.“

Tendenzielle Aussage	Anzahl Nennungen
vollständig zutreffend	18
größtenteils zutreffend	6
teilweise zutreffend	3
größtenteils nicht zutreffend	0
gar nicht zutreffend	0

Tab. C.2.3-4: Antwortverteilung zur Sinnhaftigkeit des Prototypeneinsatzes

Ermittelter Durchschnittswert: +4,56

C.3 Realisierter Prototyp

Dieses Kapitel behandelt die erstellte Applikation und enthält u. a. eine Anweisung zur Installation des Prototypen von der beiliegenden CD-ROM aus. Bei der Formulierung der Installationsanweisung geht der Autor davon aus, daß der Leser Kenntnisse des Betriebssystems und von Lotus Notes besitzt. Bei Fragen zu diesen Bereichen lesen Sie bitte in dem zuständigen Handbuch nach.

Bitte beachten Sie, daß die beigefügten Lotus Notes-IDs und -Datenbanken der Deutschen Bundesbank nur zum Testen der Applikation beigefügt wurden und eine weitergehende Nutzung hiermit untersagt wird.

C.3.1 Dateien des Prototypen

Alle notwendigen Dateien des Prototypen befinden sich auf der beigefügten CD-ROM im Verzeichnis `\APPLIKATION` bzw. in dessen Unterverzeichnissen. Zur Installation auf Ihrem Rechner beachten Sie bitte die Systemanforderungen in Kapitel C.3.2 und folgen Sie der Installationsanleitung in Kapitel C.3.3. Für eine anschließende De-Installation beachten Sie bitte die Hinweise in Kapitel C.3.4.

C.3.2 Anforderungen an Hard- und Software

Zur Installation des Prototypen bestehen folgende Systemvoraussetzungen:

- IBM kompatibler Computer mit 80486 Prozessor (Pentium oder höher empfohlen)
- 16 MB RAM (32 MB oder mehr empfohlen)
- SVGA (800*600) mit mind. 256 Farben
- mindestens 20 MB freier Speicherplatz auf der Festplatte
- Windows 95 bzw. Windows NT 3.51 (oder höher)
- Lotus Notes Version 4.5.2 oder höher

C.3.3 Installationsanleitung

Applikation installieren

Erzeugen Sie ein neues Verzeichnis auf Ihrer Festplatte. Dieses sollte am Besten relativ unterhalb des Notes-Datenverzeichnisses liegen.

Kopieren Sie nun alle Dateien aus dem Verzeichnis `\APPLIKATION` der CD-ROM in das neue Verzeichnis. Die Datei `CAPNAMES.NSF` muß dabei in jedem Fall im Notes-Datenverzeichnis oder darunter liegen.

Die Test-IDs aus dem Verzeichnis `\APPLIKATION\IDS` werden ebenfalls in das neu angelegte Verzeichnis kopiert. Die Dateien aus dem Verzeichnis `\APPLIKATION\MIDAS` müssen in das Notes-Programmverzeichnis kopiert werden. Die Textdateien im Verzeichnis `\APPLIKATION\PROBLEM MANAGEMENT` kopieren Sie bitte nach `C:\`.

Applikation einrichten

Als erster Schritt muß das Namens- und Adreßbuch `CAPNAMESBBK.NSF` als solches verfügbar gemacht werden. Dazu rufen Sie in Lotus Notes über das Menü *bitte Datei - Extras - Benutzervorgaben...* auf. Dort können Sie im Bereich *Mail* die *Lokale*

Adreßbücher einstellen. Nehmen Sie hier bitte die Datei `CAPNAMESBBK.NSF` über die Schaltfläche hinzu.

Fügen Sie nun alle kopierten Lotus Notes-Datenbanken zu Ihrer Arbeitsoberfläche hinzu (bitte tatsächlich *Symbol hinzufügen* und nicht *Öffnen* wählen). Für den besten Überblick über die sieben Datenbanken sollten Sie diese gemäß dem vorgestellten Architekturmodell in Abbildung 6.1.1-1 anordnen.

Bevor Sie den Prototypen nun einsetzen können, müssen die Datenbanken noch konfiguriert werden. Den ersten Schritt müssen Sie dazu mit Ihrer eigenen Lotus Notes-ID ausführen, da zwei Designelemente angepaßt werden müssen (Sie benötigen eine Vollversion von Lotus Notes, um diesen Schritt auszuführen). Öffnen Sie daher die Anwendungsdatenbank und bestätigen Sie einfach den Einstellungsdialog, ohne daß Sie sich um diesen weiter kümmern. Wählen Sie nun das gemeinsam genutzte Feld *BBKpfDBPath* aus und passen Sie die Pfadeinstellung an das von Ihnen gewählte Verzeichnis an. Genau die gleiche Einstellung nehmen Sie bitte in der Script-Bibliothek *BbkAgentLibBasis*. Deklarieren Sie dort entsprechend die Konstante *pfDBPath*. Desweiteren sollten Sie noch die Einstellungsdokumente für die Espresso Runtime in der Ansicht Einstellungen an Ihre Verzeichnisstruktur anpassen.

Wechseln Sie nun zu einer der kopierten Lotus Notes Test-IDs. Das Paßwort für jede ID lautet *passwort*. Wählen Sie sodann die Espresso Vorgangstypendatenbank in Ihrer Arbeitsoberfläche durch Selektieren oder Öffnen aus und rufen Sie die Aktion *Vorgangstypen-DB zur Auswahl* aus dem Menü *Aktionen* aus. Wiederholen Sie dies für die Organisations- und die Vorlagendatenbank. Wenn Sie nun die Anwendungsdatenbank öffnen, erscheint wiederum der Einstellungsdialog, den Sie auch über die Aktion *Weitere - Einstellungen Espresso...* erreichen können. In diesem Dialog referenzieren Sie bitte die anderen drei Datenbanken, indem Sie der Beschreibung in Kapitel 5.5 der Datei `ESPRESSO INSTALLATIONSANLEITUNG.PDF` im Verzeichnis `\APPLIKATION\DOKUMENTATION` folgen.

Applikation nutzen

Sie können nun den Prototypen einsetzen. Eine Anleitung entnehmen Sie bitte dem Benutzerhandbuch der Projektdokumentation (Datei `BENUTZERHANDBUCH.PDF` im Verzeichnis `\APPLIKATION\DOKUMENTATION`). Einen Einblick in die umgesetzte Funktionalität vermittelt auch die Lotus ScreenCam `CAP-PROJEKT.SCM` im Verzeichnis

\APPLIKATION\DEMONSTRATION. Einen ScreenCam-Player für alle Windows-Plattformen finden Sie im gleichen Verzeichnis. Bitte starten Sie diese Datei SCPLAYER.EXE und wählen Sie dann die erwähnte ScreenCam-Datei aus.

Die beigefügten zehn Test-IDs sind für eine Evaluation des Prototypen so eingerichtet, daß sie möglichst alle Rollen abdecken. Die konkrete Zuordnung entnehmen Sie bitte der Espresso Organisationsdatenbank. Wichtig ist anzumerken, daß die Test-ID *ccm1* als Workflow-Designer immer alle Dokumente editieren darf. Die Test-ID *ccm10* wurde keiner Rolle zugeordnet, um einen normalen Antragsteller zu simulieren.

C.3.4 De-Installationsanleitung

Für die De-Installation löschen Sie einfach die laut Kapitel C.3.3 installierten Dateien und entfernen die Datenbanken aus der Arbeitsoberfläche von Lotus Notes. Löschen Sie weiterhin den Verweis auf das aufgenommene Namens- und Adreßbuch in Ihren Benutzervorgaben. Die verbleibenden Einstellungen in Ihrer NOTES.INI-Datei beeinträchtigen nicht die zukünftige Funktion Ihres Computers.

C.3.5 Dokumentation

Die Dokumentation befindet sich aus Gründen der Übersichtlichkeit auf der beigefügten CD-ROM im Verzeichnis \APPLIKATION\DOKUMENTATION. Zum Einsehen der Dokumente steht im Verzeichnis \QUELLEN auf der CD-ROM eine Installationsversion des Adobe Acrobat Reader 3.01 für Microsoft Windows 95 und NT bereit (Datei ACROBATREADER.EXE).

Die Projekt-Dokumentation besteht aus folgenden Komponenten:

- ESPRESSO INSTALLATIONSANLEITUNG.PDF
Espresso Runtime - Einführung und Installation
- BENUTZERHANDBUCH.PDF
Projekt-Dokumentation - Benutzerhandbuch [PAVONE 1998b]
- ADMINISTRATORHANDBUCH.PDF
Projekt-Dokumentation - Administratorhandbuch [PAVONE 1998c]
- PORTFOLIO BENUTZERHANDBUCH.PDF
Projekt-Dokumentation - Portfolio DB Benutzerhandbuch [PAVONE 1998d]
- PORTFOLIO ADMINISTRATORHANDBUCH.PDF
Projekt-Dokumentation - Portfolio DB Administratorhandbuch [PAVONE 1998e]

Zusätzlich zur eigentlichen Projekt-Dokumentation wurden noch aufgenommen:

- ESPRESSO FLYER . PDF
PAVONE Espresso Produkt-Flyer
- ESPRESSO ERSTE SCHRITTE . PDF
PAVONE Espresso - Die ersten Schritte [PAVONE 1998a]

C.4 Online-Quellen

Alle genutzten Online-Quellen befinden sich auf der beigelegten CD-ROM im Verzeichnis \QUELLEN. Zusätzlich sind dort weiterführende Quellen abgelegt, die im Rahmen der Diplomarbeit ermittelt, in der Arbeit selbst aber nicht verwendet wurden.

Die enthaltenen Dateien sind:

- PERWEB . NSF
Online-Quellen in Form einer Personal Web Navigator-Datenbank
- RTHELP4 . NSF
Midas Rich Text LSX On-Line Hilfe
- NOTEARCH . PDF
Artikel: „Notes: A sustainable platform architecture“
- LSPROGd . PDF
Artikel: „LotusScript R3 Programmer’s Guide“
- LSLANREF . PDF
Artikel: „LotusScript R3 Language Reference“
- LN-DBMS . PDF
Artikel: „Lotus Notes Enterprise and DBMS Integration“
- DIN29241 . PDF
DIN/EN 29241 Teil 10 (Entwurf Januar 1994)

Beigefügt wurde ferner der Adobe Acrobat Reader in der Version 3.01 als Installationsversion für Microsoft Windows 95 und NT (Datei ACROBATREADER . EXE).

C.5 Diplomarbeit in elektronischer Form

Die elektronische Form der Diplomarbeit befindet sich auf der beigefügten CD-ROM im Verzeichnis \DIPLOMARBEIT.

Die enthaltenen Dateien sind:

- DIPLOMARBEIT RALF LUTTER.DOC und DIPLOMARBEIT RALF LUTTER.DOT
Diplomarbeit im Format Microsoft Word für Windows 95 mit zugehöriger Dokumentenvorlage
- DIPLOMARBEIT RALF LUTTER.PRZ
Genutzte Graphiken im Format Lotus Freelance Graphics 97