

KONZEPTION UND ENTWURF EINER KOMPONENTE FÜR DIE ARBEITSLISTENVERWALTUNG

Masterarbeit an der Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
Institut für Datenbanken und Informationssysteme



Verfasser:
Romy Opitz

Gutachter:
Prof. Dr. Peter Dadam
Dr. Stefanie Rinderle-Ma

Juni 2007

Inhaltsverzeichnis

Kurzfassung.....	5
1 Einleitung	7
1.1 Motivation	7
1.2 Aspekte der Arbeitslistenverwaltung	10
1.3 Beispiel.....	13
1.4 Aufgabenstellung und Zielsetzung.....	14
1.5 Aufbau der Arbeit.....	15
2 Anforderungen an die Arbeitslistenverwaltung	17
2.1 Erweiterte Funktionalität der Arbeitslistenverwaltung	17
2.2 Das Workflow-Referenzmodell	18
3 Bearbeiterzuordnung	21
3.1 Auflösung der Bearbeiterformeln.....	21
3.2 Hinterlegung der Bearbeiterformeln	27
3.3 Verwaltung der Arbeitslisten und Aktivitäten	29
3.4 Kommunikation mit dem Arbeitslistenkunden	34
3.5 Zusammenfassung.....	35
4 Verteilungsverfahren.....	37
4.1 Gleichmäßige Vorabverteilung durch Round Robin.....	37
4.2 Lastabhängige Verteilung	41
4.3 Auswahl des Verteilungsverfahrens.....	43
4.4 Zusammenfassung.....	44
5 Dynamische Neuverteilung	45
5.1 Anpassung der Verteilung an Umgebungsbedingungen	45
5.2 Dynamische Änderung der Bearbeiterzuordnung	47
5.2.1 Delegation	47
5.2.2 Vertreterregelungen.....	53
5.2.3 Gegenüberstellung Delegation und Vertreterregelung.....	56
5.3 Zusammenfassung.....	57
6 Priorisierung	59
6.1 Abläufe einer Priorisierung	59

6.2	Automatische Priorisierung	60
6.3	Auswirkung der Priorisierung	61
6.4	Priorisierungsinformationen	65
6.5	Zusammenfassung	66
7	Entwurf	67
7.1	Datenmodell	68
7.2	Schnittstelle zum WfMS	74
7.2.1	Anforderungen	74
7.2.2	ADEPT2	78
7.2.3	MQ Workflow	81
7.2.4	TIBCO iProcess Engine	84
7.2.5	Vergleich der Systeme	87
7.2.6	Entwurf einer Schnittstelle zu ADEPT2	88
7.3	Schnittstelle zum Arbeitslistenklienten	91
7.4	Schnittstellen für die Verarbeitung von Arbeitslisten	94
7.4.1	Bearbeiterzuordnung	94
7.4.2	Delegation	98
7.4.3	Priorisierung	99
7.5	Überblick über die Schnittstellen	103
8	Ausblick	105
9	Zusammenfassung	109
	Literaturverzeichnis	111
	Abbildungsverzeichnis	117
	Glossar	119

Kurzfassung

Workflow-Management-Systeme unterstützen die automatische Ausführung von Arbeitsabläufen innerhalb einer Organisation. Sie erlauben es Prozesse zu definieren, auszuführen und ihren Ablauf zu überwachen und zu steuern. Dadurch können Arbeitsabläufe optimiert, Durchlauf- und Bearbeitungszeiten verbessert und folglich Kosten reduziert werden.

Die Arbeitslistenverwaltung stellt eine grundlegende und wichtige Komponente innerhalb eines jeden Workflow-Management-Systems dar. Sie ist das Bindeglied zwischen laufenden Prozessen und den Benutzern, die für ihre Abarbeitung verantwortlich sind. Die Benutzer haben über die Arbeitslistenverwaltung Zugriff auf eine oder mehrere Arbeitslisten, in denen alle Aktivitäten zusammengefasst sind, die sie zu einem gegebenen Zeitpunkt bearbeiten können. Außerdem bietet die Arbeitslistenverwaltung den Benutzern die Möglichkeit ihre Arbeitslisten nach eigenem Ermessen zu aktualisieren und trägt somit zur schnellen und reibungslosen Abarbeitung der Aktivitäten bei.

Für einen erfolgreichen und dauerhaften Einsatz in der sich rasant weiterentwickelnden Umgebung der Workflow-Management-Systeme reicht diese Funktionalität jedoch nicht aus. Möglichkeiten zur Erweiterung eröffnen sich beispielsweise durch verschiedene Verfahren zur Erstverteilung bzw. zur manuellen oder dynamischen Neuverteilung von Aktivitäten. Auch eine Priorisierung von Aktivitäten muss von der Arbeitslistenverwaltung erfasst und an die betroffenen Bearbeiter weitergegeben werden. Da beim Einsatz von Workflow-Management-Systemen mit einer großen Anzahl von Bearbeitern und Arbeitslisten sowie einem hohen Durchsatz von Aufgaben gerechnet werden muss, ist es außerdem dringend notwendig, dass sich eine Arbeitslistenverwaltung durch geeignete Verwendung und Verteilung von Ressourcen auszeichnet. Besondere Anforderungen werden im Bereich der komponentenbasierten Entwicklung von Workflow-Anwendungen gestellt. Als eigenständige Komponente muss die Arbeitslistenverwaltung sowohl vom Workflow-System-Kern als vom Arbeitslistenklienten unabhängig sein.

Im Rahmen dieser Arbeit werden die genannten Aspekte untersucht und ihre Einsatzmöglichkeiten diskutiert. Dabei wird die Zuordnung von Aktivitäten auf einzelne Arbeitslisten und deren Verwaltung und Aktualisierung überprüft. Für die Verteilung von Aktivitäten auf Bearbeiter werden verschiedene Verfahren vorgestellt. Im Rahmen der dynamischen Neuverteilung werden sowohl Delegation als auch Vertreterregeln als Varianten der manuellen Bearbeiterzuordnung diskutiert. Die Priorisierung von Aktivitäten und ihre Darstellung innerhalb der Arbeitslisten werden untersucht. Des Weiteren wird ein Entwurf für eine Arbeitslistenverwaltungskomponente angefertigt und vorgestellt.

1 Einleitung

1.1 Motivation

Durch technischen Fortschritt, Automatisierung und Vernetzung ist in den letzten Jahrzehnten das Arbeitsaufkommen innerhalb einzelner Unternehmen beträchtlich gestiegen. Parallel dazu hat sich die durchschnittliche Anzahl der Beschäftigten pro Unternehmen stark vergrößert. Vorbei sind die Zeiten, in denen eine Handvoll Arbeiter in kleinen Familienbetrieben jeden Tag die gleiche Arbeit verrichtete. Heute operieren die Firmen international, sind die Arbeitsprozesse komplexer und die Arbeiter verteilt auf verschiedene Standorte.

Das neue Konzept heißt verteilte Zusammenarbeit. Dabei wird ein komplexer Prozess, ein so genannter Arbeitsablauf, in viele überschaubare Einzelschritte eingeteilt, welche von mehreren Arbeitern nacheinander gemeinsam abgearbeitet werden, auch unabhängig von räumlicher oder zeitlicher Nähe.

Eine solche Arbeitsweise bedarf einer guten Organisation und Kommunikation. Unterstützung holt man sich durch den Einsatz geeigneter Computer- und Softwaresysteme, die in den letzten Jahren unverzichtbar geworden sind. Netzwerke und Internetzugang stellen die Grundlage für verteiltes Arbeiten. Die Kommunikation der Mitarbeiter läuft von Emails über interne Nachrichtensysteme bis hin zu Videokonferenzen. Content-Management-Systeme (CMS) regeln den gleichzeitigen, entfernten Zugriff auf arbeitsrelevante Daten. Der ordnungsgemäße Ablauf eines Prozesses und die Reihenfolge der einzelnen Prozessschritte kann über Workflow-Management-Systeme (WfMS) sichergestellt werden.

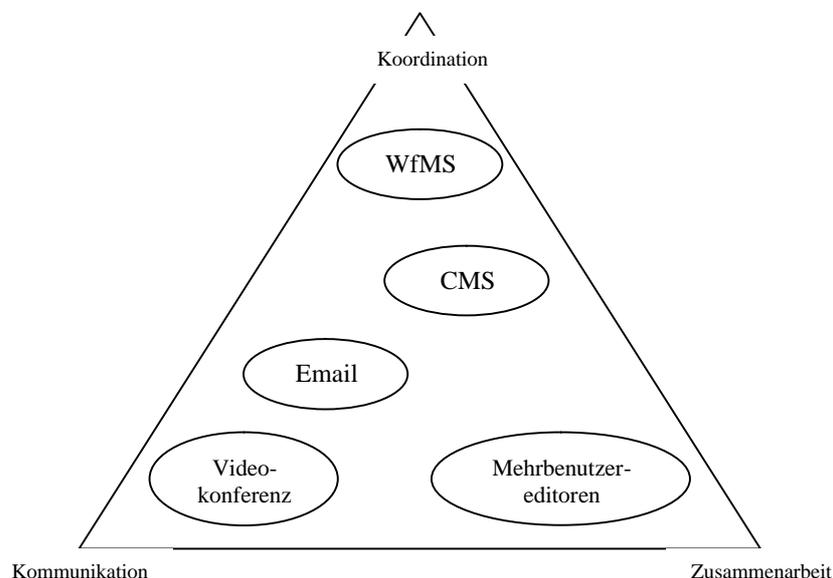


Abbildung 1-1: Einteilung von unterstützenden Softwaresystemen [LeRo00]

Ein WfMS ist ein umfassendes Softwaresystem, welches definierte Arbeitsabläufe (*Workflows*) verwaltet und ihre korrekte Ausführung informationstechnisch unterstützt. Ein Arbeitsablauf wird als *Prozess* definiert und als *Prozessvorlage* im *Prozessmodell* des WfMS hinterlegt. Jeder

Prozess besteht aus mehreren Knoten bzw. *Prozessschritten*, welche in einer vorgegebenen Reihenfolge abgearbeitet werden müssen. Um diese Abarbeitung zu starten, wird zur Laufzeit des WfMS eine *Prozessinstanz* von der Prozessvorlage gebildet. Die Laufzeitkomponente des WfMS ist dafür verantwortlich, die einzelnen Prozessschritte dieser Instanz zu aktivieren, zu überwachen und bei Beenden den Arbeitsablauf zum nächsten Prozessschritt weiterzuschalten. Aktivierte Prozessschritte – *Aktivitäten* – können dabei automatisch oder manuell ausgeführt werden, d. h. ihre Abarbeitung wird vom einer Systemkomponente oder einem Bearbeiter vorgenommen. Das WfMS muss diese Aktivitäten erkennen und entsprechend verteilen. Im Normalfall existieren viele verschiedene Prozessvorlagen und auf jeder Vorlage laufen mehrere Instanzen gleichzeitig. Das bedeutet, das WfMS muss in der Lage sein, alle Vorlagen zu hinterlegen und die einzelnen Prozessinstanzen zu unterscheiden und zu verwalten.

Obwohl Workflow-Management-Systeme ein relativ neues Forschungsgebiet darstellen, gibt es bereits mehrere kommerzielle Hersteller. Zu den erfolgreichsten Vertretern gehören IBM's WebSphere MQ Workflow [MQWf], die TIBCO iProcess Suite [TIBCO], Ultimus BPM [Ultimus] und COSA BPM [COSA]. Auch in der Forschung sind WfMS ein Thema. Zwei der größeren Projekte laufen momentan an der University of Georgia [METEOR] und an der Universität Ulm [ADEPT2].

Ein WfMS ermöglicht unter anderem die Erstellung und Überwachung von Arbeitsabläufen, kümmert sich um die Datenhaltung, startet andere Anwendungen zur Ausführung einer bestimmten Aktivität und sorgt dafür, dass einzelne Aktivitäten geeigneten Bearbeitern zugeordnet und in ihre Arbeitsliste gestellt werden. Die Vorteile liegen auf der Hand: geregelte Arbeitsabläufe, lückenlose Datenbereitstellung, weniger Verzögerungen bei der Abarbeitung von Arbeitsschritten und bessere Verteilung der Aufgaben.

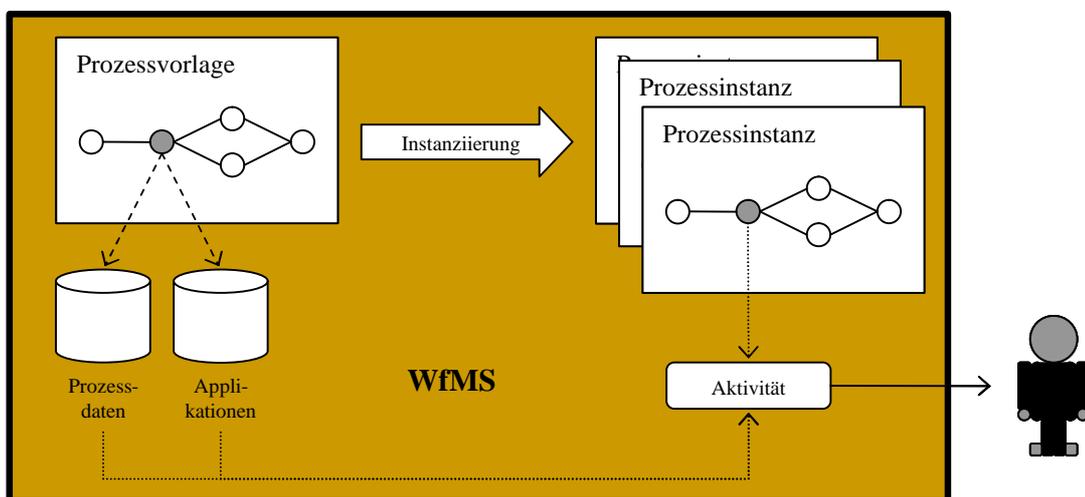


Abbildung 1-2: Arbeitsumgebung eines Workflow-Management-Systems

Am wichtigsten für den schnellen und reibungslosen Ablauf einer Aktivität ist die Effizienz der Arbeitsverteilung. Dazu wird für jeden einzelnen Prozessschritt eine bestimmte Menge von Bearbeitern definiert, die alle die Fähigkeiten und Möglichkeiten besitzen, diese Aufgabe optimal zu bearbeiten. Damit die Bearbeiter über die ihnen zugewiesenen Aufgaben informiert werden

und dabei nicht den Überblick verlieren, stellt das WfMS ihnen ein besonderes Mittel zur Verfügung: elektronische *Arbeitslisten*. Die Abbildungen 1-3 und 1-4 zeigen zwei unterschiedliche Darstellungen solcher Listen.

Name	Beschreibung	Priorität	Aktion
Blutabnahme	Patient: Herr Mueller	Normal	Starten
Röntgen	Patient: Herr Schmidt	Normal	Starten
Blutabnahme	Patient: Frau Meier	Kritisch	Starten
Blutabnahme	Patient: Frau Wolf	Niedrig	Starten
MRT	Patient: Frau Sieger	Hoch	Starten
Röntgen	Patient: Herr Vierse	Normal	Starten

Abbildung 1-3: Darstellung einer Arbeitsliste in einem einfachen HTML-Klienten

Name	Beschreibung	Priorität	Zustand	Komplexität	Deadline
Blutabnahme	Frau Meier	Kritisch	Reserviert	1	<1h
MRT	Frau Sieger	Hoch	Reserviert	5	3h
Blutabnahme	Herr Müller	Normal	Gestartet	1	<20min
Röntgen	Herr Schmidt	Normal	Frei	4	2h
Röntgen	Herr Vierse	Normal	Frei	4	<1h
Blutabnahme	Frau Wolf	Niedrig	Frei	1	<1h

Abbildung 1-4: Darstellung einer Arbeitsliste in einem Rich Client

Die Arbeitslisten enthalten alle Aktivitäten, die ein Bearbeiter aktuell bearbeiten kann. Die Instandhaltung der Arbeitslisten und ihre Übermittlung an den Bearbeiter übernimmt die *Arbeitslistenverwaltung* (ALV).

1.2 Aspekte der Arbeitslistenverwaltung

Die Verwaltung von Arbeitslisten birgt viele Aspekte. Jeder am System angemeldete Bearbeiter (das kann ein menschlicher Bearbeiter sein oder eine weitere Softwarekomponente) besitzt eine solche Liste, die ihm eindeutig zugeteilt ist. Alle aktivierten Prozessschritte werden vom WfMS an die Arbeitslistenverwaltung weitergeleitet und müssen zuerst daraufhin überprüft werden, welche Bearbeiter für ihre Ausführung in Frage kommen. Diese Zuordnungen werden über die Prozessvorlage definiert und im Prozessmodell des WfMS hinterlegt. Über das *Organisationsmodell*, in dem Informationen über hierarchische Strukturen und Mitarbeiter eines Unternehmens festgelegt sind, können die Bearbeiterzuordnungen aufgelöst werden.

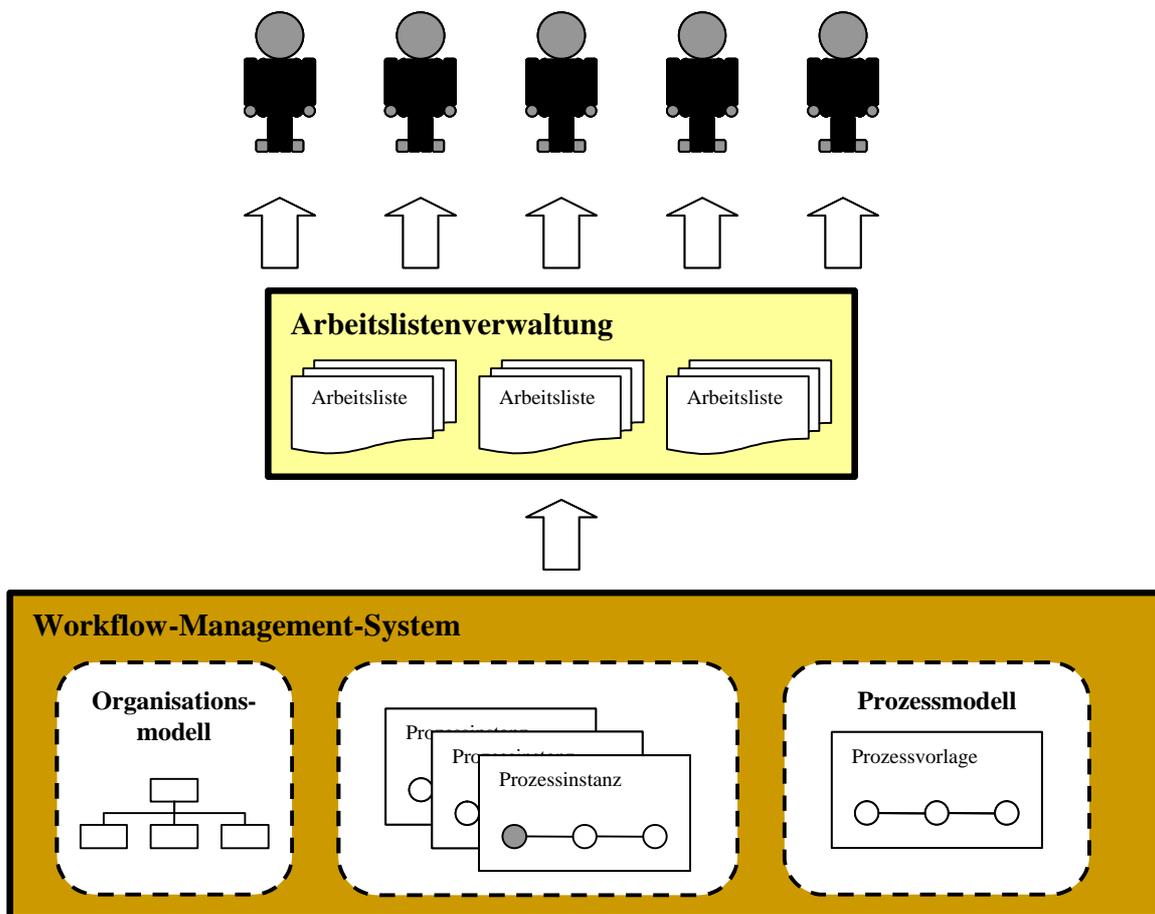


Abbildung 1-5: Die Arbeitslistenverwaltung als Verbindung zwischen WfMS und Klient

Ist die Bearbeitermenge erst einmal identifiziert, ist es die Aufgabe der ALV, die zugehörigen Arbeitslisten zu finden und die Aktivitäten dort hineinzustellen. Damit wird die Arbeitslistenverwaltung zum Bindeglied zwischen dem WfMS mit seinen Prozessen und Prozessschritten und dem Klienten, der eben diese Prozesse voranbringt, indem er die einzelnen Aktivitäten bearbeitet. Abbildung 1-5 verdeutlicht diese Verbindung, während Abbildung 1-6 veranschaulicht, wie sich die Arbeitsliste eines Benutzers aus verschiedenen Aktivitäten zusammensetzt.

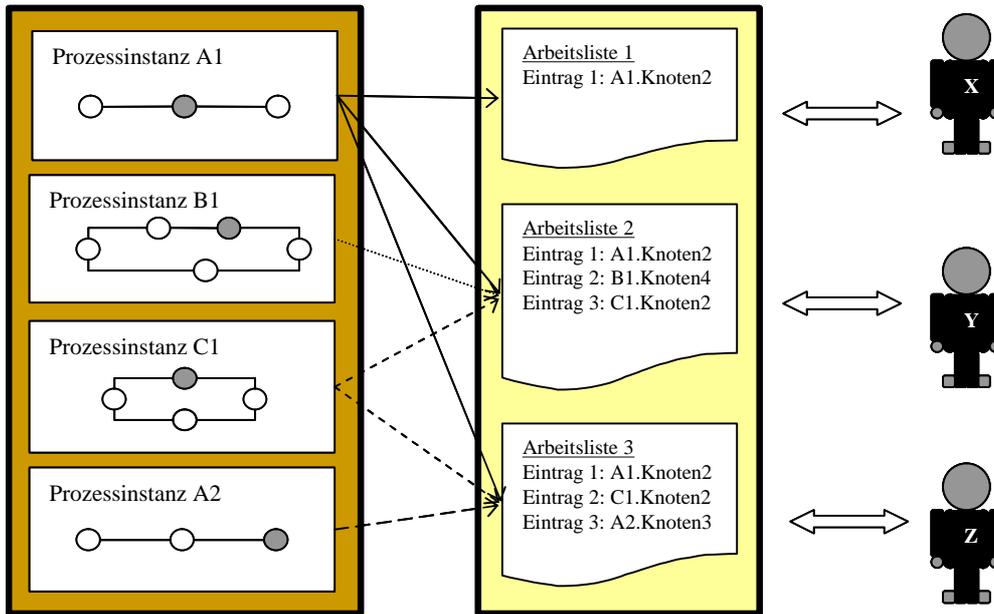


Abbildung 1-6: Verteilung der Aktivitäten auf die Arbeitslisten der Benutzer

Der Besitzer einer Arbeitsliste hat nun die Möglichkeit, eine Aktivität zur Bearbeitung auszuwählen. Dazu startet er sie über das WfMS, welches alle nötigen Daten zu Ausführung bereitstellt und eventuell das mit der Aktivität verknüpfte Anwendungsprogramm beim Bearbeiter öffnet. Das WfMS informiert wiederum die Arbeitslistenverwaltung, welche nun die Aufgabe hat, die anderen Bearbeiter über die Reservierung der Aktivität zu informieren. Das tut sie, indem sie den betreffenden Eintrag aus deren Arbeitslisten entfernt.

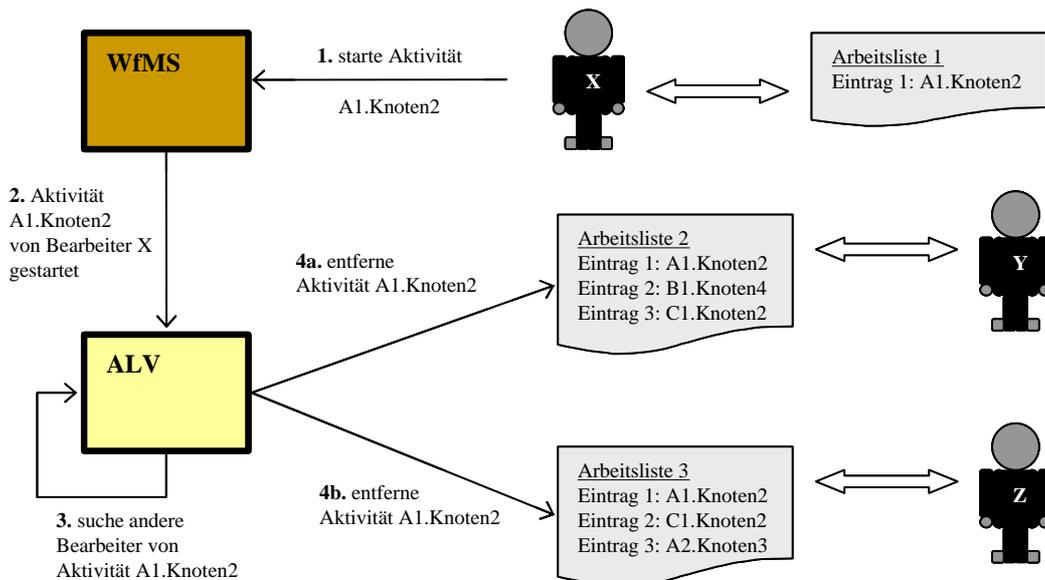


Abbildung 1-7: Starten und Entfernen von Aktivitäten aus den Arbeitslisten

Die Verteilung von Aktivitäten auf vordefinierte Bearbeitermengen und die Instandhaltung von Arbeitslisten sind wesentliche Aufgaben innerhalb eines WfMS. Aber um in der heutigen Zeit konkurrenzfähig zu sein reicht es nicht mehr aus, nur grundlegende Funktionen zu erfüllen. Und das Potential der Arbeitslistenverwaltung ist noch lange nicht ausgeschöpft.

Neben der reinen Zuteilung von Arbeitsaufgaben an 'alle' haben sich in den letzten Jahren andere Verteilungsverfahren erfolgreich in der Praxis bewährt. Algorithmen wie *Round Robin*, ein auslastungsregulierendes oder *Load Balancing*, ein auslastungsabhängiges Verteilungsverfahren sorgen dafür, dass die Bearbeiter gleichmäßig ausgelastet sind ohne überlastet zu sein. Nebenbei sorgen sie noch dafür, dass Aktivitäten schneller abgearbeitet werden können.

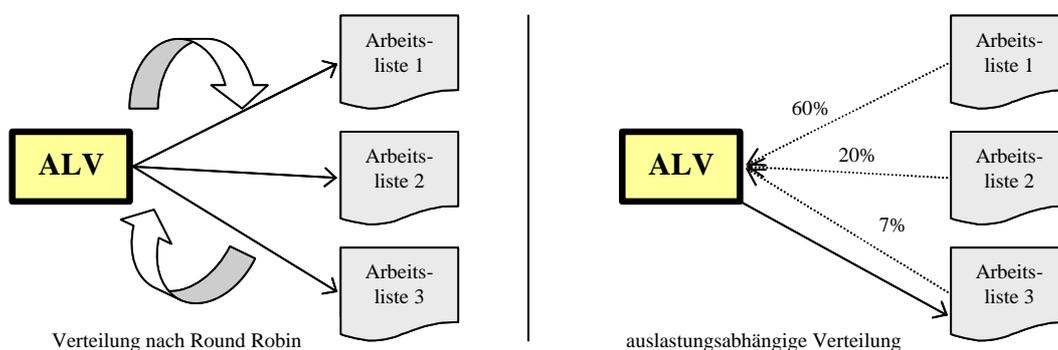


Abbildung 1-8: erweiterte Verteilungsverfahren

Geht die Abarbeitung von Aktivitäten immer noch nicht schnell genug, kann die Verwendung von Priorisierungen Abhilfe schaffen. Einzelne Aktivitäten werden dabei automatisch oder manuell mit einer höheren Priorität versehen, was die Bearbeiter dazu anregen soll, die Abarbeitung zu beschleunigen. Die Unterstützung von Priorisierungen und Prioritäten und ihre Weitergabe an den Klienten ist eine Aufgabe, die von der Arbeitslistenverwaltung vorgenommen werden kann.

Des Weiteren sind auch dynamische Veränderungen von Bearbeiterzuordnungen zur Laufzeit ein viel versprechendes Anwendungsgebiet, in dem eine ALV eingesetzt werden kann. Sei es bei einer Delegation oder Eskalation von Aktivitäten oder bei der Verwendung von Vertreterregeln.

Nicht zuletzt bringt es Vorteile, die Dienstleistung der Arbeitslistenverwaltung als eigenständige Komponente zu implementieren. Unabhängigkeit zum WfMS und zum Arbeitslistenklienten bedeutet, dass eine ALV mehrere verschiedene Systeme unterstützen kann, sowohl auf Seiten des WfMS als auch in Hinblick auf den Arbeitslistenklienten, der die Aufgabe hat, die Arbeitslisten beim Bearbeiter darzustellen. Von der Unterstützung eines einfachen *HTML-Klienten* bis zum voll funktionalen Klienten mit eigener Verarbeitungslogik (*Rich Client*) ist alles denkbar.

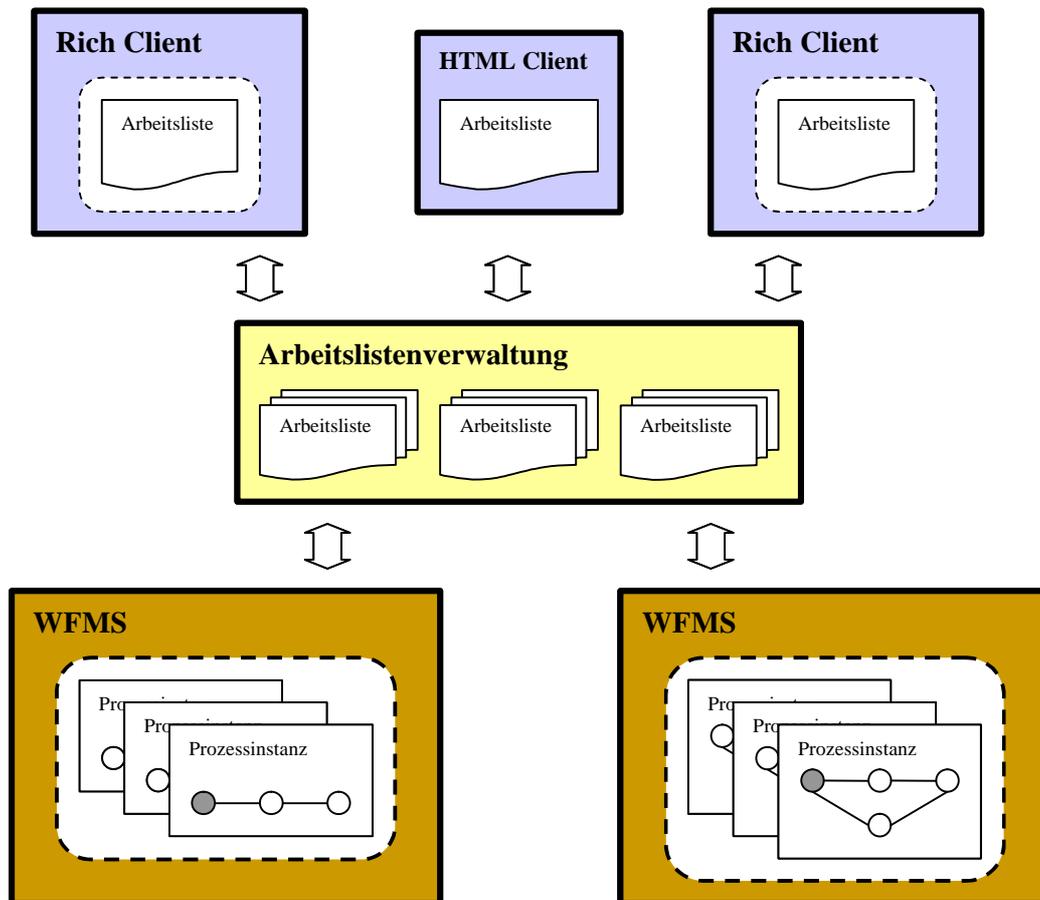


Abbildung 1-9: Die ALV als unabhängige Komponente

Auch die oben schon erwähnte zusätzliche Funktionalität der Arbeitslistenverwaltung kann so über selbstständige Komponenten in die ALV integriert werden. Komponentenbasierte und serviceorientierte Entwicklung von Workflow-Systemen sind zwei Schlagworte der aktuellen Forschungsarbeit. [ABLZ00, Atkin03]

1.3 Beispiel

In den folgenden Kapiteln werden einige Aspekte und Sachverhalte zum besseren Verständnis an konkreten Beispielen erklärt. Dazu werden nachfolgend Darstellungen und Erläuterungen eines Workflows und eines Organisationsmodells eingeführt, die in den Kapiteln durchgehend verwendet werden.

Abbildung 1-10 zeigt einen Ausschnitt aus dem Organisationsmodell einer Klinik. Die Stelle *Innere Medizin* wird dabei durch drei Rollen beschrieben. Es gibt einen *Chefarzt*, welcher die Stelle leitet. Er ist verantwortlich für die strukturellen und medizinischen Abläufe innerhalb seiner Abteilung. Ihm untergeben sind ein oder mehrere *Oberärzte*, welche sich um die

Behandlung der Patienten kümmern und zusammen mit dem Chefarzt mehrere *Assistenzärzte* ausbilden und anleiten.

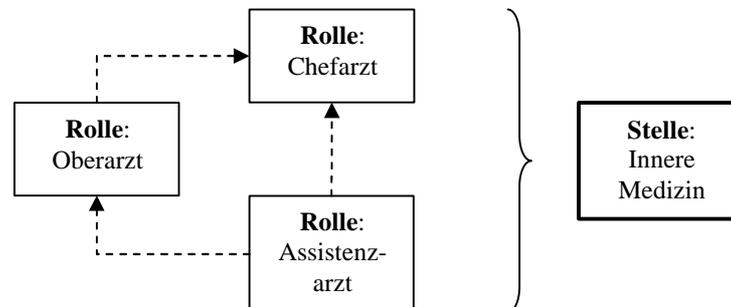


Abbildung 1-10: Ausschnitt aus dem Organisationsmodell einer Klinik

Abbildung 1-11 zeigt einen Workflow, wie er innerhalb der Abteilung für Innere Medizin vorkommen kann. Ein Patient kommt mit Beschwerden ins Krankenhaus. Nach einer körperlichen *Voruntersuchung* werden parallel *Röntgenbilder* und eine *Magnetresonanztomographie (MRT)* erstellt, um im nächsten Schritt eine *Diagnose* stellen zu können. Diese Aufgaben können von Assistenzärzten durchgeführt werden. Die gestellte Diagnose muss dann von einer zweiten Person, in der Regel von einem Oberarzt, *bestätigt* werden, damit zuletzt eine geeignete *Behandlung* vorgenommen werden kann.

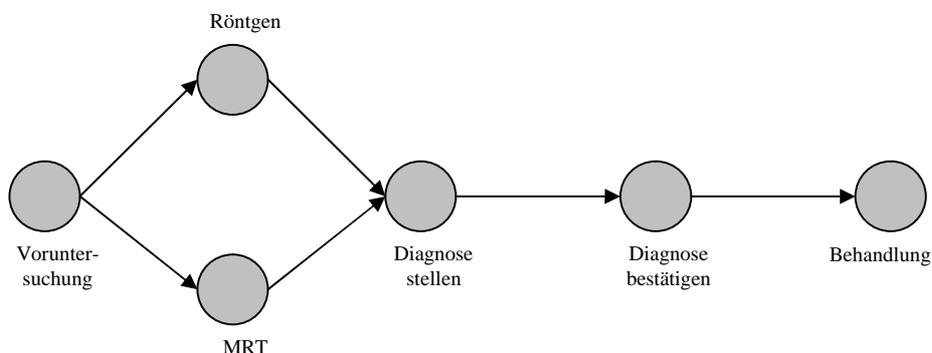


Abbildung 1-11: Workflow einer Untersuchung innerhalb der Abteilung für Innere Medizin

1.4 Aufgabenstellung und Zielsetzung

Im Rahmen dieser Arbeit werden Aspekte zur Erweiterung der Funktionalität der Arbeitslistenverwaltung untersucht. Dabei wird vor allem darauf eingegangen, welche Anwendungsgebiete sich für diese Aspekte im Rahmen einer ALV eröffnen und welche Anforderungen an die Arbeitslistenverwaltung selbst gestellt werden, sollten diese Erweiterungen übernommen werden. Für nachstehende Aspekte werden in den folgenden Kapiteln Konzepte erörtert:

- Verwaltung von Arbeitslisten
- Verfahren für die Verteilung von Aktivitäten
- Dynamische Änderung von Bearbeiterzuordnungen durch Delegation und Vertreterregeln
- Priorisierung von Aktivitäten

Des Weiteren wird auf der Grundlage einer komponentenbasierten Entwicklung eine Arbeitslistenverwaltung entworfen, deren Einsatz nicht auf ein WfMS beschränkt ist. Dafür werden die Schnittstellen zweier kommerzieller Systeme untersucht: WebSphere MQ Workflow [MQWf] und TIBCO iProcess Suite [TIBCO]. Das Ziel dabei ist, eine generische Schnittstelle für die Anbindung der ALV zu erstellen.

Das Prinzip einer Komponentenarchitektur verlangt ebenfalls, dass eine Arbeitslistenverwaltung in der Lage ist mehrere Arbeitslistenkunden zu versorgen. Hierbei muss untersucht werden, wie viel Funktionalität die ALV mitbringen muss, um die verschiedensten Klientensysteme zu unterstützen.

Ebenfalls von Bedeutung ist die Möglichkeit externe Komponenten mit der Arbeitslistenverwaltung zu verknüpfen, welche die gegebene Funktionalität noch erweitern. Das gilt vor allem in Bezug auf die in der Konzeption betrachteten Aspekte Verteilungsverfahren, Delegation und Priorisierung.

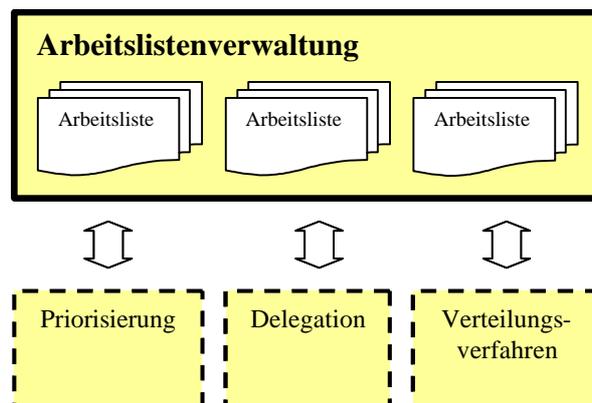


Abbildung 1-12: Komponenten zur Erweiterung der Funktionalität

Ziel dieser Arbeit ist es, eine eigenständige Komponente für die Arbeitslistenverwaltung zu entwerfen. Dafür sollen Konzepte und Schnittstellen erstellt werden, welche die Funktionalität der ALV erweitern und eine Integration beliebiger Systeme möglich machen. Die Ergebnisse sollen auf andere ALVs und WfMS übertragbar sein, so dass diese optimiert werden können.

1.5 Aufbau der Arbeit

Diese Arbeit gliedert sich in zwei grundlegende Teile: die Untersuchung der einzelnen funktionalen Aspekte und der Entwurf der eigentlichen Komponente. Kapitel 2 analysiert die Anforderungen, die an eine Arbeitslistenverwaltung gestellt werden und ordnet die Komponente

in den Bereich eines WfMS ein. Kapitel 3 beschäftigt sich mit der Bearbeiterzuordnung innerhalb der ALV. In Kapitel 4 werden Verfahren für die Verteilung von Aktivitäten auf Bearbeiter vorgestellt und diskutiert, während Kapitel 5 sich mit der dynamischen Neuverteilung von Aktivitäten beschäftigt. Dort wird auch die Delegation von Aktivitäten beziehungsweise die Anwendung von Vertreterregeln untersucht. Kapitel 6, die Priorisierung von Aktivitäten, bildet den Abschluss der konzeptionellen Analysen. In Kapitel 7 wird der Entwurf des Systems vorgenommen. Hier werden die Schnittstellen der Arbeitslistenverwaltung erstellt. Kapitel 8 bildet den Abschluss und gibt einen Ausblick auf weiterführende Themen, die im Rahmen dieser Arbeit nicht behandelt wurden. Kapitel 9 fasst schließlich die gewonnenen Erkenntnisse zusammen.

2 Anforderungen an die Arbeitslistenverwaltung

Dieses Kapitel gibt einen Überblick darüber, welche Anforderungen an eine Arbeitslistenverwaltung gestellt werden. Dabei wird besonders auf erweiterte Funktionalität und die Implementierung als eigenständige Komponente eingegangen. In Abschnitt 2.1 werden die internen Mechanismen einer Arbeitslistenverwaltung betrachtet und ihre Arbeitsweise allgemein beschrieben. Daraus werden Rückschlüsse auf die Anforderungen an die ALV gezogen. In Abschnitt 2.2 wird dann die Umgebung der angehenden Komponente genauer untersucht, wobei als Ausgangspunkt das Workflow-Referenzmodell der Workflow Management Coalition herangezogen wird. Anhand dessen können Anforderungen an die Schnittstellen der Arbeitslistenverwaltung zum WfMS und zum Arbeitslistenklienten deutlich gemacht werden.

2.1 Erweiterte Funktionalität der Arbeitslistenverwaltung

Die Arbeitslistenverwaltung ist hauptsächlich dafür verantwortlich, eingehende Aktivitäten den betreffenden Bearbeitern zuzuweisen und in deren Arbeitslisten zu stellen. Dazu muss die Bearbeitermenge zuerst aufgelöst werden, was in der Regel über das Organisationsmodell geschieht. Die Arbeitslisten selbst müssen in einer ressourcenschonenden Weise in der ALV hinterlegt und für den Klienten zugänglich gemacht werden. Die Anforderungen an die Arbeitslistenverwaltung liegen bei der Bearbeiterzuordnung in folgenden Bereichen:

- Hinterlegung und Auflösung der Bearbeiterzuordnungsvorschriften
- Hinterlegung und Pflege der Arbeitslisten
- Ermöglichung eines effizienten Zugriffs auf Arbeitslisten durch den Klienten
- Kommunikation mit dem Klienten bei Aktualisierung der Arbeitslisten

Die eigentliche Verteilung der Aktivitäten auf die einzelnen Mitarbeiter kann über verschiedene Verteilungsverfahren vorgenommen werden. Diese können unter anderem auch dazu verwendet werden, in gewissen Situationen eine Neuverteilung der Aktivitäten zu veranlassen. Dynamische Änderungen der Bearbeiterzuordnung ziehen ebenfalls eine Neuverteilung nach sich. Zum Beispiel können über eine Delegation Aktivitäten von der eigenen Arbeitsliste in eine andere verschoben werden. Jedem Delegationsauslöser steht dabei eine beschränkte Menge von Delegationsempfängern gegenüber. Vertreterregeln können eingesetzt werden, um bei Bedarf kurzfristig Bearbeiterzuordnungsvorschriften zu umgehen. Die Anforderungen an die Arbeitslistenverwaltung sehen folgendermaßen aus:

- Einbindung von Anwendung von Verteilungsverfahren
- Regelung der Voraussetzungen einer dynamischen Neuverteilung
- Bestimmung von Delegationsempfängern abhängig vom Auslöser oder der Aktivität
- Behandlung der Auswirkungen einer Delegation auf die betroffenen Arbeitslisten

Eine Priorisierung von Aktivitäten erfolgt in den meisten Fällen automatisch, kann aber auch manuell ausgelöst werden. Dabei werden die betroffenen Aktivitäten gekennzeichnet, damit ihre Bearbeitung im Idealfall schneller vonstatten geht. Die Arbeitslistenverwaltung hat dabei folgende Aufgaben:

Das Referenzmodell identifiziert innerhalb eines Workflow-Management-Systems eine Kernkomponente: den Workflow Enactment Service. Er enthält eine oder mehrere Workflow Engines, welche Prozessdefinitionen interpretieren, die Instanziierung und den Ablauf von Prozessen überwachen, Arbeitslisten erstellen und wenn nötig externe Applikationen aufrufen. Des Weiteren verfügt der Workflow Enactment Service über eine Reihe von Datensätzen, wie z. B. das Organisationsmodell (Organisational Model) oder Daten zur Steuerung eines Workflows (Workflow Control Data). Weiterhin definiert das Referenzmodell eine Reihe von Komponenten, die mit der Kernkomponente interagieren. Darunter befindet sich auch die Arbeitslistenverwaltung (Worklist Handler). [WfRM95]

Das Workflow-Referenzmodell und die Standards der WfMC werden in der Praxis selten realisiert. [VdA03, STO99, RRD03c] Aber das Referenzmodell gibt Aufschluss darüber, welche Komponenten an einem WfMS beteiligt sind und wie sie miteinander interagieren. Das Hauptaugenmerk dieser Arbeit liegt auf der Komponente für die Arbeitslistenverwaltung. Sie gehört nicht zum Kern eines WfMS, sondern bildet eine Vermittlungsstelle zwischen dem Workflow-System und dessen Benutzern. In Abbildung 2-2 wird diese Verbindung dargestellt.

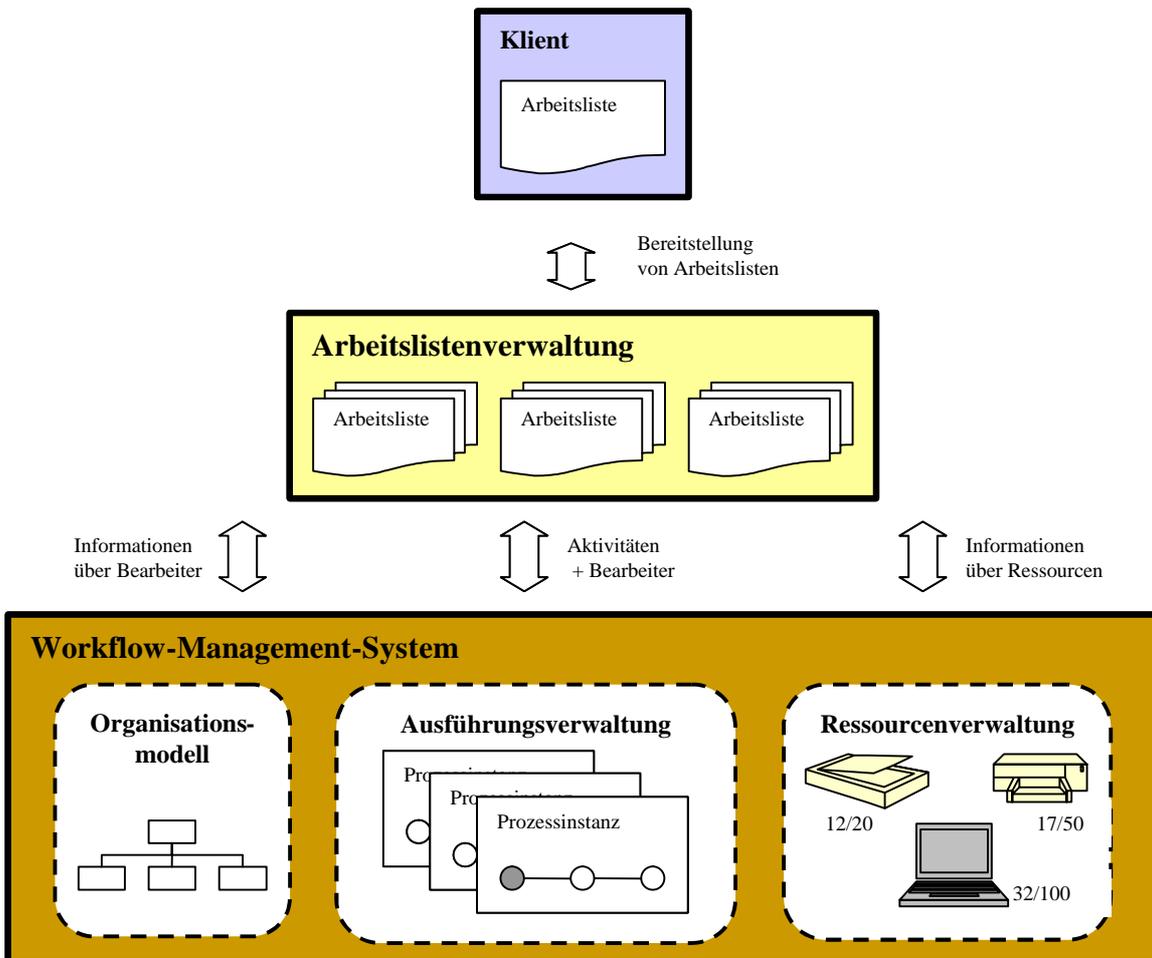


Abbildung 2-2: Die Verbindung der ALV zu WfMS und Klient

Die Arbeitslistenverwaltung bekommt alle notwendigen Daten, die sie für ihre Arbeit braucht vom WfMS. Die aktivierten Prozessschritte und deren Bearbeiterzuordnungen werden von der Ausführungsverwaltung (dem *Execution Manager*) übermittelt. Diese ist für die Ausführung und Weitschaltung aller Prozessinstanzen verantwortlich. Über das Organisationsmodell können Bearbeiterzuordnungen aufgelöst und Anfragen zu den einzelnen Bearbeitern gestellt werden. Eine *Ressourcenverwaltung* stellt Informationen über die Verfügbarkeit von Ressourcen, wie zum Beispiel Drucker, Kopierer oder andere Hilfsmittel zur Verfügung.

Neben der Arbeitslistenverwaltung identifiziert das Referenzmodell auch noch eine User-Interface-Komponente auf der Seite des Klienten, die dafür zuständig ist, die Arbeitslisten beim Benutzer anzuzeigen. Wie diese (graphische) Benutzerschnittstelle letztendlich aussieht liegt nicht im Ermessen der ALV. Diese ist nur dafür verantwortlich, dem Klienten Informationen über aktuell ausführbare Aktivitäten zu übermitteln.

3 Bearbeiterzuordnung

Für jeden einzelnen Prozessschritt eines Prozesses ist vorgegeben, wer ihn bearbeiten darf. In der Regel geschieht das in Form von Zuordnungsvorschriften. Eine solche Zuordnung muss zu einer Bearbeitermenge aufgelöst werden, bevor die Arbeitslistenverwaltung die Aktivität verteilen kann. Bei der Verteilung werden die Aktivitäten in die Arbeitslisten der jeweiligen Bearbeiter eingetragen. Diese Arbeitslisten werden in der ALV hinterlegt und von ihr fortwährend aktualisiert. Zusammen mit den Aktivitäten gehören die Arbeitslisten zu den wichtigsten Ressourcen der ALV und haben damit großen Einfluss auf die Performanz. Die Übermittlung der aktualisierten Arbeitslisten an den Klienten kann von der Arbeitslistenverwaltung ausgelöst werden (*Push*) oder vom Klienten selbst (*Pull*).

Dieses Kapitel beschäftigt sich mit Untersuchungen hinsichtlich der Auflösung und Hinterlegung von Bearbeiterformeln und der Vorratshaltung und Übergabe der Arbeitslisten. In Abschnitt 3.1 wird diskutiert, wie und wo man Bearbeiterformeln am besten auflöst, während Abschnitt 3.2 untersucht, wo sie am günstigsten hinterlegt werden können. Die folgenden beiden Abschnitte 3.3 und 3.4 erörtern die Verwaltung der Arbeitslisten durch die ALV und ihre Übergabe an den Klienten durch synchrone oder asynchrone Kommunikation. In Abschnitt 3.5 wird eine Zusammenfassung über die Erkenntnisse des vorliegenden Kapitels gegeben.

3.1 Auflösung der Bearbeiterformeln

Bei der Modellierung und Definition eines Prozesses wird festgelegt, welche Bearbeiter für die Ausführung einer Aktivität in Frage kommen. Zum Beispiel kann eine Voruntersuchung in der Abteilung für Innere Medizin von verschiedenen Assistenzärzten oder auch von einem Oberarzt vorgenommen werden. Diese Informationen sind in der Regel als eine Reihe verknüpfter oder alternativer Bearbeiterzuordnungen, so genannten *Bearbeiterformeln* (BF), hinterlegt. Sie werden im Prozessmodell des WfMS, bei der Prozessvorlage gespeichert und bei der Instanziierung des Prozesses in der Prozessinstanz hinterlegt. Bei der Aktivierung eines Prozessschrittes werden die Bearbeiterformeln dann an die ALV übergeben.

Bevor die Arbeitslistenverwaltung eine Bearbeiterformel verwenden kann, muss diese aufgelöst werden. Das geschieht über das Organisationsmodell des WfMS, welches im Idealfall als weitgehend autonome Komponente innerhalb des Workflow-Systems realisiert ist. [Atkin03] Die Aufgabe des Organisationsmodells ist es, eine Bearbeiterformel in eine Menge potentieller Bearbeiter aufzulösen. Es enthält Informationen über Organisationseinheiten, wie z. B. Stellen und Rollen innerhalb eines Unternehmens. Die einzelnen Mitarbeiter sind einer oder mehrerer dieser Organisationseinheiten zugewiesen. Eine Bearbeiterformel enthält keine konkreten Bearbeiter, sondern bestimmt für eine Aktivität, von welcher Stelle oder Rolle sie ausgeführt werden kann. Zum Beispiel kann eine Diagnose im Krankenhaus nur von Chef-, Ober- oder Assistenzärzten gestellt werden. Das Organisationsmodell hat dann die Aufgabe, für die angegebene Organisationseinheit, z.B. die Gruppe der Assistenzärzte, alle betroffenen Mitarbeiter zu ermitteln und diese als Ergebnis der Auflösung zurückzugeben.

Bearbeiterformeln können abhängig sein, d. h. die Entscheidung, welcher Bearbeiter eine Aktivität bearbeiten darf, wird von bereits in der Vergangenheit liegenden Umständen beeinflusst. Zum Beispiel ist es sinnvoll festzulegen, dass die Bestätigung einer Diagnose nicht von dem Bearbeiter vorgenommen werden darf, der die Diagnose selbst gestellt hat. Diese Abhängigkeiten müssen vor der eigentlichen Auflösung der Bearbeiterformeln analysiert und beseitigt werden.

Die Bearbeiterformeln sind die wichtigsten Bestandteile einer Bearbeiterzuordnung. Die Bestimmung, wer Bearbeiterformeln letztendlich auflöst, hat Einfluss auf den Kommunikationsaufwand, die Performanz und die Funktionalität der Arbeitslistenverwaltung.

Es gibt generell zwei Möglichkeiten die Bearbeiterformeln aufzulösen: über die Arbeitslistenverwaltung oder über das WfMS. Beide Varianten haben ihre Vor- und Nachteile. Um eine Entscheidung zu treffen müssen einige Aspekte untersucht werden.

Einsatzmöglichkeiten der Bearbeiterformeln

Neben der eigentlichen Funktion der Bearbeiterformel – der Zuordnung einer Aktivität zu konkreten Bearbeitern – kann diese auch noch für andere Zwecke verwendet werden. So ist es denkbar, sie in der Delegation oder Eskalation von Aktivitäten einzusetzen, indem man deren Empfänger als Bearbeiterformeln hinterlegt. Diese können dann für einen beliebigen Mitarbeiter angelegt werden, indem man schaut, wo er innerhalb der Organisationshierarchie steht und wer sein direkter Vorgesetzter ist (Eskalation), beziehungsweise welche Mitarbeiter er unter sich hat (Delegation). Für einen Oberarzt ergibt sich somit als Eskalationsempfänger der Chefarzt und als Delegationsempfänger die Menge der Assistenzärzte. Genauso ist es möglich Delegationsempfänger für eine bestimmte Aktivität anzulegen, z. B. den Leiter oder eine bestimmte Rolle innerhalb der Stelle, wo die Aktivität bearbeitet wird. Bei Komplikationen während einer MRT wird die Aktivität dann beispielsweise an den Chefarzt der Abteilung für Innere Medizin delegiert.

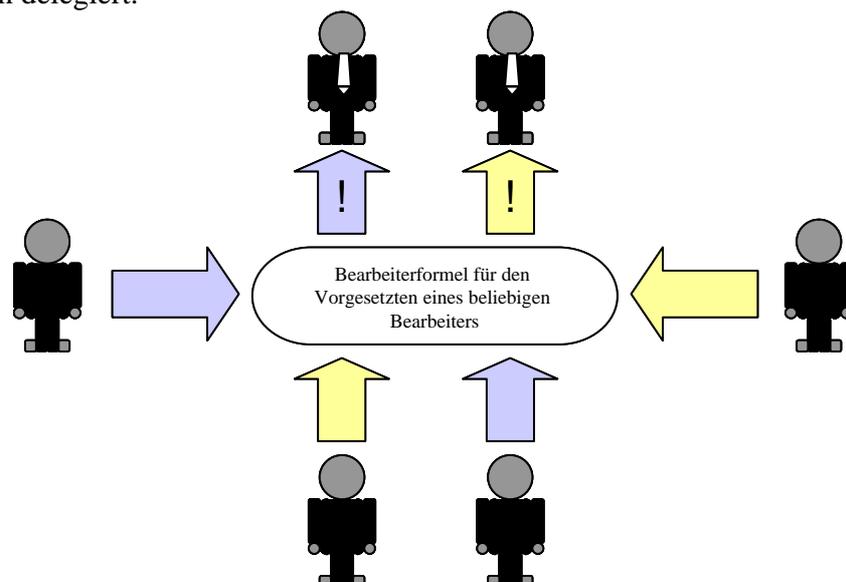


Abbildung 3-1: Auflösung von Eskalationsempfängern über eine Bearbeiterformel

Ein weiteres potentielles Anwendungsfeld für den Einsatz von Bearbeiterformeln ist die Performanzverbesserung im Bereich der Datenhaltung. Anstatt Arbeitslisten für jeden einzelnen Bearbeiter vorrätig zu halten, kann man eine gruppenbezogene Arbeitsliste erstellen und jedem Bearbeiter der Gruppe zuweisen. Anstatt einer Arbeitsliste für jeden Assistenzarzt gibt es dann nur noch eine gemeinsame Gruppenarbeitsliste, auf die alle Assistenzärzte Zugriff haben.

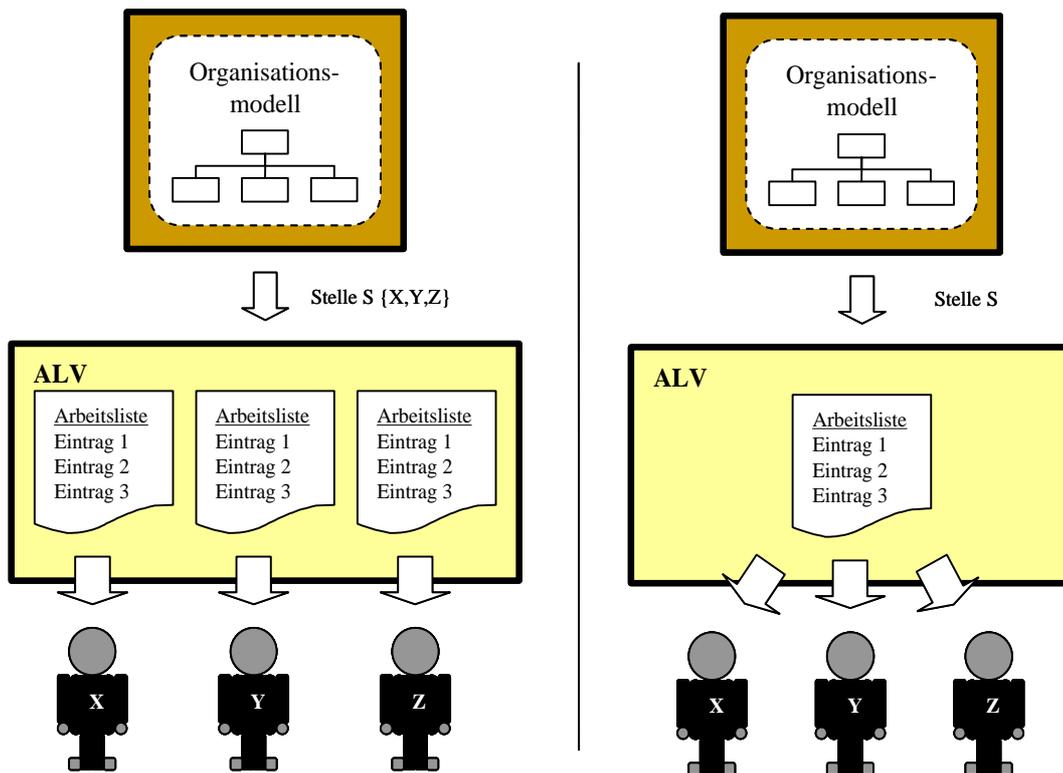


Abbildung 3-2: persönliche und gruppenbezogene Arbeitslisten

Eine dritte Einsatzmöglichkeit ist die Verwendung von Bearbeiterformeln in Verteilungsverfahren. Wenn ein Verfahren nicht global innerhalb der gesamten ALV angewendet wird, sondern sich auf einzelne Aktivitäten beschränkt, lohnt es sich, die Auswahl der Bearbeiter über extra dafür entwickelte Bearbeiterformeln zu bestimmen.

Beispielsweise ist es denkbar, dass ein Krankenhaus in bestimmten Zeiten stark mit Arbeit überlastet ist, z. B. im Hochsommer oder in der Wintersaison. Um die Überlastung der Mitarbeiter in solchen Situationen abzuschwächen, können Zeit- oder Ferienarbeiter eingestellt werden. Diese sind meist unerfahren und werden für die kurze Zeit ihres Einsatzes nur wenig geschult. Infolge dessen können sie nur die einfachsten Arbeiten mit geringer Komplexität übernehmen, z. B. das Säubern von Hilfsmitteln oder das Abtippen von Berichten. Wird eine solche Aktivität vom WfMS an die ALV übergeben, löst diese nicht die mitgelieferte Originalbearbeiterformel auf, sondern eine speziell für diesen Einsatz entwickelte Sonderformel. Diese verweist dann z. B. nicht auf eine bestimmte aktivitätsabhängige Stelle, sondern liefert alle Mitarbeiter mit der Fähigkeit oder dem Status 'Zeitarbeiter' zurück. Die Aktivität wird also nicht an die ursprüngliche Stelle ausgeteilt, sondern an einen oder mehrere der Hilfskräfte.

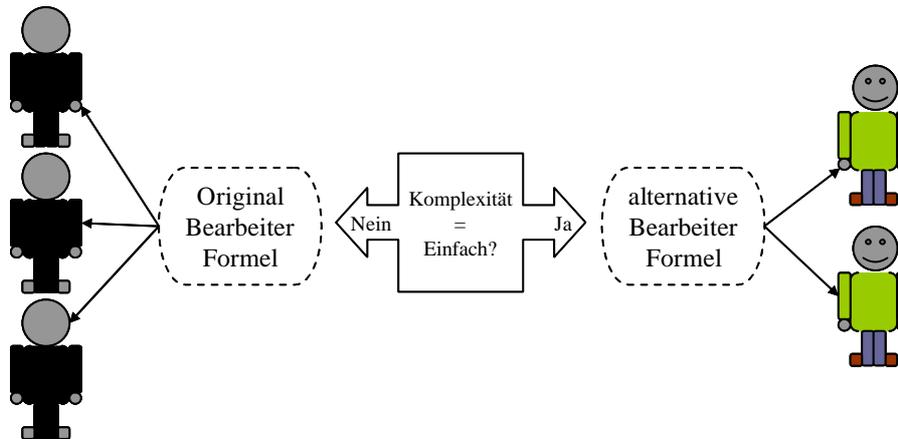


Abbildung 3-3: Verwendung alternativer Arbeiterformeln

Kommunikationsaufwand bei Auflösung der Arbeiterformeln

Die Auflösung der Arbeiterformeln muss in jedem Fall über das Organisationsmodell laufen, welches sich in der Regel innerhalb des WfMS befindet. Je nachdem welche Komponente die Auflösung vornimmt, erfordert das mehr oder weniger Kommunikationsaufwand und Datenverkehr.

Ressourcenverbrauch bei der Auflösung von Arbeiterformeln

Eine einzelne Arbeiterformel verbraucht im Normalfall weitaus weniger Speicher als eine aufgelöste Arbeitermenge, die unter Umständen aus mehreren Duzend Arbeitern besteht. Dem gegenüber steht die Zeit, die zusätzlich benötigt wird, wenn die Arbeiterformeln bei jeder Verwendung aufgelöst werden müssen. Daher ist abzuwägen was wichtiger ist: weniger Speicherverbrauch oder schnellerer Datenzugriff.

Auflösung der Arbeiterformeln über die ALV

Die Auflösung der Arbeiterformeln ist für die Arbeitslistenverwaltung mit einigem Aufwand verbunden. Das WfMS muss die Arbeiterformeln zusammen mit den Aktivitäten übergeben und die ALV muss sich dann selbstständig an das Organisationsmodell wenden, um sie aufzulösen.

Die Auflösung der Arbeiterformeln über die ALV verlangt das Vorhandensein entsprechender Funktionen und den Zugriff auf das Organisationsmodell über definierte Schnittstellen. Der Vorteil dabei ist, dass diese Schnittstelle von der Arbeitslistenverwaltung verwendet werden kann, um Arbeiterformeln auch zu anderen Zwecken – wie den oben beschriebenen – selbst zu hinterlegen und bei Bedarf aufzulösen. Der Nachteil ist, dass die Schnittstelle zum Organisationsmodell im Rahmen der Unabhängigkeit der ALV möglichst generisch sein muss, um verschiedene Modelle ansprechen zu können. Außerdem ist ein Zugriff auf das

Organisationsmodell – eine WfMS-interne Komponente – unter Umständen sicherheitsrelevant und muss vom Workflow-System erst gestattet werden.

Eine Auflösung über die Arbeitslistenverwaltung bedeutet ebenfalls, dass bei jeder Übergabe einer Aktivität mehrmals zwischen ALV und WfMS kommuniziert wird. Das WfMS übergibt die Aktivität und die Bearbeiterformel. Die ALV sendet die Bearbeiterformel an das Organisationsmodell und dieses wiederum übermittelt die aufgelöste Bearbeitermenge zurück an die ALV. Liegen das WfMS und die ALV nicht zentral auf einem Rechner, müssen diese Daten über entfernte Aufrufe übertragen werden, was einen hohen Kommunikationsaufwand verursachen kann. Außerdem steigt das Volumen des Datenverkehrs an, da mehrere Informationen zwischen den beiden Komponenten ausgetauscht werden müssen.

Durch die Schnittstelle zum Organisationsmodell muss die Arbeitslistenverwaltung nicht die aufgelösten Bearbeitermengen vorrätig halten, wodurch der Speicherplatzbedarf verringert wird. Bei Bedarf können die Bearbeiterformeln erneut aufgelöst werden. Das wiederum erhöht die Zugriffszeiten. Da die Arbeitslistenverwaltung aber eine Aktivität in der Regel nur einmal auf die Bearbeitermenge verteilt und diese danach wieder verwirft, ist es günstiger, die höhere Zugriffszeit zu Gunsten des besseren Ressourcenverbrauchs in Kauf zu nehmen.

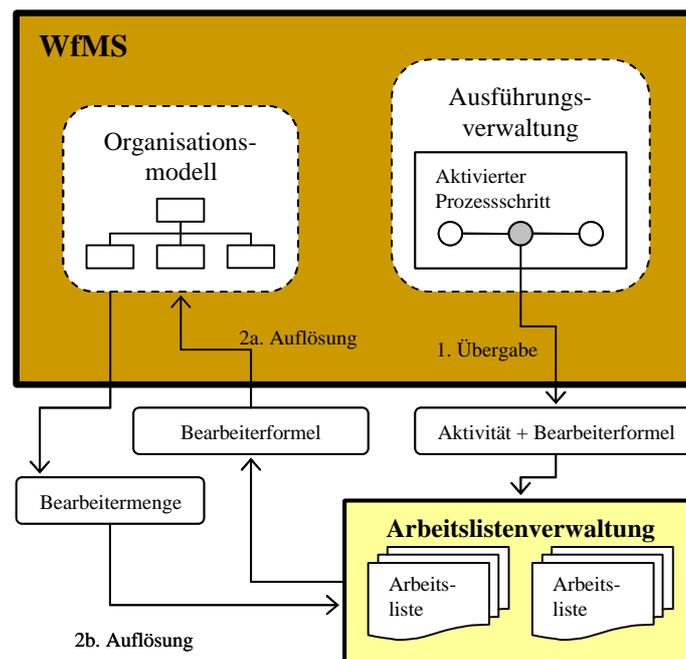


Abbildung 3-4: Auflösung der Bearbeiterformeln über die ALV

Auflösung der Bearbeiterformeln über das WfMS

Zur Entlastung der Arbeitslistenverwaltung und zur Vermeidung zusätzlicher Schnittstellen zum Organisationsmodell können die Bearbeiterformeln auch selbstständig vom WfMS aufgelöst

werden. Dieses übergibt dann die aufgelöste Bearbeitermenge zusammen mit der Aktivität an die ALV.

Wenn die Bearbeiterformeln schon im Vorhinein vom WfMS aufgelöst werden, benötigt die Arbeitslistenverwaltung keine speziellen Schnittstellen zum Organisationsmodell und hat somit keinen Zugriff auf WfMS-interne Daten. Der Nachteil dabei ist, dass die ALV damit keine Möglichkeit hat, andere Bearbeiterformeln, wie z. B. für Delegationsempfänger aufzulösen. Dadurch können Bearbeiterformeln für diese Einsatzgebiete nicht verwendet werden.

Bei einer Auflösung der Bearbeiterformeln durch das Workflow-System werden die Daten, d. h. die Aktivität und die Bearbeitermenge nur einmal übertragen und dann in der ALV hinterlegt. Damit entfällt die Anfrage an das Organisationsmodell, wodurch weniger Datenverkehr erzeugt wird. Durch die einmalige Verbindung bei Übergabe der Aktivität wird auch der Kommunikationsaufwand verringert.

Wenn die ALV nicht die Möglichkeit hat, Bearbeiterformeln selbst aufzulösen, muss sie stattdessen die aufgelösten Bearbeitermengen vorrätig halten. Dadurch wird der Ressourcenverbrauch vergrößert. Der Vorteil ist allerdings, dass dadurch die Zugriffszeiten auf die Bearbeitermenge verkürzt werden und diese bei Bedarf schnell zur Verfügung steht.

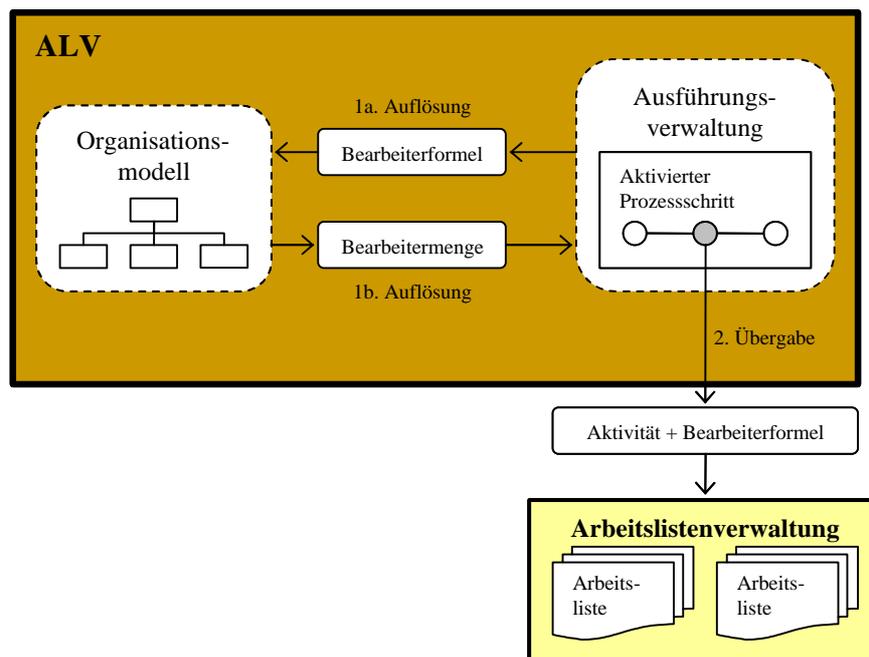


Abbildung 3-5: Auflösung der Bearbeiterformeln über das WfMS

Betrachtet man die vielen Einsatzmöglichkeiten der Bearbeiterformeln in der ALV ist es sinnvoll, dieser eine Schnittstelle zum Organisationsmodell zur Verfügung zu stellen. Dadurch werden zwar der Datenverkehr und der Kommunikationsaufwand gesteigert, aber durch die Bearbeiterformeln werden Anwendungsgebiete ermöglicht, die die Funktionalität der Arbeitslistenverwaltung sinnvoll erweitern.

3.2 Hinterlegung der Bearbeiterformeln

Wenn die Arbeitslistenverwaltung die Auflösung der Bearbeiterformeln übernimmt, stellt sich die Frage, ob es eventuell sinnvoll ist, die Formeln auch gleich in der ALV zu hinterlegen. Um hier eine Entscheidung zu treffen muss man untersuchen, von wem Bearbeiterformeln zusätzlich noch verwendet oder manipuliert werden können. Neben der Arbeitslistenverwaltung sind das vor allem die Prozessmodellierer, welche die Formel während der Prozessdefinition erstellen oder bei Änderungen entsprechend anpassen.

Eine erste Möglichkeit ist, die Bearbeiterformeln im Prozessmodell zu verankern, d. h. sie sowohl in der Prozessvorlage als auch in der Prozessinstanz zu hinterlegen. Das hat den Vorteil, dass die Formeln genau dort liegen, wo sie logisch gesehen hingehören, da sie Teil der Prozessbeschreibung sind. Sie können schnell von der Prozessvorlage zur Prozessinstanz übertragen und bei Änderungen angepasst werden. Das spart Kommunikationsaufwand. Außerdem verbessert es die Performanz, da durch die direkte Verknüpfung mit dem Prozessschritt ein schnellerer Zugriff auf bestimmte Bearbeiterformeln möglich ist. Zuletzt ist es auch für die ALV von Vorteil, nicht etliche Bearbeiterformeln vorrätig halten zu müssen, für die momentan kein Prozessschritt aktiv ist.

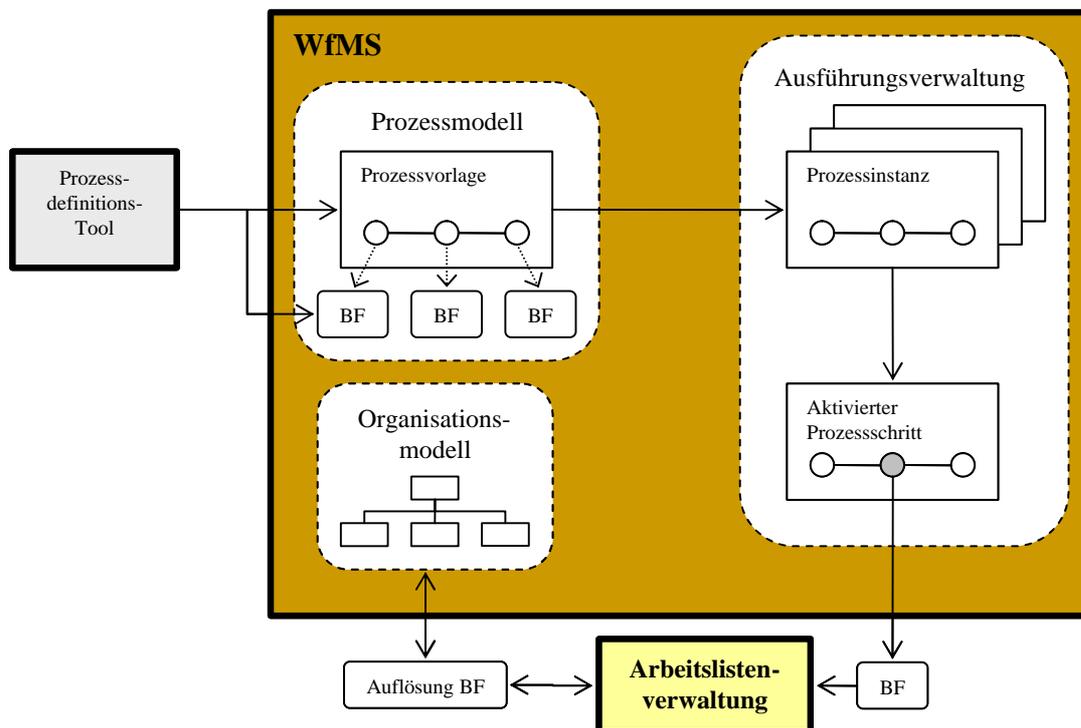


Abbildung 3-6: Hinterlegung der Bearbeiterformeln im Prozessmodell

Die zweite Möglichkeit ist, die Bearbeiterformeln in der Arbeitslistenverwaltung zu hinterlegen. Damit muss das WfMS bei aktiviertem Prozessschritt nur noch die Aktivität übergeben und die ALV sucht dann selbstständig nach der verknüpften Bearbeiterformel. Dagegen spricht, dass

Änderungen der Formeln problematisch sind, weil Prozessentwickler erst über Umwege Zugriff auf die Originalformeln bekommen. Eine entsprechende Schnittstelle muss extra eingerichtet werden und führt zu einem höheren Kommunikationsaufwand zwischen Komponenten die eigentlich nichts miteinander zu tun haben, wie z. B. das Werkzeug zur Prozessdefinition und die ALV. Außerdem muss die Arbeitslistenverwaltung wie schon beschrieben Bearbeiterformeln für jede Prozessvorlage vorrätig halten, ob davon Instanzen existieren oder nicht, was sich negativ auf die Performanz auswirkt.

Abhängige Bearbeiterformeln erfordern Informationen über Prozessabläufe, die der ALV nicht zugänglich sind. Daher müssen diese Abhängigkeiten vor der Auflösung einer Bearbeiterformel beseitigt werden. Dafür wird eine weitere Schnittstelle zum WfMS benötigt, wodurch sich wiederum der Kommunikationsaufwand vergrößert.

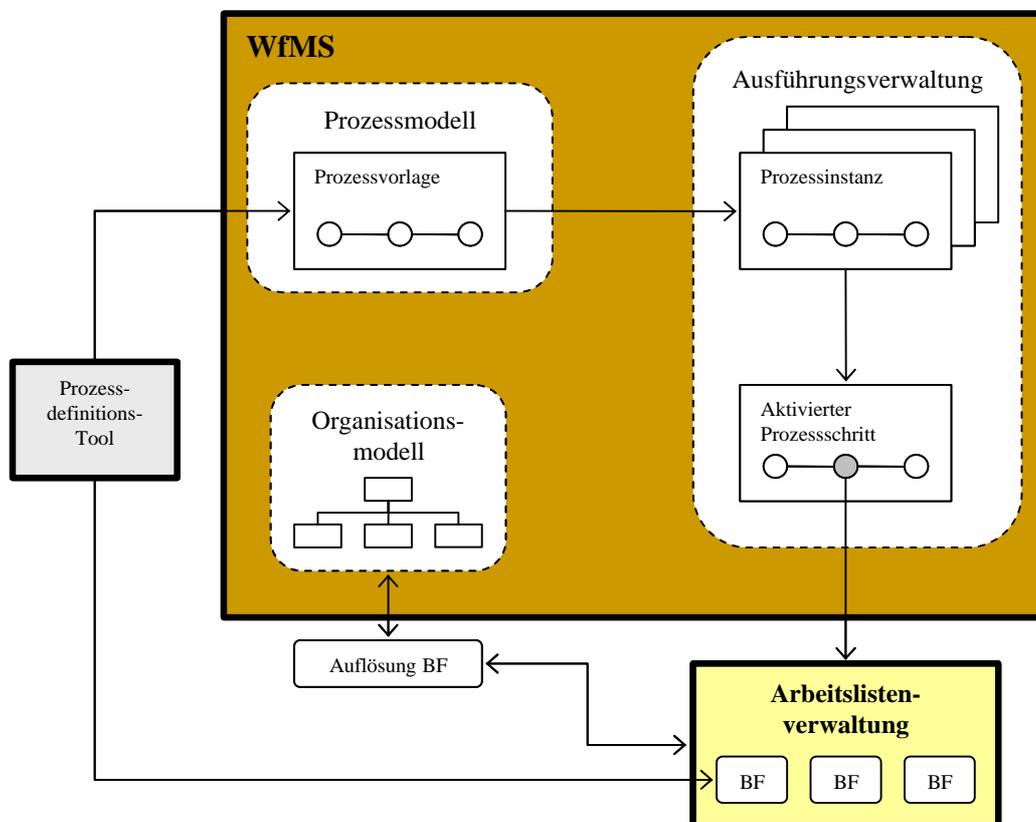


Abbildung 3-7: Hinterlegung der Bearbeiterformeln in der ALV

Damit ist es sinnvoller die Bearbeiterformeln im Prozessmodell des WfMS zu hinterlegen. Dort liegen sie an einem zentralen Platz, auf den alle Komponenten gleichberechtigt Zugriff haben, es ist ressourcenschonender für die ALV und es erspart zusätzliche Interaktion beziehungsweise Kommunikation zwischen einzelnen Komponenten.

3.3 Verwaltung der Arbeitslisten und Aktivitäten

Die Arbeitslisten und die in ihnen enthaltenen Aktivitäten haben einen großen Einfluss auf die Performanz der Arbeitslistenverwaltung. Im Normalfall hat jeder Benutzer mindestens eine Arbeitsliste. Wenn das System ausreichend groß ist, muss die ALV unter Umständen mit 1000 oder mehr Arbeitslisten umgehen können, die hinterlegt und aktuell gehalten werden müssen. Hinzukommt, dass jede Arbeitsliste zahlreiche Einträge in Form von Aktivitäten enthält, die ebenfalls in der Arbeitslistenverwaltung hinterlegt und bei Bedarf geändert werden müssen.

Folgende Aspekte sind hier in Bezug auf die Performanz relevant: die Anzahl der Arbeitslisten, die vorrätig gehalten werden, die Anzahl der Kopien einer Aktivität und die Übertragung der aktualisierten Arbeitslisten an den Klienten. Bei Arbeitslisten und Aktivitäten muss zusätzlich untersucht werden, ob und wie sie im Klienten verwaltet werden.

Verwaltung der Arbeitslisteneinträge

Jede Aktivität, die vom WfMS an die ALV übergeben wird, wird über ihre Bearbeiterformel an verschiedene Klienten verteilt und in deren Arbeitslisten gestellt. Dabei kann entweder für jede Arbeitsliste eine eigene Kopie der Aktivität erzeugt werden oder jede Arbeitsliste bekommt eine Referenz auf das gleiche Objekt zugewiesen.

Wird jeder Arbeitsliste eine eigene Kopie einer Aktivität zugewiesen, bedeutet das, dass deren Attribute unabhängig von denen anderer Kopien geändert werden können. So ist es beispielsweise möglich, individuelle Prioritäten abhängig vom Bearbeiter zu vergeben, so dass eine Aktivität nur bei jemandem priorisiert wird, der wenige Einträge in seiner Liste hat. Ein anderer Bearbeiter, dessen Arbeitsliste bereits voll ist, wird dadurch nicht mit zusätzlichen Priorisierungen weiter überlastet. Die Nachteile, die sich durch zahlreiche Kopien der gleichen Aktivität ergeben sind der gestiegene Ressourcenbedarf und der Koordinationsaufwand. Wenn eine Aktivität von 100 Bearbeitern ausgeführt werden kann, dann müssen auch 100 Kopien der Aktivität erzeugt und hinterlegt werden. Dadurch steigt der Speicherplatzbedarf maßgeblich und es müssen andere Verfahren entwickelt und angewendet werden, um ihn zu senken. Möglichkeiten dafür sind z. B. Verteilungsverfahren, Gruppenarbeitslisten oder eingeschränktere Bearbeiterformeln. Auch die Neuverteilung von Aktivitäten ist schwierig, wenn diese als Kopien für jeden Bearbeiter hinterlegt werden. Eine einzelne Kopie kann zwar einfach von einer Person auf eine andere übertragen werden, aber mehrere Kopien gleichzeitig neu zuzuordnen ist aufwändig, vor allem wenn die Größen der alten und neuen Bearbeitermenge nicht übereinstimmen. Dann ist entweder ein komplexes Verfahren zur Neuverteilung oder ein umfassendes Löschen und Neuanlegen von Aktivitätskopien erforderlich.

Wird jeder Arbeitsliste nur eine Referenz auf die Aktivität zugewiesen, verringert das den Ressourcenverbrauch und den Kommunikationsaufwand. Es wird selbst bei 100 Bearbeitern nur ein Objekt angelegt und gespeichert. Dadurch müssen keine aufwändigen Verfahren zur Reduzierung des Speicherplatzbedarfs angewendet werden. Auch die Koordination wird einfacher, da bei einer etwaigen Neuverteilung nur die Bearbeiterzuordnung innerhalb der Aktivität geändert wird und keine Objekte gelöscht oder neu angelegt werden müssen. Der Nachteil bei Referenzen ist, dass Attribute nicht individuell änderbar sind. Änderungen werden

am Aktivitätsobjekt vorgenommen und wirken sich auf alle Referenzen aus. Die individuelle Behandlung von Aktivitäten ist dabei z. B. nicht möglich.

Die Entscheidung, ob für jeden Bearbeiter Kopien oder Referenzen einer Aktivität erzeugt werden, hat großen Einfluss auf den Ressourcenbedarf der Arbeitslistenverwaltung. Sie hängt jedoch hauptsächlich davon ab, ob die Aktivitäten sich in den einzelnen Arbeitslisten unterscheiden können oder nicht. Unterscheiden sich die Aktivitätsexemplare nicht, genügt es nur ein Objekt vorrätig zu halten. Unterschiedliche Attribute sind jedoch nur in speziellen Situationen erwünscht oder sinnvoll und bilden daher eher Ausnahmefälle. Sämtliche Zustandsänderungen, Prioritätswechsel oder Bearbeiterzuordnungen gelten in der Regel global und beschränken sich nicht auf das Aktivitätsobjekt eines einzelnen Bearbeiters. Daher reicht es aus, jeder Arbeitsliste eine Referenz auf die Aktivität zu übergeben.

Verwaltung der internen Arbeitslisten

Für das Reduzieren des Speicherbedarfs der Arbeitslisten innerhalb der ALV besteht unter anderem die Möglichkeit, Arbeitslisten nicht für eine Person, sondern für eine bestimmte Gruppe zu erstellen. Dafür werden Bearbeiterformeln nicht vollständig bis zur Bearbeitermenge aufgelöst, sondern nur bis zu einer Stelle oder Rolle innerhalb des Organisationsmodells. Jedem Mitarbeiter innerhalb einer solchen Organisationseinheit wird Zugriff auf die gemeinsame Arbeitsliste gegeben. Damit wird beispielsweise innerhalb einer Stelle mit 10 Mitarbeitern statt zehn gleichen Arbeitslisten nur eine gemeinschaftliche Arbeitsliste benötigt.

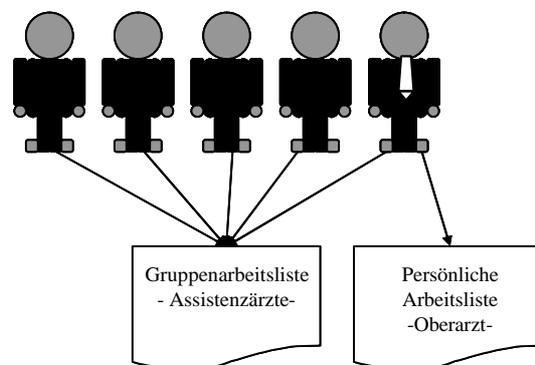


Abbildung 3-8: Persönliche und Gruppenarbeitslisten

Nachteilig ist, dass Aktivitäten, die Mitarbeitern aufgrund einer Fähigkeit oder anderen Eigenschaft zugewiesen wurde, nicht in eine Gruppenarbeitsliste gestellt werden können. Für diesen Fall müssen weiterhin persönliche Arbeitslisten gehalten werden. Zum Beispiel kann ein Assistenzarzt durch die Ausbildung zum Rettungssanitäter eine zusätzliche Qualifikation bzw. Fähigkeit erlangen, die andere Assistenzärzte nicht haben. Auch Verteilungsverfahren, welche auf einer Bearbeitermenge durchgeführt werden, können auf Gruppenarbeitslisten nicht angewendet werden.

Ein weiterer Aspekt der Vorratshaltung von Arbeitslisten beschäftigt sich mit der Unterscheidung von angemeldeten und abgemeldeten Benutzern. Angemeldete Benutzer interagieren mit dem System, bearbeiten Aktivitäten und benötigen daher eine aktuelle Arbeitsliste. Abgemeldete

Benutzer tun dies nicht. Dennoch ist es sinnvoll, zwischen einer kurzfristigen Abmeldung, wobei die Arbeitsliste intern weiter gepflegt wird und einer langfristigen Abmeldung, die ein Verwerfen der Arbeitsliste bewirkt, zu unterscheiden. Erstens ist es einem Bearbeiter so möglich, sich kurz an einer Stelle abzumelden und an einer anderen wieder anzumelden, ohne dass die Arbeitsliste an der ersten Stelle verloren geht und bei erneutem Anmelden erst wieder erstellt werden muss. So kann ein Chefarzt, der für zwei Abteilungen verantwortlich ist, schnell zwischen seinen Arbeitslisten in diesen Abteilungen wechseln. Zweitens bietet es Unterstützung, wenn sich mehrere Personen, z. B. mehrere Assistenzärzte, einen Rechner teilen müssen. Die Arbeitsliste jedes Bearbeiters wird im Hintergrund auf aktuellem Stand gehalten und kann beim Wechsel des Benutzers einfach wieder abgerufen werden. Zuletzt ermöglicht ein kurzzeitiges Hinterlegen der Arbeitslisten auch, dass sich ein Bearbeiter vorübergehend abmelden kann, wenn er seinen Arbeitsplatz verlassen muss. Anstatt sich komplett vom System abzumelden oder den Rechner zu sperren und damit für andere unzugänglich zu machen, kann er so nach seiner Rückkehr seine aktuelle Arbeit schnell fortsetzen.

Verwaltung der Arbeitslisten und Aktivitäten beim Klienten

Neben der Arbeitslistenverwaltung benötigt auch der Arbeitslistenklient Zugriff auf Arbeitslisten und Aktivitäten. Da er aber nur für die Darstellung der Arbeitslisten zuständig ist und sie selbst nicht verändern kann oder darf, sollte er nicht auf der internen Arbeitsliste arbeiten. Besser ist es, ihm eine eigene Kopie zuzuweisen, die inhaltlich mit der internen Liste übereinstimmt, aber nur lesend zugegriffen werden kann.

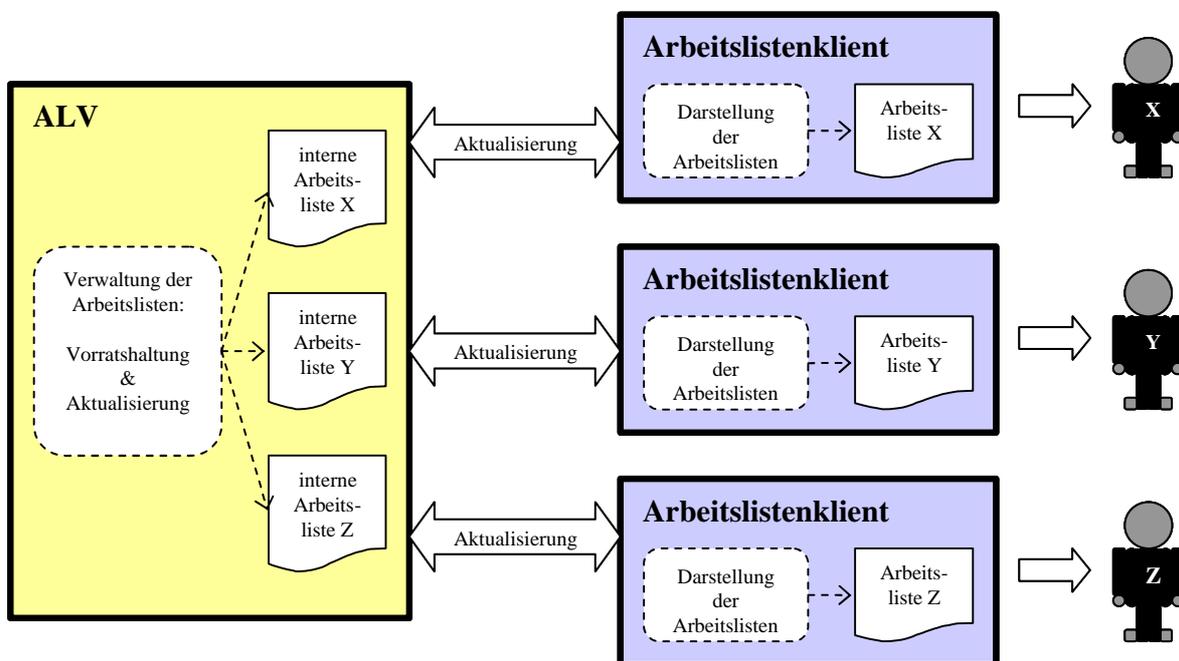


Abbildung 3-9: interne und Klient-Arbeitslisten

Der Arbeitslistenklient benötigt eigene Kopien, sowohl von der Arbeitsliste als auch von den einzelnen in ihr enthaltenen Aktivitäten. Begründen lässt sich das durch das Prinzip verteilter Systeme. Eine Referenz auf ein Objekt der Arbeitsliste bedeutet, dass für dieses Objekt ein Proxy oder Stellvertreter für den Klienten erstellt werden muss. Eine Arbeitsliste mit hundert Einträgen multipliziert mit 1.000 Bearbeitern erfordert dann schon über 100.000 Stellvertreter, die von der ALV bereitgestellt werden müssen. Daher ist es günstiger, dem Klienten eigene Kopien der Aktivitäten zu übergeben und diese dort auch zu verwalten.

Allerdings erhöht eine Klientenarbeitsliste mit hundert Einträgen den Ressourcenverbrauch beim Klienten maßgeblich und ist nicht besonders übersichtlich für den Bearbeiter. Einige Möglichkeiten, die Menge von Aktivitäten in einer Arbeitsliste zu reduzieren sind die Anwendung von Verteilungsverfahren und die Filterung von Aktivitäten. Eine Filterung von Arbeitslisteneinträgen ermöglicht es dem Klienten seine Arbeitsliste selbstständig übersichtlicher zu machen und hat zudem den Vorteil, dass weniger Ressourcen belegt werden. Verteilungsverfahren und ihre Vorteile für die Auslastung von Arbeitslisten werden in Kapitel 4 diskutiert.

Übertragung aktualisierter Arbeitslisten an den Klienten

Je nach Durchsatz der Aktivitäten und Anzahl der Bearbeiter ist eine einzelne Arbeitsliste ständigen Änderungen unterworfen und ihre Aktualisierung beim Klienten führt zu einem hohen Datenaufkommen zwischen der Arbeitslistenverwaltung und dem Arbeitslistenklienten. Daher muss eine Möglichkeit gefunden werden, eine Aktualisierung mit so wenigen Daten wie möglich durchzuführen. Dabei gibt es verschiedene Varianten: die Arbeitsliste kann jedes Mal komplett übertragen werden, oder es werden nur die Änderungen in Bezug auf die letzte Aktualisierung übermittelt. Diese Änderungen können erst kurz vor der Aktualisierung zusammengestellt oder laufend in der ALV verwaltet werden.

Die Übertragung der gesamten Arbeitsliste bei jeder Aktualisierung hat den Vorteil, dass weder bei der ALV noch beim Klienten aufwändige Berechnungen durchgeführt werden müssen, um herauszufinden, was sich seit der letzten Auslieferung geändert hat. Der große Nachteil ist aber, dass damit ein hohes Verkehrsaufkommen verursacht wird. Selbst wenn keine oder nur sehr wenige Aktualisierungen vorliegen wird die gesamte Arbeitsliste übertragen.

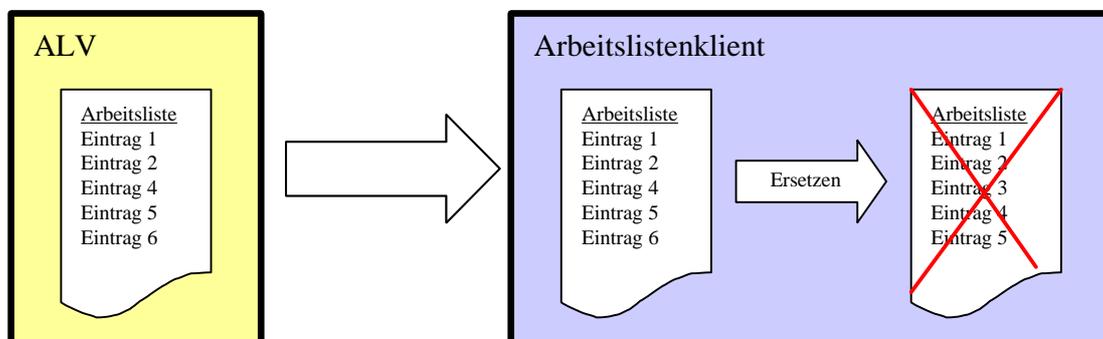


Abbildung 3-10: Übermittlung einer kompletten Arbeitsliste

Die zweite Möglichkeit ist, nur die aktualisierten Aktivitäten zu übertragen, d. h. Aktivitäten, die neu hinzugekommen sind, aus der Arbeitsliste entfernt wurden oder deren Attribute sich geändert haben. Die Informationen, welche Art von Aktualisierung vorliegt, müssen allerdings mit übertragen werden, da es für den Arbeitslistenklienten zu aufwändig wäre, das selbst mit seiner Arbeitsliste abzugleichen. Es wird also eine Liste von Tupeln bestehend aus Aktivität und Änderungsgrund übertragen. Diese Liste kann erstellt werden, wenn eine Übertragung zum Klienten ansteht oder sie wird in der ALV hinterlegt und ständig gepflegt. Beide Varianten haben ihre Vor- und Nachteile.

Wird die Liste erst bei einer anstehenden Übertragung zusammengestellt, müssen die Informationen über die Änderungen einzelner Aktivitäten in der Arbeitslistenverwaltung hinterlegt werden. Dazu müssen sowohl die Klientenarbeitsliste als auch die Aktivitäten der internen Arbeitsliste mit Zeitstempeln versehen werden, welche die letzte Aktualisierung angeben. Des Weiteren muss zu jeder Aktivität angegeben werden, welche Art von Aktualisierung vorgenommen wurde. Das Problem hierbei ist, dass diese Information nicht bei der Aktivität selbst hinterlegt werden kann, da sie sich nicht auf das Objekt selbst, sondern auf die einzelnen Einträge in den Arbeitslisten bezieht. Da jede Arbeitsliste nur eine Referenz auf die Aktivität bekommt, können die Aktivitätsattribute nicht unterschiedlich belegt werden. Die Lösung verlangt also nach einer zweiten Liste in jeder Arbeitsliste, die zu allen Arbeitslisteneinträgen die Aktualisierungsinformationen speichert. Ein weiterer Nachteil ist, dass Informationen über gelöschte Arbeitslisteneinträge weiterhin in der internen Arbeitsliste vorrätig gehalten werden müssen, und zwar solange, bis der auch der letzte Bearbeiter die entsprechende Aktualisierung abgerufen hat. Die interne Arbeitsliste enthält dadurch veraltete Einträge und die Überwachung, welcher Klient wann seine Arbeitsliste aktualisiert, ist aufwändig.

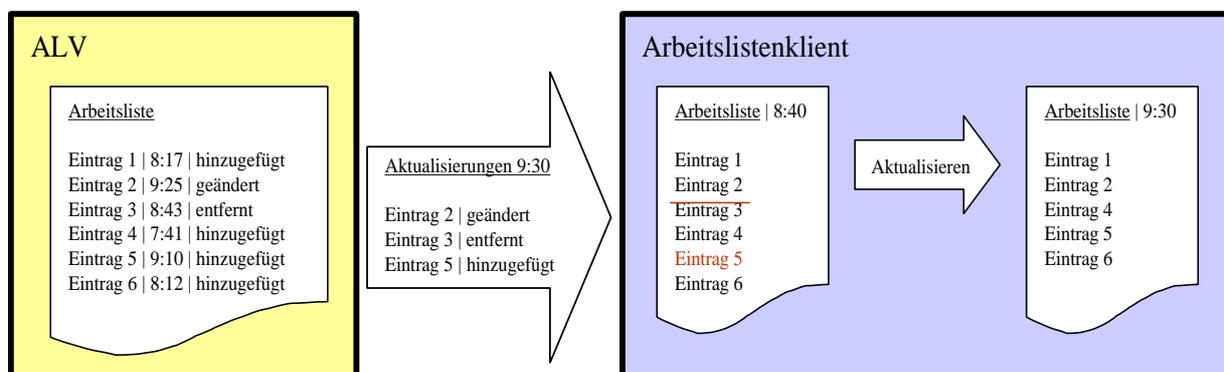


Abbildung 3-11: Erstellen und Übermitteln einer Aktualisierung

Eine von der ALV hinterlegte und verwaltete Aktualisierungsliste bedeutet, dass sämtliche Änderungen an Aktivitäten in einer separaten Liste hinterlegt und bei Bedarf an den Klienten ausgeliefert werden. Diese Liste wird ständig gepflegt und ist somit immer auf dem neuesten Stand. Nach einer Übertragung an den Klienten wird die Liste geleert, um die nächsten Neuerungen aufnehmen zu können. Das hat den Vorteil, dass keine Zeitstempel hinterlegt und verwaltet werden müssen, um die Inhalte der internen Liste und der Arbeitsliste des Klienten abzugleichen. Außerdem müssen Informationen über die Art der Aktualisierung nicht bei der Aktivität oder zusätzlich in der Arbeitsliste hinterlegt werden. Entfernte Arbeitslisteneinträge

können in der Aktualisierungsliste hinterlegt und daher aus der internen Arbeitsliste gelöscht werden. Schwierig wird es bei Gruppenarbeitslisten. Da reicht eine einzelne Aktualisierungsliste nicht aus, da mehrere Arbeiter zu unterschiedlichen Zeiten aktualisieren können. Deshalb muss in diesem Fall für jeden Bearbeiter der Gruppenarbeitsliste eine eigene Aktualisierungsliste angelegt werden. Das bedeutet für die ALV, dass sie mehr Speicherplatz benötigt, um sämtliche dieser Listen zu hinterlegen. Dafür aber ist der Verwaltungsaufwand hier geringer, als wenn die Aktualisierung erst nach Bedarf zusammengestellt wird.

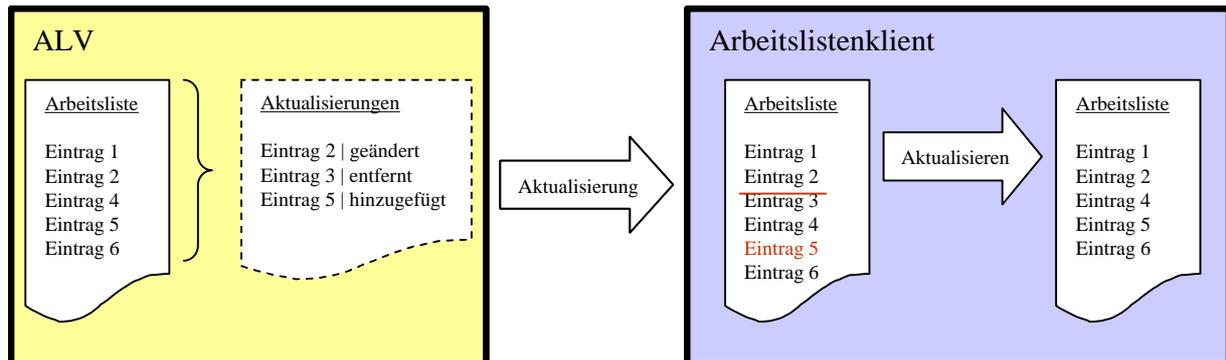


Abbildung 3-12: Verwaltung und Übermittlung einer Aktualisierung

Von diesen drei Möglichkeiten besitzt eine von der Arbeitslistenverwaltung gepflegte Aktualisierungsliste das größte Potential. Sie hat zwar einen etwas höheren Verbrauch an Speicherplatz, ist aber in ihrer Verwaltung wenig aufwändig und kann bei Bedarf ohne Verzögerung an den Klienten übertragen werden. Des Weiteren wird bei dieser Methode die geringstmögliche Menge an Daten zum Klienten übermittelt, wodurch das Verkehrsaufkommen in Grenzen gehalten wird.

3.4 Kommunikation mit dem Arbeitslistenklienten

Die Arbeitslistenverwaltung hält für jeden Klienten eine interne Arbeitsliste vorrätig. Diese interne Liste ist änderbar und wird von der ALV stets aktuell gehalten. Der Klient arbeitet auf seiner eigenen Arbeitsliste, die nur eine flache, weniger informationshaltige Kopie der internen Variante darstellt. Beispielsweise kann der Klient sich entscheiden seine Arbeitsliste zu filtern und so die Menge der Einträge zu reduzieren. Auch Informationen, die von der ALV zur Verarbeitung benötigt werden, wie z. B. die Bearbeiterformel einer Aktivität, werden nicht an den Arbeitslistenklienten übertragen. Dieser hat nur die Aufgabe die Arbeitslisten darzustellen. Weder die Arbeitsliste selbst noch die beinhalteten Aktivitäten können durch den Klienten manipuliert werden.

Die interne Arbeitsliste unterliegt der ständigen Veränderung. Die Arbeitsliste des Klienten wird jedoch nur aktualisiert, wenn die Änderungen physisch an diese Liste gesendet werden. Dieser Datenaustausch kann synchron oder asynchron erfolgen.

Eine synchrone Kommunikation (*Pull*) bedeutet, dass der Klient sich die Aktualisierung selbst bei der ALV holen muss, indem er eine Anfrage stellt. Die Antwort erfolgt sofort und der Arbeitslistenklient blockiert solange, bis er die Antwort erhalten hat oder eine Zeitgrenze überschritten wird. Wann und wie oft die Arbeitsliste aktualisiert wird hängt vom Klienten ab. Aktualisiert er selten, hält das zwar das Kommunikationsaufkommen in Grenzen, führt aber zu veralteten Arbeitslisten. Aktualisiert er dagegen ständig, hat er immer aktuelle Arbeitslisten, aber der Kommunikationsaufwand steigt, was nachteilig ist, vor allem wenn keine neuen Aktualisierungen vorliegen.

Abhilfe schafft ein zeitgesteuertes Pull. Dabei wird im Arbeitslistenklienten ein Zeitintervall gesetzt über das in geregelten Abständen Pulls an die ALV gesendet werden. Aber auch hier besteht das Problem weiterhin, dass Arbeitslisten beim Klienten nicht immer auf dem neuesten Stand sind und wichtige Informationen, wie z. B. Priorisierungen nicht zeitnah übertragen werden.

Dieses Problem behebt die asynchrone Kommunikation (*Push*). Dabei registriert sich der Klient bei der ALV und diese schickt dann bei jeder einzelnen Änderung der Arbeitsliste die Aktualisierung an den Klienten. Damit ist dessen Arbeitsliste stets aktuell und wichtige Informationen werden ihm sofort mitgeteilt. Auch findet eine Kommunikation wirklich nur dann statt, wenn Neuerungen vorliegen. Der Nachteil ist aber, dass unter Umständen ein hohes Verkehrsaufkommen zwischen Klient und ALV verursacht wird, welches abhängig von der Anzahl der Arbeitslisten und dem Durchsatz von Aktivitäten die Performanz der beteiligten Komponenten stark beeinträchtigen kann.

Um das etwas abzuschwächen gibt es auch hier wieder die Möglichkeit ein zeitbasiertes Verfahren anzuwenden. Der Vorteil dabei ist, dass im Gegensatz zum zeitgesteuerten Pull keine Nachrichten verschickt werden, wenn keine Aktualisierungen vorliegen. Dadurch erhält man eine Arbeitsliste, die weitgehend aktuell ist. Wichtige Informationen werden verhältnismäßig schnell übertragen und unnötige Kommunikation wird vermieden.

Um sicherzustellen, dass wichtige Informationen zeitnah übertragen werden, ist es sinnvoll eine prioritätsgesteuerte Aktualisierung anzuwenden. Dabei werden Aktivitäten, die auf eine höhere Prioritätsstufe gesetzt werden, mittels Push unmittelbar an den Klienten übertragen. Dadurch werden unabhängig vom eigentlichen Kommunikationsverfahren wichtige Aktivitäten sofort an die Bearbeiter ausgeliefert.

Letztendlich liegt die Entscheidung darüber, welche Kommunikation er mit der Arbeitslistenverwaltung betreiben will, im Ermessen des Klienten. Für die ALV ist es daher sinnvoll, alle Möglichkeiten zur Verfügung zu stellen.

3.5 Zusammenfassung

In diesem Kapitel wurde das Hauptarbeitsgebiet der Arbeitslistenverwaltung untersucht: die Zuordnung von Aktivitäten und Arbeitslisten auf einzelne Benutzer. Dabei wurde genauer auf die Behandlung von Bearbeiterformeln sowie auf die Verwaltung und Übertragung von Arbeitslisten eingegangen.

Die erste Untersuchung beschäftigte sich mit der Frage, wo Bearbeiterformeln am günstigsten hinterlegt und von wem sie aufgelöst werden.

Die Erkenntnis ist, dass Bearbeiterformeln ins Prozessmodell des WfMS gehören, da auf sie nicht nur von der Arbeitslistenverwaltung, sondern auch von anderen Komponenten aus zugegriffen wird. Die Auflösung der Bearbeiterformeln sollte jedoch der Arbeitslistenverwaltung überlassen werden. Eine Bearbeiterformel nimmt weitaus weniger Platz in Anspruch als eine Menge von Bearbeitern. Außerdem bietet eine Schnittstelle für die Auflösung solcher Formeln die Möglichkeit, auch andere Mitarbeitermengen (Delegationsempfänger) oder Gruppenarbeitslisten durch Bearbeiterformeln abzubilden und diese Platz sparend zu hinterlegen oder gar bei Bedarf erst zu erstellen.

Als nächstes wurde die Verwaltung der Arbeitslisten genauer untersucht. Prinzipiell kann für jeden Bearbeiter eine Arbeitsliste vorrätig gehalten werden. Allerdings ist es aus Gründen der Performanz sinnvoll, möglichst Gruppenarbeitslisten zu verwenden. Um den Speicherbedarf innerhalb der ALV gering zu halten, sollten Aktivitäten an die einzelnen internen Arbeitslisten nur als Referenz zugewiesen werden. Aus dem gleichen Grund ist es günstiger an die Klienten nicht nur Referenzen, sondern vollständige Objekte der Aktivitäten und Arbeitslisten zu übergeben. Dadurch können sie von den Arbeitslistenklienten selbstständig verwaltet werden und setzen keine durchgehende Verbindung zur ALV voraus.

Zuletzt wurden noch verschiedene Arten der Kommunikation zwischen ALV und Klient vorgestellt und diskutiert. Die Auswirkungen einer Kommunikationsart beeinflussen vor allem den Arbeitslistenklienten, daher sollte die Entscheidung von ihm getroffen werden. Für die ALV entsteht dadurch die Anforderung, Unterstützung für alle Varianten anzubieten.

4 Verteilungsverfahren

Ist die Bearbeitermenge für eine Aktivität einmal aufgelöst, stellt sich die Frage, welchen Bearbeitern aus dieser Menge die Aktivität wirklich zugeteilt werden soll.

Die einfachste Möglichkeit ist, die Aktivität an alle Bearbeiter der aufgelösten Menge zu verteilen. Das hat den Vorteil, dass es den wenigsten Aufwand für die ALV erzeugt, da sie die Aktivitäten einfach weiterleitet, ohne ein kompliziertes und zeitaufwendiges Verfahren anzuwenden. Außerdem macht es eine Neuverteilung innerhalb derselben Bearbeitermenge unnötig, da bereits alle Bearbeiter informiert sind. Nachteilig ist dabei jedoch, dass einzelne Aktivitäten unter Umständen sehr lange in den Arbeitslisten stehen, ohne bearbeitet zu werden, weil sich niemand dafür verantwortlich fühlt. Das hat zur Folge, dass ungeliebte, schwierige und langwierige Aufgaben liegen bleiben [GaEd97]. Eine schnelle Abarbeitung bedarf dann des manuellen Eingriffs von außen durch Priorisierungen oder Delegation, was wiederum Mehraufwand für die Verantwortlichen bedeutet. Des Weiteren werden die einzelnen Arbeitslisten bei entsprechendem Arbeitsangebot sehr groß und unübersichtlich. Eine überlastete Arbeitsliste kann negative psychologische Auswirkungen bei den Bearbeitern auslösen, wodurch deren Motivation und Zufriedenheit sinken und die Arbeitsleistung zurückgeht [Vand05, DeHa01]. Ein hoher Durchsatz der Aktivitäten verursacht zudem ständige Änderungen der Arbeitslisten und damit sowohl hohen als auch unnötigen Datentransfer zum Klienten.

Daher ist die Anwendung von Verteilungsverfahren sinnvoll, die eine schnelle Abarbeitung und einen hohen Durchsatz von Aktivitäten ermöglichen. Dabei wird zudem die Performanz der Datenhaltung verbessert und der Ressourcenbedarf beim Klienten verringert. Im Folgenden werden zwei Verteilungsverfahren vorgestellt und auf ihre Einsatztauglichkeit hin untersucht. Abschnitt 4.1 erörtert das Round-Robin-Verfahren und Abschnitt 4.2 die Lastabhängige Verteilung. Beide Verfahren versuchen, die Aktivitäten so zu verteilen, dass die Bearbeiter gleichmäßig ausgelastet und dabei nicht überlastet sind.

4.1 Gleichmäßige Vorabverteilung durch Round Robin

Das Round-Robin-Verfahren basiert auf einer geordneten Liste über allen Mitarbeitern und weist jede ankommende Aktivität einem bestimmten Bearbeiter zu. Die Liste wird bei jeder neu zuzuweisenden Aktivität sequentiell abgearbeitet. Die Aufgabe wird dann jeweils dem Mitarbeiter zugeteilt, der als nächstes an der Reihe ist und als Bearbeiter der Aktivität in Frage kommt. [FSP00, ZhTo93]

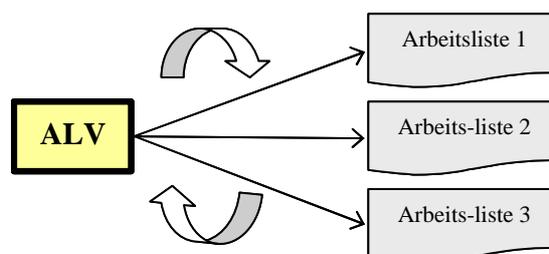


Abbildung 4-1: Round-Robin-Prinzip

Eine Voraussetzung von Round Robin ist, dass die Menge, über die verteilt wird, einheitlich ist. In einem WfMS ist es jedoch so, dass aufgelöste Bearbeitermengen von zwei verschiedenen Aktivitäten auch paarweise disjunkt sein können oder sich nur teilweise überschneiden. Die Bearbeitermenge einer Aktivität besteht im Normalfall auch nur aus einer Teilmenge der gesamten Mitarbeitermenge, über die das Round-Robin-Verfahren angewendet wird.

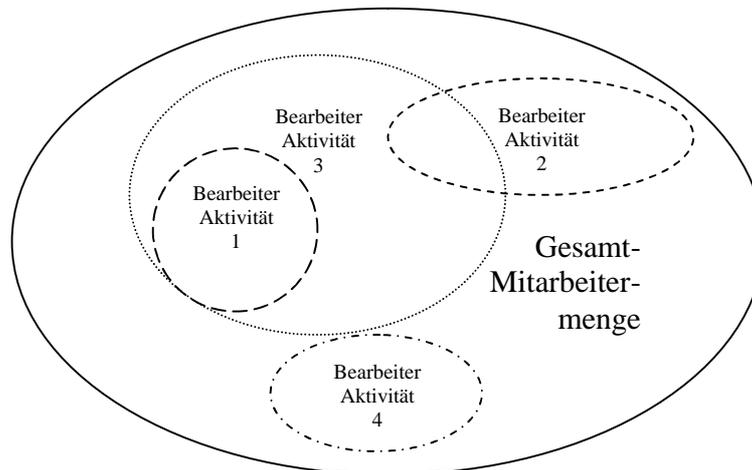


Abbildung 4-2: Zusammenhang von Bearbeitermengen

Es stellt sich also die Frage, wie man den Mitarbeiter für die nächste Zuteilung findet, ohne das Round-Robin-Prinzip zu verletzen. Das heißt, es muss der erste Mitarbeiter gefunden werden, der als nächstes an der Reihe ist *und* die Aktivität auch bearbeiten kann.

Die Lösung ist eine geordnete Menge über der Gesamt-Mitarbeitermenge. Diese Mitarbeiterliste ist aufsteigend sortiert nach der letzten Zuteilung einer Aktivität an einen Bearbeiter. Das heißt am Anfang stehen die Mitarbeiter, die als nächstes an der Reihe sind, eine Aktivität zugewiesen zu bekommen und am Ende der Liste steht der Mitarbeiter, dem zuletzt eine Aktivität zugewiesen wurde. Bekommt ein Mitarbeiter eine Aktivität zugewiesen, wird er von seiner Position in der Liste entfernt und am Ende der Liste wieder eingestellt.

Beispiel: Die als nächstes aktivierten Prozessschritte sind Aktivität c und Aktivität a. Abbildung 4-3 zeigt die gegebene Mitarbeiterliste, über die das Round-Robin-Verfahren angewendet wird.

Zuerst wird Aktivität c verteilt. Der nächste Mitarbeiter in der Liste ist Mitarbeiter 1, aber der kann Aktivität c nicht ausführen. Also wird die Liste weiter sequentiell durchgegangen, bis der erste qualifizierte Mitarbeiter gefunden wird: Mitarbeiter 3.

Mitarbeiter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Aktivitäten	b		a, c		d	a, c	a	d	a	b, c		b		c	b

Abbildung 4-3: Mitarbeiterliste vor Round-Robin-Zuweisung

Die Aktivität *c* wird Mitarbeiter 3 in die Arbeitsliste gestellt und dieser wird danach in der Round-Robin-Liste an die letzte Stelle gesetzt. Abbildung 4-4 zeigt die Mitarbeiterliste nach erfolgter Umordnung. Als nächstes wird Aktivität *a* aktiviert. Die Mitarbeiter 1,2,4, und 5 können Aktivität *a* nicht ausführen. Mitarbeiter 3 könnte, aber hat gerade eine Aktivität zugewiesen bekommen, also kommt als nächstes nur Mitarbeiter 6 in Frage.

Mitarbeiter	1	2	4	5	6	7	8	9	10	11	12	13	14	15	3
Aktivitäten	b			d	a, c	a	d	a	b, c		b		c	b	a, c

Abbildung 4-4: Mitarbeiterliste nach 1. Round-Robin-Zuweisung

Die Aktivität *a* wird Mitarbeiter 6 zugeteilt und die Mitarbeiterliste entsprechend angepasst.

Mitarbeiter	1	2	4	5	7	8	9	10	11	12	13	14	15	3	6
Aktivitäten	b			d	a	d	a	b, c		b		c	b	a, c	a, c

Abbildung 4-5: Mitarbeiterliste nach 2. Round-Robin-Zuweisung

Um den jeweils nächsten qualifizierten Bearbeiter für eine Aktivität zu finden wird aus der sortierten Mitarbeitermenge und der aufgelösten Bearbeitermenge eine sortierte Schnittmenge gebildet. Diese Schnittmenge besitzt alle Elemente der Bearbeitermenge in der Sortierung der Mitarbeitermenge, d. h. das erste Element der Schnittmenge ist der nächste Bearbeiter der gegebenen Aktivität. Die nachfolgende Abbildung verdeutlicht, wie ausgehend von der geordneten Mitarbeitermenge und einer ungeordneten Bearbeitermenge eine geordnete Schnittmenge erzeugt wird.

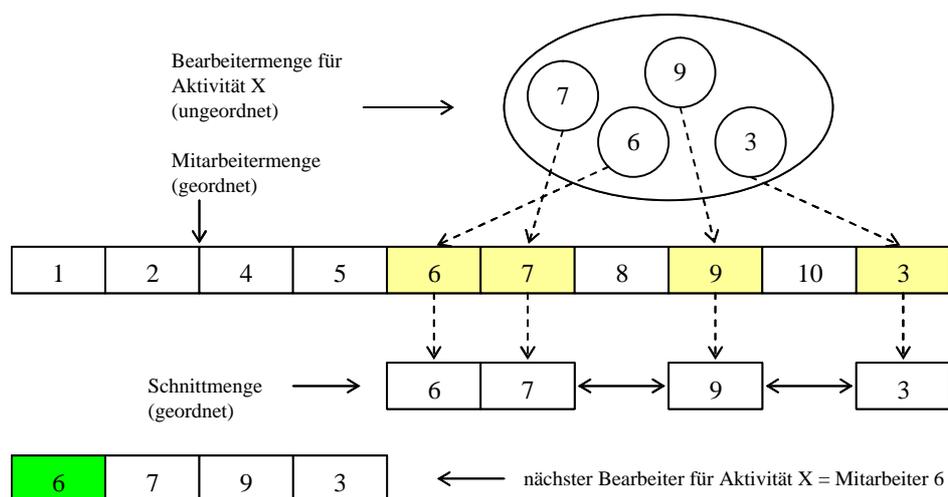


Abbildung 4-6: Bestimmung des nächsten Bearbeiters nach Round Robin

Ein vereinfachter Algorithmus für die Bestimmung der Schnittmenge sieht folgendermaßen aus:

```
MM := sortedSet;      {geordnete Mitarbeitermenge}
BM := unsortedSet;   {ungeordnete Bearbeitermenge}
SM := sortedSet;     {geordnete Mitarbeitermenge}

foreach mitarbeiter in MM do
  if mitarbeiter in BM then
    füge mitarbeiter am Ende von SM ein;
  end;
end;

bestimmter_bearbeiter := erstes Element in SM;
```

Die Vorteile des Round-Robin-Verfahrens sind offensichtlich. Jeder Bearbeiter ist allein für eine gewisse Menge von Aufgaben verantwortlich, auch schwierigere oder ungeliebte Aufgaben bleiben nicht liegen. Die Gesamtmenge der Aktivitäten wird gleichmäßig auf alle Bearbeiter verteilt. Die Aktionen der Bearbeiter beeinflussen nur ihre eigene Arbeitsliste, nicht die der anderen. Der Kommunikationsaufwand bei Aktualisierungen von Arbeitslisten und die Datenübertragungen hinzugefügter oder entfernter Aktivitäten beschränken sich auf ein Minimum.

Ein Nachteil ist, dass Aktivitäten, die immer nur einer Person zugewiesen werden, neu verteilt werden müssen, wenn sich diese Person abmeldet. Aus dem gleichen Grund dürfen Aktivitäten von vornherein nur an angemeldete Bearbeiter verteilt werden, weil sonst Aufgaben liegen bleiben, wenn der betreffende Bearbeiter gerade nicht da ist. Das bedeutet aber auch, dass ein Bearbeiter, der sich gerade neu angemeldet hat, erst einmal eine leere Arbeitsliste vorfindet, da sie nicht intern aktuell gehalten werden kann. Außerdem ist die Wahrscheinlichkeit, dass eine Aktivität schnell abgearbeitet wird geringer, wenn der Bearbeiter gerade mit einer anderen Aufgabe beschäftigt ist. Und zuletzt muss auch die geordnete Mitarbeiterliste, über die das Round-Robin-Verfahren angewendet wird, irgendwo hinterlegt und aktuell gehalten werden.

Ein weiterer Nachteil ist, dass Round Robin keine Unterschiede zwischen den zu verteilenden Aktivitäten macht. Einfache Aufgaben werden genauso an den nächsten Bearbeiter verteilt wie langwierige oder komplexe Aufgaben. Dadurch kann es passieren, dass einem Bearbeiter nur schwierige Aktivitäten zugewiesen werden und er dadurch seine Arbeitsliste langsamer abarbeiten kann, während ein andere Bearbeiter vielleicht nur einfache Aufgaben zugeteilt bekommt, diese schnell erledigen kann und dann vor einer leeren Arbeitsliste warten muss. Round Robin lohnt sich also nur, wenn sich die verschiedenen Aktivitäten innerhalb des WfMS in Komplexität und Abarbeitungsdauer ähneln.

Abgesehen davon bringt eine Verteilung nach Round Robin viele Vorteile mit sich, ist aber auch aufwendig zu realisieren. Dennoch ist es ein in der Praxis oft und gern angewendetes Verfahren zur vorsorglichen Lastverteilung und damit auch ein Kandidat für eine Anwendung innerhalb der Arbeitslistenverwaltung.

4.2 Lastabhängige Verteilung

Bei der Lastabhängigen Verteilung (*Load Balancing*) werden Aktivitäten den Bearbeitern zugewiesen, die am wenigsten ausgelastet sind. Die Auslastung der Bearbeiter wird dabei anhand ihrer Arbeitsliste berechnet und bewertet, was auf verschiedene einfache oder komplexere Arten geschehen kann. Die einzelnen Auslastungen werden dann miteinander verglichen und die Aktivität einem oder mehreren Bearbeitern mit der geringsten Auslastung zugeordnet. [WaTa97, SiKa93, AlHa97]

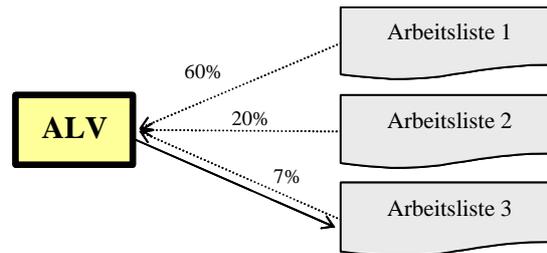


Abbildung 4-7: Auslastungsverteilung nach Auslastungsgrad

Das einfachste Verfahren beschränkt sich darauf, die Größe der Arbeitsliste zu bestimmen, indem die Anzahl der enthaltenen Aktivitäten ermittelt wird. Die neue Aktivität wird dann dem Bearbeiter zugewiesen, der die wenigsten Einträge in seiner Arbeitsliste vorweisen kann. Der Vorteil dabei ist, dass die Berechnung schnell und ohne großen Aufwand vorgenommen werden kann. Allerdings wird die unterschiedliche Komplexität der Aktivitäten nicht beachtet, was dazu führen kann, dass einige Bearbeiter mit schwierigen Aufgaben überlastet sind während andere nur einfache Arbeiten zu erledigen haben. Auch priorisierte Aktivitäten werden nicht individuell berücksichtigt, so dass es vorkommen kann, dass sie alle an dieselbe Arbeitsliste verteilt werden, wodurch der Bearbeiter überlastet und eine zeitnahe Abarbeitung verhindert wird.

$$\text{workload}(\text{bearbeiter}) = \text{size}(\text{arbeitsliste}) = S(\text{aktivität})$$

Eine genauere Bewertung der Auslastung erhält man, wenn den Aktivitäten individuelle Komplexitätswerte zugewiesen werden. Das bedeutet, einfache Aktivitäten bekommen einen niedrigen Wert, z. B. '1' und schwierigere und langwierige Aufgaben erhalten entsprechend höhere Werte. Auch priorisierte Aktivitäten können durch einen eigenen Komplexitätsindex berücksichtigt werden. Der Vorteil dabei ist, dass die Auslastung der Arbeitslisten sehr genau bestimmt werden kann und die Bearbeiter nahezu gleichmäßig ausgelastet sind. Die Komplexitätswerte müssen jedoch für jede Aktivität bestimmt, hinterlegt und verwaltet werden. Auch die Berechnung der Auslastung ist aufwändiger als bei der reinen Bestimmung der Anzahl der Aktivitäten.

$$\text{workload}(\text{bearbeiter}) = \text{size}(\text{arbeitsliste}) = S(\text{aktivität} \times \text{komplexität})$$

Noch individueller kann die Auslastung über Leistungsindexe der Mitarbeiter bestimmt werden. Dabei wird jedem Besitzer einer Arbeitsliste eine Kapazitätsgrenze bzw. obere Belastungsgrenze zugewiesen. Das kann z. B. ein Gesamtkomplexitätsindex oder eine Höchstanzahl von Aktivitäten sein, die ein Bearbeiter abarbeiten kann, ohne dabei überlastet zu sein. Dieser Leistungsindex wird verglichen mit dem momentanen Inhalt der Arbeitsliste, d. h. den Aktivitäten und ihren Komplexitätswerten, um die prozentuale Auslastung zu bestimmen. Ein Mitarbeiter mit 20 Einträgen in der Arbeitsliste und einer Kapazitätsgrenze von 50 hat damit eine höhere Auslastung als jemand, der zwar 50 Einträge hat, aber eine Belastungsgrenze von 200. Der Vorteil ist bei diesem Verfahren, dass auf die Fähigkeiten der einzelnen Mitarbeiter Rücksicht genommen werden kann. So ist es beispielsweise möglich, unerfahrenen oder neuen Mitarbeitern einen geringeren Leistungsindex zuzuweisen als denen, die schon länger die gleiche Arbeit machen und dadurch in ihrer Abarbeitung effizienter sind. Auch hier müssen die Vergleichswerte, d. h. die Leistungsindexe der Mitarbeiter festgelegt und verwaltet werden.

$$\text{workload}(\text{bearbeiter}) \text{ in } \% = \text{size}(\text{arbeitsliste}) / \text{kapazitätsgrenze}$$

Eine Erweiterung der Mitarbeiterbewertung bieten dynamische Leistungsindexe. Diese werden für die einzelnen Mitarbeiter nicht statisch festgelegt, sondern während der Laufzeit ermittelt. Dafür kann z. B. der Arbeitsdurchsatz eines Mitarbeiters herangezogen werden. Dieser wird in bestimmten Abständen, beispielsweise jede Stunde berechnet und für die Berechnung der prozentualen Auslastung verwendet. Beispielsweise hat ein Mitarbeiter mit einem Durchsatz von fünf Aktivitäten pro Stunde und zwei in der Arbeitsliste eingetragenen Aktivitäten eine Auslastung von 40%. Dynamische Leistungsindexe haben den Vorteil, dass auf die jeweilige Tagesform und den mentalen und gesundheitlichen Zustand eines Bearbeiters flexibel reagiert werden kann. Somit werden die wichtigsten Ressourcen einer Organisation – die Mitarbeiter – optimal eingesetzt und beansprucht.

$$\text{workload}(\text{bearbeiter}) \text{ in stunden} = \text{size}(\text{arbeitsliste}) / \text{durchsatz pro stunde}$$

Bei den Leistungsindexen ist es auch denkbar, nicht die Auslastung *vor* der Verteilung zu berechnen und zu vergleichen, sondern die Auslastung *nach* einer potentiellen Zuweisung zu bestimmen. Dabei wird die neue Aktivität jeder betroffenen Arbeitsliste hypothetisch zugewiesen und dann deren neue Auslastung bestimmt. Real zugeordnet wird die Aufgabe dann dem Bearbeiter, der nach der Zuweisung die geringste prozentuale Auslastung besitzt. Zum Beispiel ist es möglich, dass ein Bearbeiter, welcher nur zu 20% ausgelastet ist, nach einer potentiellen Zuweisung 40% Auslastung vorweist. Gleichzeitig könnte die Auslastung eines anderen Bearbeiters mit höherem Leistungsindex von 30% auf nur 35% steigen. In dem Fall wird die Aktivität dann dem zweiten Bearbeiter zugewiesen, obwohl er vor dem Zeitpunkt der Zuweisung höher belastet ist als der erste Bearbeiter. Der Einsatz dieses Verfahrens ist sehr aufwändig und erfordert eine Menge Berechnungen.

Die Einsatzmöglichkeiten eines auslastungsabhängigen Verteilungsverfahrens sind vielseitig. Neben der gleichmäßigen Belastung der Bearbeiter kann es unter anderem auch dazu verwendet werden, Leistungsunterschiede der Mitarbeiter auszugleichen. Außerdem funktioniert es

unabhängig von Aufgabentypen und -komplexität oder disjunkten Bearbeitermengen. Es eignet sich gut zur dynamischen Neuverteilung von Aktivitäten auf Bearbeiter und Aktivitäten werden schnell und effizient abgearbeitet. Im Gegensatz zu Round Robin ist es auch denkbar, Aktivitäten nicht nur an eine Person zu verteilen, sondern an mehrere, so dass die Chance einer schnellen Abarbeitung größer ist.

Auch hier muss man wieder darüber nachdenken, ob es sinnvoll ist, die Aktivitäten an abgemeldete Bearbeiter zu verteilen. Wenn jeweils nur ein Bearbeiter die Aktivität in der Arbeitsliste hat, dann stellt sich das gleiche Problem wie beim Round Robin: die Arbeitsliste muss nach dem Abmelden des Benutzers auf die anderen Bearbeiter aufgeteilt werden. Erstellt man die Liste erst, wenn sich ein Benutzer anmeldet, so wird sie zu Beginn erst einmal leer sein. Ein weiteres Problem ist, dass eine Auslastungsverteilung sich verzerren kann, wenn Arbeitslisten in die Berechnung mit einbezogen werden, die keinen angemeldeten Bearbeiter haben. Die einzige Bewegung kommt dann zustande, wenn andere Bearbeiter Aktivitäten reservieren und diese aus nicht beteiligten Arbeitslisten entfernt werden.

Ebenfalls nachteilig ist, dass viele Hintergrundinformationen nötig sind, um den Algorithmus anwenden zu können. Beispielsweise müssen die Effizienz der Bearbeiter, Auslastungsgrenzen und der Aufwand oder die Komplexität der einzelnen Aktivitäten hinterlegt sein. Und natürlich muss die Auslastung der einzelnen Arbeitslisten erst berechnet und dann verglichen werden, bevor eine Entscheidung über die Zuteilung getroffen werden kann.

Dennoch, auslastungsabhängige Verteilungsverfahren bergen viel Potential, nicht nur für die gleichmäßige Auslastung von Bearbeitern, sondern z. B. auch für die leistungsabhängige Belastung der Mitarbeiter. Daher lohnt sich auch ein Einsatz in der Arbeitslistenverwaltung.

4.3 Auswahl des Verteilungsverfahrens

Mit mehreren Verteilungsverfahren zur Auswahl muss ein Weg gefunden werden, sich für das Verfahren zu entscheiden, welches am geeignetsten ist.

Eine Möglichkeit ist, die Auswahl des Verfahrens manuell über Einstellungsmöglichkeiten der Arbeitslistenverwaltung festzulegen. Eine Änderung des Verfahrens zur Laufzeit ist dabei nicht möglich. Das Verfahren wird nach genauester Abschätzung der Umgebungsbedingungen (Anzahl Prozessinstanzen, Anzahl Mitarbeiter, Durchsatz bei Abarbeitung der einzelnen Aktivitäten) und abhängig vom Einsatzgebiet (Gruppenarbeit, Einzelplatzarbeit, Kommunikationsbedarf) einmal vorkonfiguriert. Das hat den Vorteil, dass individuell auf Anforderungen, drohende Probleme und spezielle Umgebungsbedingungen reagiert werden kann. Es droht keine Überlastung des Systems durch ständige Verfahrensänderungen und das angewendete Verfahren läuft stabil. Das ist z. B. sehr wichtig bei der auslastungsabhängigen Verteilung, damit eine Gleichverteilung erreicht und gehalten werden kann. Außerdem müssen so nur die für den aktuellen Algorithmus benötigten Daten vorrätig gehalten werden. Der Nachteil ist, dass eine Umstellung des Verfahrens nur möglich ist, wenn die Arbeitslistenverwaltung neu gestartet wird. Dabei müssen dann alle erforderlichen Daten bezogen oder initialisiert werden. Unter Umständen wird auch eine aufwändige Neuverteilung der Aktivitäten nötig, z. B. von Round Robin auf eine lastabhängige Verteilung umgestiegen wird.

Eine weitere Möglichkeit verlagert die Auswahl des Verfahrens dynamisch ins Ermessen der ALV. Zum Beispiel lohnen sich die meisten Verfahren erst ab einer bestimmten Anzahl von Bearbeitern oder ab einer gewissen Auslastung der Arbeitslisten. Die Arbeitslistenverwaltung überwacht das und ändert das Auswahlverfahren dynamisch ab. Der Vorteil dabei ist, dass die ALV automatisch überwacht, welches Verfahren sich zu welchem Zeitpunkt lohnt und am besten dazu geeignet ist den Durchsatz der Aktivitäten zu vergrößern. Aber hier ist das Problem, dass eine Umstellung des Verfahrens zur Laufzeit viel Aufwand und Berechnungszeit erfordert und ein hohes Datenverkehrsaufkommen durch erhebliche Änderungen von Arbeitslisten hervorgerufen wird. Ebenso müssen die erforderlichen Daten für das neue Verteilungsverfahren vorhanden sein und Grenzen zur Auswahländerung müssen überlegt gesetzt werden, so dass nicht in kurzen Zeitabständen das Verfahren durch die ALV geändert wird.

Eine dritte Möglichkeit sieht vor, ein Verteilungsverfahren als eigenständige Komponente zu realisieren. Dabei werden die Funktionalität und sämtliche benötigte und verfahrensspezifische Daten in der Komponente hinterlegt. Die Arbeitslistenverwaltung bindet dann je nach Bedarf und Situation die Komponente mit dem entsprechenden Verfahren ein. Ein Vorteil dabei ist, dass alle notwendigen Daten zentral in der Komponente des jeweiligen Verteilungsverfahrens hinterlegt werden können. Damit müssen sie beim Neustart oder Wechsel eines Verfahrens nicht erst aufwändig beschafft werden. Ein weiterer Vorteil ist, dass die ALV flexibel Anfragen an verschiedene Verteilungsverfahren stellen kann und somit nicht auf eines beschränkt ist. Dadurch können beispielsweise Aktivitäten individuell nach verschiedenen Verfahren verteilt werden. Es ist so auch möglich Verteilungsverfahren schnell zur Laufzeit zu wechseln, ohne erst einen Neustart vornehmen zu müssen, da sie nicht manuell in der ALV konfiguriert sind.

Damit bieten eigenständige Komponenten für die Verteilungsverfahren die besten Einsatzmöglichkeiten. Durch sie können Verfahren individuell und vor allem dynamisch eingesetzt werden. Alle erforderlichen Daten können in den Komponenten hinterlegt und verwaltet werden, selbst wenn ein Verfahren momentan nicht verwendet wird. Das ermöglicht einen schnellen Wechsel und problemlosen Einsatz der einzelnen Verteilungsverfahren. Der Nachteil ist auch hier, dass zumindest globale Verfahrenswechsel aufwändig sind und Neuverteilungen sämtlicher Aktivitäten nach sich ziehen können. Wenn man jedoch angemessene Grenzen setzt, in denen die ALV dynamische Wechsel vornehmen darf, kann davon abgesehen werden, Neuverteilungen vorzunehmen. Je nach Qualität des Verfahrens wird sich die Gleichmäßigkeit der Verteilung nach einer gewissen Laufzeit von selbst einpegeln.

4.4 Zusammenfassung

Im vorliegenden Kapitel wurden verschiedene Verfahren der Zuteilung von Aktivitäten auf Bearbeiter untersucht. Zwei Verfahren – Round Robin und Load Balancing – wurden vorgestellt und diskutiert. Round Robin ermöglicht eine gleichmäßige Vorabverteilung von Aktivitäten auf eine geordnete Menge von Bearbeitern während Load Balancing eine gleichmäßige Auslastung aller Bearbeiter zum Ziel hat. Beide Verfahren haben ihre Einsatzmöglichkeiten. Der Einsatz eines Verfahrens kann vorkonfiguriert oder dynamisch bestimmt werden. Am günstigsten ist jedoch die Realisierung der Verteilungsverfahren in Komponenten, die von der ALV nach Bedarf eingesetzt werden können.

5 Dynamische Neuverteilung

Unter gewissen Umständen kann eine Neuverteilung einer bereits verteilten Aktivität notwendig werden. Dabei wird die Aktivität bei den aktuellen Bearbeitern aus den Arbeitslisten genommen und anderen Bearbeitern zugeteilt. Eine Neuverteilung wird notwendig, wenn sich Umgebungsbedingungen des benutzten Verteilungsverfahrens ändern oder Bearbeiterzuordnungen zur Laufzeit geändert werden. Sie wird manuell über eine berechnete Person oder dynamisch über die Arbeitslistenverwaltung vorgenommen.

In den folgenden Abschnitten werden die Anforderungen und Auswirkungen einer dynamischen Neuverteilung näher untersucht. Dabei wird zuerst in Abschnitt 5.1 diskutiert, welche Änderungen an der Umgebung der Arbeitslistenverwaltung eine dynamische Neuverteilung auslösen können und wie darauf reagiert wird. Abschnitt 5.2 beschreibt die dynamische Änderung von Bearbeiterzuordnungen. Zwei Verfahren werden dabei genauer untersucht und miteinander verglichen: die Delegation und die Vertreterregelungen. In Abschnitt 5.3 werden die Erkenntnisse des Kapitels abschließend zusammengefasst und analysiert.

5.1 Anpassung der Verteilung an Umgebungsbedingungen

Die Verteilungsverfahren, die in Kapitel 4 vorgestellt wurden, basieren zum Teil auf Bedingungen und Voraussetzungen, die zum ordnungsgemäßen Ablauf der Verfahren erfüllt sein müssen. Beispielsweise geht *Round Robin* (Abschnitt 4.1) davon aus, dass die zur Verteilung verwendete geordnete Mitarbeiterliste nur angemeldete Mitarbeiter enthält. Außerdem wird eine Aktivität bei Round Robin nur an einen einzelnen Bearbeiter verteilt, der dann allein für die Abarbeitung verantwortlich ist und demzufolge am System angemeldet sein muss. Das Verfahren der *lastabhängigen Verteilung* (Abschnitt 4.2) versucht durch die gewählte Zuordnung der Aktivitäten eine gleichmäßige Auslastung der Mitarbeiter zu erreichen. Wenn sich die Auslastungsverteilung verschiebt, kann das eine Neuverteilung von Aktivitäten nach sich ziehen. Drei Änderungen von Umgebungsbedingungen, die relativ häufig vorkommen, werden nachfolgend untersucht.

Anmeldung eines neuen Mitarbeiters: Wenn sich ein neuer Bearbeiter am System anmeldet, muss ihm eine Arbeitsliste erstellt und übergeben werden. Das kann eine leere Liste sein, die erst gefüllt wird, wenn die nächsten Aktivitäten vom WfMS an die ALV übergeben werden. Oder die Liste des Bearbeiters wird mit bereits vorhandenen Aktivitäten gefüllt, für deren Abarbeitung er qualifiziert ist. Das können z. B. Aktivitäten sein, die sich schon länger in der Arbeitslistenverwaltung befinden und noch nicht abgearbeitet wurden. Oder man nimmt über die lastabhängige Verteilung eine Auslastungsanpassung vor und verschiebt Aktivitäten in die neue Arbeitsliste. Der genaue Ablauf hängt vor allem davon ab, wie vielen Bearbeitern eine Aktivität jeweils zugewiesen wird und welches Verteilungsverfahren angewendet wird. Eine Neuverteilung der Aktivitäten erfordert in diesem Fall jedoch einen hohen Berechnungsaufwand, da zuerst ermittelt werden muss, welche momentan in der ALV vorhandenen Aktivitäten von dem neu angemeldeten Bearbeiter erledigt werden können. Dazu müssen die Bearbeiterformeln jeder Aktivität aufgelöst und die Bearbeitermenge untersucht werden. Das führt zu hohem Kommunikationsaufwand und Datenverkehr zwischen der ALV und dem Organisationsmodell.

Je nachdem, wie hoch der Durchsatz von Aktivitäten einer Organisation ist, ist es daher sinnvoller, dem neuen Bearbeiter vorerst nur eine leere Liste zuzuweisen und diese dann auf natürlichem Wege zu füllen. Im Falle einer Round-Robin-Verteilung wird der Bearbeiter bei der Anmeldung an die erste Stelle der Mitarbeiterliste gestellt und bekommt somit die nächste für ihn gültige Aktivität zugewiesen. Auch bei einem lastabhängigen Verteilungsverfahren werden die nächsten Aktivitäten an den neuen Bearbeiter zugewiesen, da er durch seine leere Arbeitsliste automatisch die geringste Auslastung besitzt. Bei Anmeldung eines neuen Bearbeiters ist daher die Erstellung einer leeren Arbeitsliste einer dynamischen Neuverteilung vorzuziehen.

Abmeldung eines Mitarbeiters und Auflösung seiner Arbeitsliste: Wenn sich ein Bearbeiter abmeldet kann seine Arbeitsliste je nach Länge der Abwesenheit zwischengespeichert oder die Aktivitäten neu verteilt werden. Das ist vor allem notwendig bei Verteilung der Aktivitäten an nur eine Person, wie es bei Round Robin der Fall ist. In diesem Fall fällt beim Abmelden der einzige zugewiesene Bearbeiter der Aktivität weg und um die Abarbeitung weiterhin zu garantieren muss der gesamte Inhalt der Arbeitsliste neu verteilt werden. In diesem Fall werden die betroffenen Aktivitäten so behandelt, als ob sie gerade erst aktiviert wurden und zur Erstverteilung anstehen.

Auslastungsverschiebung: Das Load Balancing versucht die gleichmäßige Auslastung der Bearbeiter durch eine geeignete Verteilung der Aktivitäten so erreichen. Verschiebt sich diese Auslastung wird unter Umständen eine Anpassung der Verteilung nötig. Möglich ist dabei neben einer Verschiebung von Aktivitäten zwischen den Mitarbeitern auch die komplette Neuverteilung aller Aktivitäten. Wichtig ist, dass die Überwachung der Auslastung in definierten Zeitabständen vorgenommen wird. Dafür müssen dynamisch die aktuellen Auslastungen aller Arbeitslisten berechnet und miteinander verglichen werden. Der Abgleich zeigt, ob die Gesamtauslastung noch im Gleichgewicht ist oder eine Neuverteilung erforderlich wird. Ob eine Neuverteilung jemals nötig wird hängt vor allem davon ab, wie oft und wie viele neue Aktivitäten bei der Arbeitslistenverwaltung ankommen und wie gut das eingesetzte Load Balancing bei der Erstverteilung arbeitet.

Eine auslastungsabhängige Neuverteilung betrifft sämtliche Arbeitslisten innerhalb der ALV, sowohl bei der Auslastungsüberwachung als auch bei der anschließenden Verschiebung der Aktivitäten. Dieser Vorgang ist sehr aufwendig und kann je nach Größe und Anzahl der Arbeitslisten die Performanz innerhalb der ALV stark beeinträchtigen. Bei einer wohlüberlegten, auslastungsabhängigen Erstverteilung sind die Chancen jedoch gering sind, dass die Auslastung zwischen den Arbeitslisten sich so sehr verschiebt. Daher muss zuvor genau abgeschätzt werden, inwieweit eine dynamische Lastverschiebung und die zugehörige Überwachung sich überhaupt auszahlen.

Eine dynamische Neuverteilung nach der Änderung von Umgebungsbedingungen ist also nur dann dringend durchzuführen, wenn Bearbeiter sich abmelden, um die weitere Abarbeitung ihrer Aktivitäten zu garantieren.

5.2 Dynamische Änderung der Bearbeiterzuordnung

Änderungen an der Prozessvorlage oder am Organisationsmodell können Neuverteilungen von Aktivitäten nach sich ziehen, vorausgesetzt die Änderungen werden auf bestehende Prozessinstanzen und Prozessschritte angewendet. Solche Änderungen sind z. B. die Anpassung von Bearbeiterformeln zu einem geänderten Prozessschritt (neue Fähigkeiten, die von einem Bearbeiter verlangt werden) oder Veränderungen am Organisationsmodell (alte Bearbeiter fallen weg, ändern ihrer Eigenschaften, Stellen oder Rollen oder neue Bearbeiter kommen hinzu).

In solchen Fällen sendet das WfMS die neue Bearbeiterformel an die ALV, welche dann eine Neuverteilung vornehmen muss. In diesem Fall besteht sie darin, die Aktivität aus den Arbeitslisten der ursprünglichen Bearbeiter zu nehmen und nach dem aktuellen Verteilungsverfahren neu zuzuweisen.

Eine zweite Möglichkeit die Bearbeiterzuordnung dynamisch zu ändern ist die Anwendung von Delegationen, Eskalationen und Vertreterregelungen. In den Abschnitten 5.2.1 (Delegation) und 5.2.2 (Vertreterregelung) werden diese näher untersucht und anschließend Gemeinsamkeiten bzw. Unterschiede diskutiert (Abschnitt 5.2.3).

5.2.1 Delegation

Delegation ist die Übertragung von Arbeitsaufgaben, d. h. Aktivitäten, von einer höheren Instanz (Delegierender) an eine niedrigere Instanz (Delegationsempfänger), z. B. vom Oberarzt zu einem Assistenzarzt. Eine *Eskalation* ist eine Delegation 'nach oben', d. h. vom Oberarzt zum Chefarzt.

Delegation und Eskalation finden in der Praxis in vielen Bereichen ihre Anwendung und sollten daher auch von der Arbeitslistenverwaltung unterstützt werden. Was dabei genau zu beachten ist und welche Anforderungen die ALV dabei erfüllen muss, wird im Folgenden erörtert.

Delegation umfasst in der Regel beide Begriffe, sowohl die Delegation 'nach unten' als auch die Eskalation 'nach oben'. Daher wird in diesem Kapitel der Begriff weitgehend gemeinsam verwendet, nur wenn eine Unterscheidung notwendig ist, dann werden Delegation und Eskalation getrennt.

Um eine Delegation innerhalb der Arbeitslistenverwaltung erfolgreich umzusetzen müssen einige Dinge diskutiert werden. Die folgenden Abschnitte beschäftigen sich mit der Frage, welche Arten von Delegation es gibt, wer die Auslöser und Empfänger von Delegationen sind und welche Auswirkungen eine Delegation auf die Arbeitslisten der Beteiligten hat.

Arten einer Delegation

Es gibt drei Möglichkeiten, wie Delegation in der Praxis vorkommen kann: als reine Delegation, reine Eskalation oder als kombiniertes Verfahren, wobei zuerst eskaliert und dann delegiert wird. Die Anwendung von Delegation und Eskalation im Rahmen einer Arbeitslistenverwaltung bedeutet, dass sich Bearbeiter um Normalfälle kümmern und Ausnahmefälle an Spezialisten

eskalieren. Die Ausnahmefälle wiederum werden vom Vorgesetzten bzw. Prozessverantwortlichen an spezielle Bearbeiter delegiert.

Reine Delegation

In gewissen Fällen kann es passieren, dass für die Ausführung einer Aktivität ein bestimmter Arbeiter erwünscht ist, der diese Aufgabe jedoch nicht in seiner Arbeitsliste hat. In dem Fall muss ein Vorgesetzter bzw. der Prozessverantwortliche die Aktivität manuell dem Bearbeiter zuweisen. Dabei handelt es sich um eine reine Delegation, wie sie in Abbildung 5-1 verdeutlicht wird.

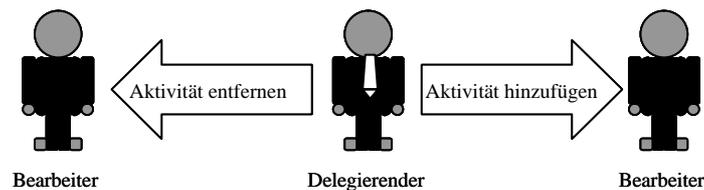


Abbildung 5-1: Delegation einer Aktivität

Weitere Einsatzmöglichkeiten ergeben sich zum Beispiel, wenn ein Oberarzt möchte, dass eine Aktivität von einem bestimmten Assistenzarzt innerhalb der Bearbeitermenge ausgeführt wird oder wenn ein Chefarzt generell sämtliche Arbeit selbst auf seine Untergebenen verteilen möchte.

Reine Eskalation

Ein Bearbeiter kann, wenn eine problematische Aktivität auftritt, diese auch direkt an einen Vorgesetzten oder die verantwortlichen Spezialisten weiterleiten. In dem Fall handelt es sich um eine reine Eskalation. Beispielsweise kann ein Assistenzarzt, der Probleme beim Erstellen einer Diagnose hat, diese Aufgabe an seinen Oberarzt eskalieren.

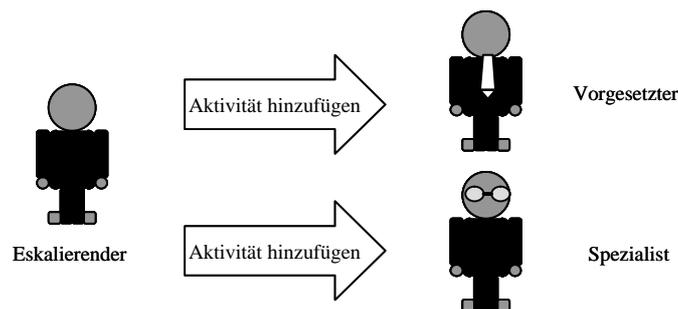


Abbildung 5-2: Eskalation einer Aktivität

Kombinierte Eskalation/Delegation

Eine Kombination von Eskalation und Delegation läuft in der Regel so ab, dass ein Arbeitsschritt erst eskaliert und dann vom Eskalationsempfänger an einen geeigneten Bearbeiter delegiert wird.

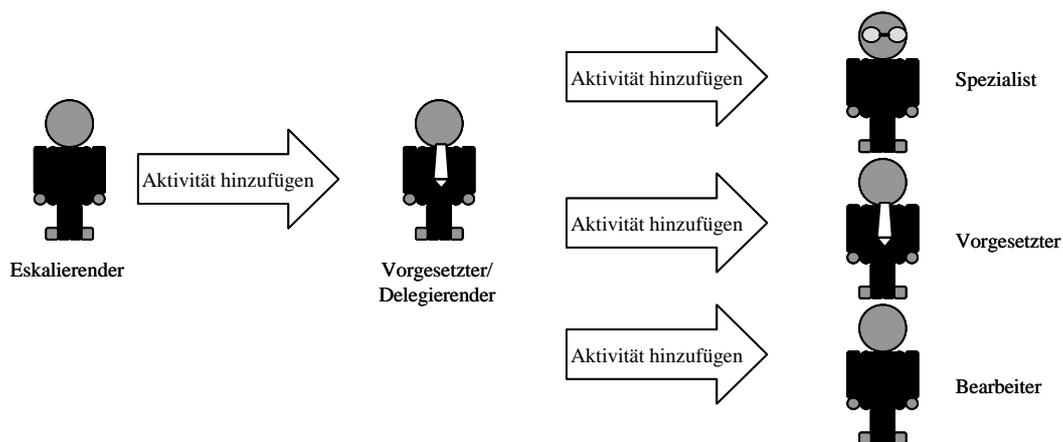


Abbildung 5-3: kombinierte Eskalation und Delegation

Diese Situation kann auftreten, wenn ein Bearbeiter mit der Abarbeitung einer Aktivität Probleme hat und seinen Vorgesetzten hinzuziehen will. Dieser ist dann dafür verantwortlich die Aktivität an jemanden weiterzuleiten, der die Fähigkeit besitzt, das Problem zu lösen.

Delegationsauslöser

Generell gibt es zwei Möglichkeiten eine Delegation auszulösen: automatisch und manuell. Beide bieten verschiedene Einsatzmöglichkeiten und verlangen nach bestimmten Anforderungen.

Eine automatische Delegation muss durch vordefinierte Ereignisse innerhalb der Arbeitslistenverwaltung ausgelöst werden. Vorhersehbare Probleme, wie z. B. eine leere, aufgelöste Bearbeitermenge oder fehlende Daten einer Aktivität können so ohne Eingriff von außen behandelt werden. Möglich ist auch bei einer Priorisierung wegen Zeitüberschreitung die betroffene Aktivität an eine höhere Stelle zu eskalieren, die dann dafür verantwortlich ist, die Arbeit manuell zu delegieren. Wichtig ist, dass die auslösenden Ereignisse vorher definiert und in der Arbeitslistenverwaltung hinterlegt werden.

Eine manuelle Delegation wird von einem Bearbeiter eigenverantwortlich angestoßen. Die Entscheidung darüber, wann und warum eine Aktivität delegiert wird, ist vom Bearbeiter selbst zu treffen und kann nicht von der Arbeitslistenverwaltung vorgegeben werden.

Delegationsempfänger

Ist eine Delegation erst einmal ausgelöst wurden, besteht der nächste Schritt darin, die Empfänger zu bestimmen. Dieser Vorgang läuft in zwei Schritten ab. Zuerst wird ausgehend vom Delegationsauslöser bzw. der zu delegierenden Aktivität eine Menge potentieller Empfänger ermittelt. Diese Menge wird dann vom Delegationsauslöser dazu verwendet den oder die endgültigen, neuen Bearbeiter zu bestimmen.

Delegation und Eskalation haben verschiedene Empfängermengen. Da Delegationen in der Organisationshierarchie von oben nach unten getätigt werden, kann jeder, der die Aufsicht über

eine Reihe von Bearbeitern hat, eine Aktivität delegieren. So kann zum Beispiel ein Oberarzt Aktivitäten an seine Assistenzärzte delegieren während die Assistenzärzte, da sie in der Hierarchie der Ärzte auf der untersten Ebene stehen, nicht delegieren können. Eine Eskalation findet innerhalb der Organisationshierarchie immer nach oben statt, d. h. alle Bearbeiter, selbst die Assistenzärzte auf der untersten Hierarchieebene, können Aktivitäten eskalieren. Bei beiden Arten ist es möglich über mehrere Hierarchieebenen hinweg nach oben oder unten zu delegieren.

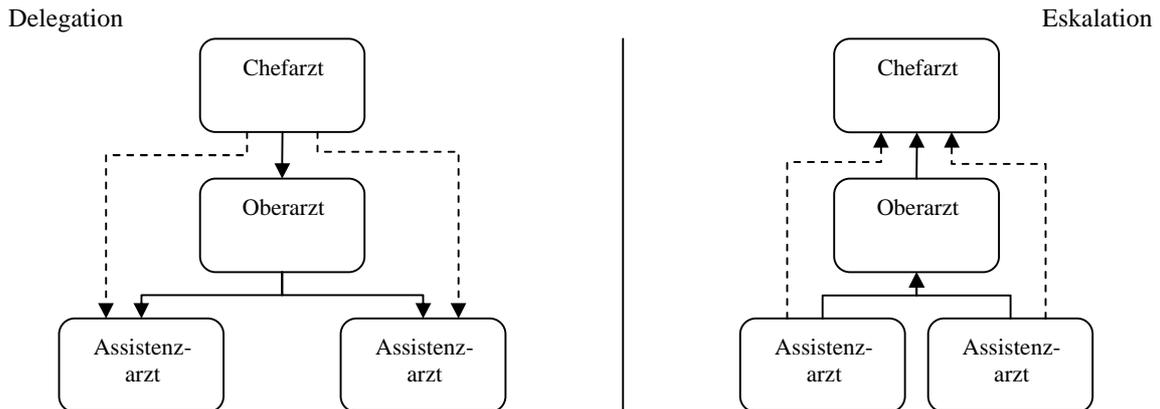


Abbildung 5-4: Delegationstiefe

Die Menge der Delegationsempfänger lässt sich leicht über Bearbeiterformeln ermitteln. Ausgehend von der Stelle und Position des Delegierenden findet man alle Unterebenen bzw. Vorgesetzten und kann diese als mögliche Empfängermenge zur weiteren Auswahl anbieten. Für einen Oberarzt ergibt sich beispielsweise als Delegationsempfänger die Menge der Assistenzärzte und als Eskalationsempfänger der Chefarzt. Solche Bearbeiterformeln können auch gut für die allgemeine Verwendung vordefiniert werden. Damit muss nur noch der Delegierende eingesetzt werden, um individuell die Menge der Empfänger zu bestimmen. Die Wahl der Delegationstiefe lässt sich dann über eine iterative Auflösung der Bearbeiterformel erreichen. Inwieweit die Bearbeiterformeln diese Anwendung zulassen hängt davon ab, wie das Organisationsmodell aufgebaut ist und welche Abfragen man daran stellen kann.

Wie ein solcher Algorithmus zur Bestimmung eines Delegationsempfängers aussehen kann wird nachfolgend dargestellt:

Bestimmung Delegationsempfänger:

```
BF: (*).vorgesetzter      {Bearbeiterformel, liefert den Vorgesetzten von *}

delegationsempfänger := delegationsaulöser.vorgesetzter;
while tiefe < limit do
    delegationsempfänger := delegationsempfänger.vorgesetzter;
    tiefe++;
end;
```

Delegationsempfänger können auch abhängig von der betroffenen Aktivität festgelegt und bereits im Prozessmodell hinterlegt werden. Beispielsweise kann die Aktivität 'Diagnose erstellen' bei Problemen grundsätzlich an den Oberarzt der zuständigen Abteilung delegiert werden. Eine weitere Möglichkeit besteht darin, alle delegierten Aktivitäten automatisch in vorher festgelegte Arbeitslisten zu verschieben, auf die nur bestimmte Personen oder Personengruppen, wie z. B. speziell ausgebildete Fachkräfte Zugriff haben.

Delegationen können durchaus wiederholt für eine bestimmte Aktivität vorkommen, d. h. eine delegierte Aktivität kann vom Delegationsempfänger weiter delegiert werden. Damit das nicht zu einem unendlichen Verschieben von Aktivitäten führt, müssen die Stufe und die Obergrenze der Delegation hinterlegt werden. Außerdem ist es sinnvoll eine delegierte Aktivität zu kennzeichnen, so dass sie nicht bei dynamischen Neuverteilungen wieder umverteilt wird.

Automatische und manuelle Delegation stellen verschiedene Anforderungen an die Menge der potentiellen Empfänger. Eine solche Menge enthält in der Regel mehrere gleichberechtigte Mitarbeiter, unter denen ausgewählt werden kann. Ein Softwaresystem, in diesem Fall die ALV, ist jedoch nicht in der Lage bewusst eine konkrete Entscheidung für einen Bearbeiter zu treffen. Daher ist es notwendig, dass die Empfängermenge bei einer automatischen Delegation minimal und eindeutig ist. Bei einer manuellen Delegation entfällt diese Beschränkung, da ein menschlicher Delegierender die Möglichkeit besitzt, sich in eigenem Ermessen für einen bestimmten Bearbeiter zu entscheiden.

Auswirkungen einer Delegation

Eine Delegation von Aktivitäten verursacht eine Neuverteilung von Arbeitsschritten, d. h. die betroffene Aktivität muss aus den Arbeitslisten der Delegationsauslöser entfernt und in die Listen der Delegationsempfänger eingefügt werden. Dabei ist es unerheblich, ob es sich um eine Delegation oder Eskalation handelt bzw. ob die Delegation manuell oder automatisch ausgelöst wurde.

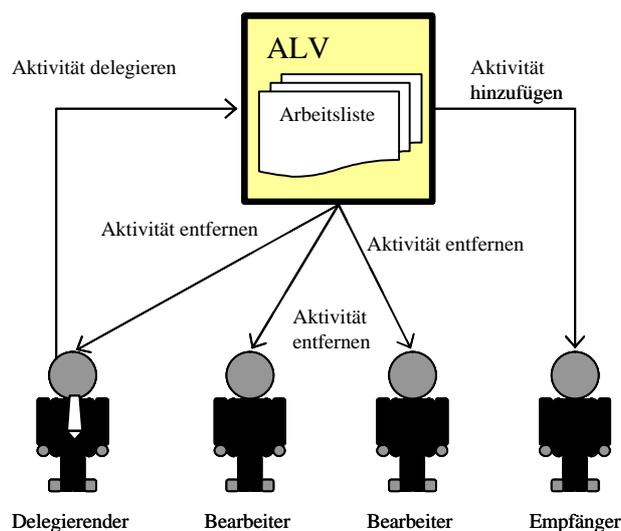


Abbildung 5-5: Änderung der Arbeitslisten nach einer Delegation

Die Auswirkungen einer Delegation beschränken sich innerhalb der ALV auf Arbeitslisten und Bearbeiterzuordnungen. Die Delegation ändert zwar die Bearbeiterzuordnung einer Aktivität (bzw. übergeht sie), aber nur kurzfristig für die Dauer ihrer Lebenszeit, d. h. das Prozessmodell selbst wird nicht verändert. Ebenso wenig wird das Organisationsmodell durch Änderungen von Bearbeiterzuordnungen beeinflusst.

Delegation bereits laufender Aktivitäten

Ein Sonderfall, der untersucht werden muss, ist die Delegation bereits laufender Aktivitäten. Im Normalfall wird eine Delegation durchgeführt, wenn die Aktivität noch in der Arbeitsliste steht und auf Bearbeitung wartet. In einigen Fällen kann es aber auch vorkommen, dass Aktivitäten delegiert werden müssen, die bereits gestartet wurden. Zum Beispiel wenn sich während der Untersuchung eines Patienten ein Problem ergibt, welches nur von einem speziell dafür ausgebildeten Facharzt behoben werden kann.

Bei der Delegation noch nicht gestarteter Aktivitäten entfernt die ALV die betroffene Aktivität aus den Arbeitslisten und übergibt sie dem Empfänger. Eine bereits laufende Aktivität kann unter Umständen nicht mehr aus der Arbeitsliste des Bearbeiters entfernt werden, da während der Abarbeitung eventuell bereits verknüpfte Anwendungen gestartet und Daten an den Klienten gesendet und von diesem geändert worden sind. Die Möglichkeiten der ALV in diesem Fall hängen davon ab, was das WfMS leisten kann. Wenn alle bereits verarbeiteten Informationen bei einer Delegation mit an den Empfänger übermittelt werden können, dann kann die Arbeitslistenverwaltung die Aktivität ohne Konsequenzen aus der Arbeitsliste des Delegierenden löschen. Ist das nicht möglich kann sie alternativ den Arbeitslisteneintrag beim Delegierenden belassen, entsprechend kennzeichnen und für weitere Aktionen sperren. Das bietet die Möglichkeit, dass der Bearbeiter die Aktivität nach der Lösung des Problems selbst fortführen und beenden kann. Die Sperrung kann außerdem dazu verwendet werden, weitere Anwendungen auf der Aktivität, wie z. B. eine Priorisierung nach Ablauf der Bearbeitungsdauer zu verhindern. Beide Varianten, d. h. das Entfernen und das Sperren der Aktivität können ohne Aufwand seitens der ALV vorgenommen werden und sollten daher auch dem Delegierenden zur Auswahl angeboten werden.

Da die Delegation einer laufenden Aktivität eine schwerwiegendere Ausnahme darstellt als normale Delegationen, muss hier auch die Möglichkeit gegeben werden eine bereits erreichte Delegationsgrenze zu überschreiten und die Delegation so trotzdem durchzuführen.

Hinterlegung der Delegationsinformationen

Die Empfänger einer Delegation können abhängig sein vom Delegierenden oder von der zu delegierenden Aktivität. Im ersten Fall ist es möglich, die Empfängermenge beim Bearbeiter zu hinterlegen, wenn sie unabhängig von der Art der Aktivität immer gleich ist, z. B. immer auf den Vorgesetzten verweist. Werden die Empfänger unabhängig von einem bestimmten Bearbeiter festgelegt, kann die Zuordnung auch in der ALV hinterlegt werden. Beispielsweise ist es möglich eine Aktivität generell an den Leiter der Stelle zu eskalieren, in dem sie bearbeitet werden soll. Über Bearbeiterformeln kann man solche Zuweisungen einfach realisieren. Empfängermengen, abhängig von einer bestimmten Aktivität, werden am besten schon im Prozessmodell definiert

und hinterlegt. Von dort aus werden sie über die einzelnen aktivierten Prozessschritte vom WfMS an die ALV übergeben und können bei den jeweiligen Aktivitäten gespeichert werden.

Die Grenzen für eine wiederholte Delegation, d. h. wie oft eine Aktivität insgesamt delegiert werden darf, können global in der ALV oder speziell für einzelne Aktivitäten im Prozessmodell hinterlegt werden. Die Zuteilung zum Prozessmodell ist jedoch nur sinnvoll, wenn sich die Grenzen für einzelne Aktivitäten unterscheiden. Die Information, wie oft eine Aktivität bereits delegiert wurde, muss direkt bei dieser hinterlegt werden.

5.2.2 Vertreterregelungen

Eine längerfristige Delegation von Aktivitäten, ohne die Bearbeiterformeln ändern zu müssen, erreicht man durch Vertreterregelungen. Die Bearbeiterformel einer Aktivität trifft keine Entscheidung dahingehend, ob ein Bearbeiter wirklich angemeldet ist und damit für die vorliegende Aufgabe zur Verfügung steht oder nicht. In solch einem Fall liefert eine Auflösung der Bearbeiterformel unter Umständen eine Bearbeitermenge zurück, die keinen gültigen Bearbeiter enthält. Damit dieser Fall abgewendet wird, wird eine zeitlich beschränkte *Vertreterregel* definiert, die zu einer gegebenen Organisationseinheit eine andere Stelle oder Rolle bestimmt, welche die Abarbeitung ihrer Aktivitäten übernimmt.

Um eine Vertreterregelung erfolgreich umzusetzen müssen einige Aspekte untersucht werden. Die folgenden Abschnitte prüfen daher, wie Vertreterregeln funktionieren, wer sie wie auslöst, wie die Empfänger bestimmt werden und welche Auswirkungen eine Vertreterregelung auf die Arbeitslisten der Beteiligten hat.

Funktionsweise von Vertreterregelungen

In einem Organisationsmodell werden mehrere Stellen und Rollen beschrieben. Eine Vertreterregel für eine Stelle bestimmt eine andere Stelle, welche die erste in genau einer Rolle vertritt. Das heißt Stelle A vertritt Stelle B in der Rolle R.

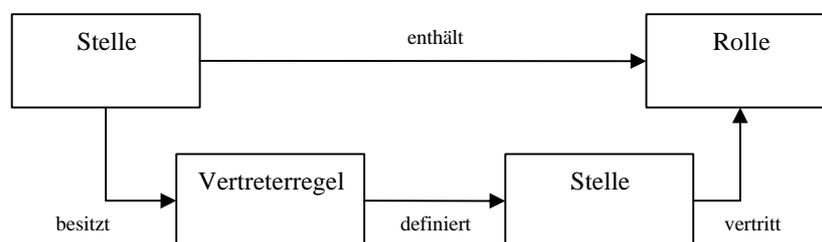


Abbildung 5-6: Vertreterregel für eine Stelle

Zwei solche Stellen sind beispielsweise die Abteilungen für 'Innere Medizin' und 'Neurologie' in einem Krankenhaus. In beiden Abteilungen gibt es mehrere Rollen, d. h. Positionen, die ein Mitarbeiter annehmen kann. Darunter befindet sich auch die Rolle der 'Krankenschwester'. Für den Fall dass eine der Abteilungen unterbesetzt ist kann man nun eine Vertreterregel definieren,

die besagt, dass die Mitarbeiter der Abteilung 'Neurologie' in der Abteilung 'Innere Medizin' aushelfen, wenn dort eine Krankenschwester benötigt wird. Das bedeutet, die 'Neurologie' vertritt die 'Innere Medizin' in der Rolle 'Krankenschwester'.

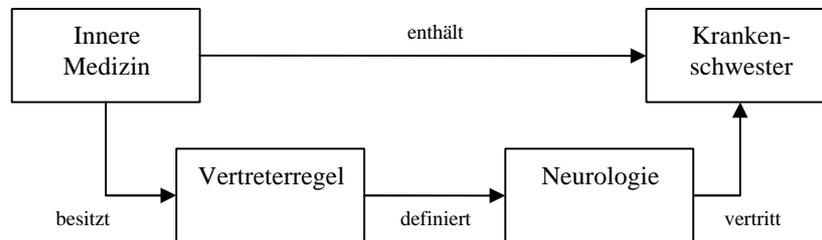


Abbildung 5-7: Beispiel einer Vertreterregel für eine Stelle

Es kann auch vorkommen, dass nur ein bestimmter Bearbeiter durch einen anderen vertreten wird. In diesem Fall verweist die Vertreterregeln nicht auf eine Stelle, sondern auf einen anderen Bearbeiter.

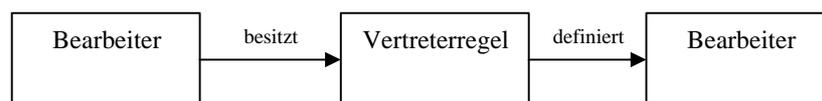


Abbildung 5-8: Vertreterregel für einen Bearbeiter

Auslöser einer Vertreterregel

Vertreterregeln sind für eine kurzfristige Anwendung gedacht, z. B. bei Krankheit oder Urlaub eines Mitarbeiters oder bei Personalmangel innerhalb einer Stelle. In solch einem Fall muss eine passende Vertreterregel definiert oder, wenn sie schon vorhanden ist, aktiviert werden. Das Anlegen bzw. Aktivieren verlangt ein manuelles Eingreifen einer berechtigten Person.

Generell kann eine Vertreterregel mit oder ohne festgelegte Zeitangabe erstellt und hinterlegt werden. Es stellt sich also die Frage, woher das System weiß, dass eine Vertreterregel aktiv ist und angewendet werden muss.

Die erste Möglichkeit besteht darin, generell bei jeder Stelle oder jedem Bearbeiter zu schauen, ob eine Vertreterregel existiert, d. h. es gibt irgendwo eine Liste mit allen Stellen und Bearbeitern, die vertreten werden und die aktuell aufgelöste Stelle oder der Bearbeiter wird damit verglichen. Diese Liste wird am günstigsten im Organisationsmodell hinterlegt, da dort die Vertretungen aufgelöst werden. Dadurch wird eine aktive Vertreterregel auf jeden Fall erkannt. Allerdings ist dieses Verfahren sehr aufwendig, da Vertretungen nur in einem kleinen Teil der Fälle aktiv sind und die Liste ständig aktuell und vorrätig gehalten werden muss.

Die zweite Möglichkeit weist jeder Stelle und jedem Bearbeiter eine Eigenschaft zu, die sie als 'vertreten' bzw. 'nicht vertreten' ausweist. Vertreterregeln werden dann nur aufgelöst, wenn die Eigenschaft 'vertreten' zutrifft. Das hat den Vorteil, dass keine zusätzliche Liste der vertretenen Stellen und Bearbeiter vorrätig gehalten werden muss und boolesche Abfragen schneller und einfacher sind als ein Vergleich mit Listen.

Eine dritte Möglichkeit ist generell die Vertretung jeder Stelle und jedes Bearbeiters aufzulösen. Eine aktive Vertreterregel verweist dabei auf die Vertretung, eine nicht-aktive Vertreterregel verweist auf die Ausgangsstelle oder den Bearbeiter zurück. Dabei muss dann zwar keine bedingte Abfrage gestellt werden, aber es muss für jede Stelle eine Vertreterregel geben, egal wohin sie verweist, was sich negativ auf den Speicherplatz auswirkt.

Da diese Überlegungen aber nicht im Einzugsbereich der Arbeitslistenverwaltung liegen, wird auf eine Entscheidung bezüglich des genauen Verfahrens verzichtet.

Auflösung von Vertreterregeln

Die Auflösung von Vertreterregeln ist eng verknüpft mit der Auflösung von Bearbeiterformeln. In der Regel werden Bearbeiterformeln bis zur verantwortlichen Stelle oder der zuständigen Rolle innerhalb dieser Stelle aufgelöst und dann die zugehörigen Mitarbeiter ermittelt. Vor der Auflösung der Mitarbeiter kann die Vertreterregel für die Stelle greifen.

Vertreterregeln für Bearbeiter verlangen, dass die Bearbeiterformeln vollständig aufgelöst werden. Die Überprüfung auf existierende Vertreterregeln wird dann auf die Bearbeitermenge angewendet.

Auswirkung einer Vertreterregel

Der Einsatz von Vertreterregeln findet im Zuge der Auflösung von Bearbeiterformeln statt. Die Arbeitslistenverwaltung veranlasst diese Auflösung zwar, wird aber von der Anwendung der Vertreterregeln nichts mitbekommen, da der eigentliche Vorgang des Auflösens für sie versteckt abläuft und nur das Ergebnis zurückgeliefert wird. Allerdings kann es für die ALV von Bedeutung sein, zu erfahren, wann Aktivitäten nicht an die Originalbearbeiter sondern an Vertretungen zugewiesen werden. Dann ist es beispielsweise möglich, die Aktivität bei der Vertretung in der Arbeitsliste entsprechend zu markieren. Da die ALV die Vertreterregeln selbst nicht hinterlegt, müssen die Informationen darüber mit der aufgelösten Bearbeitermenge vom Organisationsmodell übergeben werden. Dafür gibt es verschiedene Ansätze.

Die erste Möglichkeit besteht darin, über die Stellenbeschreibung der Aktivität herauszufinden, ob sie an eine Vertretung ausgeliefert wird oder nicht. Dabei muss zu jeder Aktivität zusätzlich zur Bearbeiterformel hinterlegt sein, von welcher Stelle sie ausgeführt werden soll. Diese Stelle wird dann verglichen mit der aufgelösten Stelle und sollten beide unterschiedliche sein, existiert eine Vertretung. Dieser Vergleich kann schnell und ohne größeren Aufwand vorgenommen werden. Allerdings muss die Stelle extra noch einmal bei der Aktivität hinterlegt werden, wodurch sich Redundanzen zu den Bearbeiterformeln ergeben.

Die zweite Möglichkeit ist, jeder Stelle und jedem Bearbeiter in der aufgelösten Bearbeitermenge zusätzlich die Eigenschaft 'vertreten' oder 'nicht vertreten' zuzuweisen. Bei der Zuweisung an die Arbeitslisten werden die Eigenschaften dann überprüft und betroffene Aktivitäten entsprechend markiert. Auch diese Überprüfungen sind ohne Aufwand durchführbar. Zusätzliche Eigenschaften der Bearbeitermenge erzeugen aber einen höheren Datenverkehr, da die Attribute bei jedem aufgelösten Bearbeiter vorhanden sind, selbst wenn er keine Vertretung übernimmt.

Eine dritte Möglichkeit sieht vor, für jede aufgelöste Bearbeitermenge eine zusätzliche Liste mit zu übergeben, die die Vertretungen innerhalb der Bearbeitermenge kennzeichnet. Diese Liste beschränkt sich nur auf die Bearbeiter, die eine Vertretung übernehmen und reduziert so den potentiellen Datenverkehr. Allerdings erfordert der Abgleich dieser Liste mit der aufgelösten Bearbeitermenge einen höheren Aufwand bei der ALV.

Die günstigste Variante ist demzufolge, alle Bearbeiter der aufgelösten Menge mit einer zusätzlichen Eigenschaft zu versehen. Vertretungen werden somit leicht erkannt und können von der Arbeitslistenverwaltung verarbeitet werden.

Auf das Prozessmodell haben Vertreterregeln keinen Einfluss. Die Bearbeiterzuordnungen der einzelnen Aktivitäten bleiben weiterhin gültig und werden nur kurzfristig durch die Vertretungen umgangen.

Hinterlegung der Vertreterregeln

Vertreterregeln sind nicht wie Bearbeiterformeln ein Teil des Prozessmodells, da sie nicht mit einem Prozessschritt sondern mit einer Stelle bzw. einem Bearbeiter verknüpft sind. Daher können sie auch nicht im Prozessmodell hinterlegt werden.

Auch eine Hinterlegung in der Arbeitslistenverwaltung ist nicht optimal, denn dadurch müssten bei der Auflösung von Bearbeiterformeln jedes Mal die betroffenen Vertreterregeln mitgeschickt werden. Die ALV weiß jedoch im Vorfeld nicht, von welcher Stelle oder von welchem Bearbeiter die Aktivität ausgeführt werden kann und kann die Vertreterregeln nicht entsprechend auswählen. Eine Alternative ist, dass das Organisationsmodell beim Auslösen einer Vertreterregel bei der ALV anfragt und sich die betroffene Regel übergeben lässt. Das hat jedoch den Nachteil, dass zusätzliche Schnittstellen und ein zusätzlicher Kommunikationsaufwand notwendig werden. Außerdem werden Vertreterregeln nicht zwangsläufig nur über die ALV angelegt bzw. zugegriffen, wodurch der Bedarf von weiteren Schnittstellen steigt.

Vertreterregeln sind eine Frage der Organisation und werden daher auch am besten im Organisationsmodell hinterlegt und verwaltet. Dort können sie zentral zugegriffen und ohne weiteren Aufwand aufgelöst werden.

5.2.3 Gegenüberstellung Delegation und Vertreterregelung

Sowohl Delegation als auch Vertreterregeln sind Verfahren, um die vordefinierte Bearbeiterzuordnung von Aktivitäten zumindest kurzfristig zu umgehen und damit dynamisch zu ändern.

Delegation bzw. Eskalation wirken sich einmalig auf die Bearbeiterzuordnung der betroffenen, delegierten Aktivität aus. Eine Vertreterregel bezieht sich nicht auf eine einzelne Aktivität, sondern auf eine Einheit des Organisationsmodells. Sie bleibt aktiv bis sie deaktiviert wird oder der vorgegebene Zeitraum endet.

Die Aktivierung einer Vertreterregel hat keine direkten Auswirkungen auf die Zusammenstellung einzelner Arbeitslisten, sondern wird außerhalb der ALV, während der Auflösung von Bearbeiterformeln angewendet. Sichtbare Auswirkungen beschränken sich auf die Darstellung vertretender Aktivitäten beim Klienten.

Die Delegation einer Aktivität hat Auswirkungen auf die Arbeitslisten aller betroffenen Bearbeiter. Darunter befinden sich der Delegierende, der oder die Delegationsempfänger und alle Bearbeiter, denen die Aktivität zuvor zugewiesen worden war. Die Arbeitslistenverwaltung muss daher nach der Delegation eine Neuverteilung der betroffenen Aktivität vornehmen. Des Weiteren sind die Delegationsempfänger abhängig vom Delegierenden bzw. der delegierten Aktivität und müssen von der ALV bestimmt und zur weiteren Einschränkung zugänglich gemacht werden.

Zusammenfassend kann gesagt werden, dass sowohl Delegation als auch Vertreterregeln ihre Anwendungsgebiete und Einsatzmöglichkeiten haben. Vertreterregeln definieren für einen kurzfristigen Zeitraum eine festgelegte Vertretung für eine Stelle oder einen Bearbeiter. Sie werden unabhängig von einer bestimmten Aktivität oder eines bestimmten Ereignisses eingesetzt. Delegation eignet sich für flexible oder selten notwendige Änderungen an Bearbeiterzuordnungen. Sie bieten sich vor allem dann an, wenn einzelne Aktivitäten aufgrund besonderer Umstände neu verteilt werden sollen. Weder Delegation noch Vertreterregeln nehmen langfristige Änderungen am Prozessmodell oder am Organisationsmodell vor und sind damit eine schnelle und flexible Alternative zur Erstellung neuer oder Änderung bereits vorhandener Bearbeiterformeln.

5.3 Zusammenfassung

In diesem Kapitel wurden die Gründe und die Verfahren der dynamischen Neuverteilung von Aktivitäten untersucht. Dabei wurden zwei grundlegende Auslöser betrachtet: die Anpassung der Verteilung an sich ändernde Umgebungsbedingungen und die Anpassung an geänderte Bearbeiterzuordnungen.

Die Anpassung einer vorliegenden Verteilung an Umgebungsbedingungen kann notwendig werden, wenn sich Voraussetzungen ändern, auf denen das angewendete Verteilungsverfahren basiert. Darunter fallen z. B. neu an- oder abgemeldete Mitarbeiter oder die Verschiebung der Auslastung. Von Änderungen des Mitarbeiterstatus werden nur die Arbeitslisten der Benutzer betroffen. Eine Verschiebung der Auslastung kann eine Neuverteilung sämtlicher Arbeitslisten der ALV nach sich ziehen. Wenn es nicht zwingend notwendig ist, sollte daher eine dynamische Neuverteilung nur vorgenommen werden, wenn sich der Aufwand für die Arbeitslistenverwaltung in Grenzen hält.

Eine Änderung von Bearbeiterzuordnungen erfordert in der Regel auch eine Neuverteilung der betroffenen Aktivitäten. Bei Änderungen der Bearbeiterformeln selbst hängt eine Neuverteilung davon ab, ob die Änderungen auf laufende Prozessinstanzen angewendet werden. Als Alternative zur langfristigen Änderung von Bearbeiterformeln bieten sich Delegation und Vertreterregeln an. Dabei werden Bearbeiterzuordnungen nur kurzfristig geändert, ohne das zugrunde liegende Prozess- bzw. Organisationsmodell zu beeinflussen. Eine entsprechende Untersuchung und ein Vergleich dieser beiden Verfahren wurden vorgenommen. Sowohl Delegation als auch Vertreterregeln haben ihre potentiellen Einsatzgebiete und sollten daher von der Arbeitslistenverwaltung unterstützt werden.

6 Priorisierung

Priorisierung bedeutet innerhalb der Arbeitslistenverwaltung, dass einzelne Aktivitäten in ihrer Bearbeitung Vorrang haben vor anderen, nicht priorisierten Aktivitäten. Gründe dafür sind häufig zeitliche Beschränkungen oder Überschreitungen, die eine schnellstmögliche Abarbeitung der betroffenen Aktivität verlangen. Die Priorisierung muss für den Klienten deutlich sichtbar in seiner Arbeitsliste kenntlich gemacht werden. Das verlangt, dass priorisierte Aktivitäten schnellstmöglich, z. B. über das Push-Verfahren an den Arbeitslistenklienten übertragen und dort entsprechend verarbeitet werden. Da die Darstellung der Arbeitsliste vom Klienten vorgenommen wird, hat die ALV nur die Aufgabe, Prioritäten bekannt zu geben.

In den nachfolgenden Abschnitten wird überprüft, welche Anforderungen eine Priorisierung von Aktivitäten an die Arbeitslistenverwaltung stellt. Abschnitt 6.1 stellt die verschiedenen Möglichkeiten der Priorisierung dar. In Abschnitt 6.2 werden Zeitgeber für die automatische Priorisierung definiert. Eine Untersuchung der Auswirkungen der Priorisierung auf Arbeitslisten wird in Abschnitt 6.3 vorgenommen und Abschnitt 6.4 befasst sich mit der Datenhaltung der verschiedenen Priorisierungsinformationen.

6.1 Abläufe einer Priorisierung

Arbeitslisteneinträge haben standardmäßig keine Priorisierung, können aber unter bestimmten Umständen und nach speziellen Regeln priorisiert werden, um eine schnellere Abarbeitung zu gewährleisten.

Eine Möglichkeit ist, die Priorisierung von Aktivitäten automatisch über die Arbeitslistenverwaltung auszulösen. Dabei werden vorgegebene Zeitspannen über Zeitgeber überwacht und bei Überschreiten eine Priorisierung ausgelöst. Solche Zeitspannen können ein nahender Endtermin sein, d. h. ein Termin an dem eine Aktivität beendet sein muss, die Verweildauer einer Aktivität in einer Arbeitsliste, ohne bearbeitet zu werden oder auch ihre maximale Bearbeitungsdauer. Sinnvoll ist eine solche Zeitspanne beispielsweise wenn die Untersuchung eines Patienten zu einem bestimmten Zeitpunkt beendet sein muss, da bereits ein OP-Termin vereinbart wurde, welcher eingehalten werden muss. Da die Arbeitslistenverwaltung immer nur für aktive Prozessschritte verantwortlich ist, kann sie auch nur deren Zeitspannen überwachen.

Eine zweite Möglichkeit besteht darin, Priorisierungen manuell von außen vorzunehmen, d. h. ein Verantwortlicher priorisiert eine Aktivität eigenständig, um ihre Abarbeitung zu beschleunigen. Dieser Fall tritt zum Beispiel ein, wenn ein Arzt dringend die Ergebnisse einer Untersuchung benötigt, um eine Diagnose stellen zu können. Die Priorisierung gilt auch in diesem Fall jeweils nur für die aktuelle Aktivität.

Eine Priorisierung kann unter Umständen auch schon bestehen, bevor der Prozessschritt aktiviert und auf die Bearbeiter verteilt wird. Diese Situation tritt beispielsweise ein, wenn der gesamte Prozess wegen eines Spezialauftrages, Eilauftrages oder Testlaufs priorisiert ist. In dem Fall muss

die Priorisierung zusammen mit dem aktivierten Prozessschritt vom WfMS übergeben und von der Arbeitslistenverwaltung entsprechend behandelt werden.

6.2 Automatische Priorisierung

Während manuelle Priorisierungen im Ermessen der verantwortlichen Person liegen, werden automatische Priorisierungen von der Arbeitslistenverwaltung übernommen. Dafür müssen gewisse Regeln existieren, die genau bestimmen, wann eine Priorisierung vorzunehmen ist.

Automatische Priorisierungen sind allesamt zeitbezogen, d. h. für jede Aktivität existiert eine vorgegebene Zeitspanne innerhalb derer sie mit normaler Priorität existiert. Wird die Zeitspanne überschritten erfolgt eine Priorisierung. Die Überwachung der Zeitvorgabe wird durch Zeitgeber bzw. Timer übernommen, die jedoch genau für diesen Zweck von der Arbeitslistenverwaltung initialisiert werden müssen. Im Folgenden findet sich eine Aufstellung verschiedener Zeitvorgaben, die im Rahmen einer Arbeitslistenverwaltung Verwendung finden.

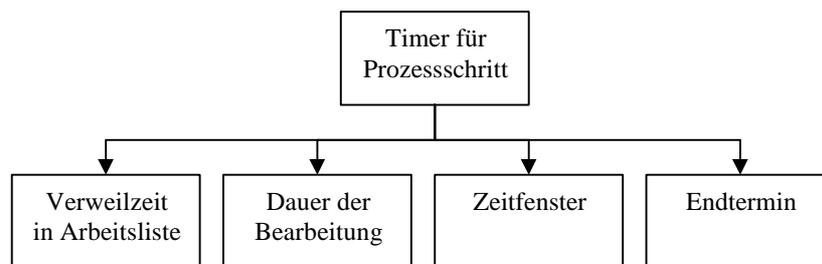


Abbildung 6-1: Zeitvorgaben für die Priorisierung in der ALV

Verweilzeit in Arbeitsliste: Eine Aktivität steht seit gewisser Zeit in den Arbeitslisten der Bearbeiter, ohne von ihnen bearbeitet zu werden. Je nach Auslastung kann die vorgegebene Zeit dynamisch an die Gegebenheiten angepasst werden. Nach Ablauf der Zeitspanne wird die Aktivität priorisiert.

Dauer der Bearbeitung: Eine Aktivität wird bereits bearbeitet, aber die Bearbeitung dauert länger als vorgegeben. Die zeitliche Vorgabe kann dabei aus den durchschnittlichen Bearbeitungsdauern der Aktivität ermittelt oder generell vorgegeben werden. Die Aktivität wird priorisiert, wenn die maximale Bearbeitungsdauer überschritten wird.

Zeitfenster: Eine Aktivität ist eng mit dem vorhergehenden Prozessschritt verknüpft und muss innerhalb einer vorgegebenen Zeitspanne nach Beendigung der vorherigen Aktivität gestartet werden. Diese Zeitspanne resultiert dabei aus Vorgaben, die durch die enge Verbindung zweier Arbeitsschritte entstehen. So kann z. B. gesagt werden, dass nach einer Blutabnahme im Krankenhaus höchstens zwei Stunden vergehen dürfen, bis das Blut untersucht wird. Solche Zeitbeschränkungen sind eng mit dem Prozessschritt verknüpft und werden günstigerweise im Prozessmodell hinterlegt. Die Priorisierung wird in diesem Fall erfolgen, bevor das Zeitfenster unterschritten wird, damit noch eine fehlerfreie Abarbeitung der Aktivität möglich ist, d. h. die vorgegebene Zeitspanne muss unter der des Zeitfensters liegen.

Endtermin: Bei Start des Prozesses wird oft eine Frist gesetzt, wann der Prozess beendet sein muss. Da die Arbeitslistenverwaltung keine Prozesse behandelt, sondern aktivierte Prozessschritte, muss diese Frist auf den Prozessschritt umgerechnet und bei Aktivierung übertragen werden. Weil es beim Überschreiten einer Frist meist schon zu spät ist, um noch angemessen zu reagieren, ist es günstiger die vorgegebene Zeitspanne kleiner zu wählen und eine Aktivität zu priorisieren, bevor die Frist abläuft.

All diese Zeitvorgaben haben eines gemeinsam: sie sind verknüpft mit einer bestimmten Aktivität und enthalten eine vorgegebene Zeitspanne, nach deren Ablauf die Aktivität priorisiert wird. Das bedeutet für die Arbeitslistenverwaltung, dass sie nur ein Verfahren anbieten muss, welches einen Zeitgeber für die angegebene Zeitspanne setzt und mit der jeweiligen Aktivität verknüpft.

6.3 Auswirkung der Priorisierung

Die Priorisierung einer Aktivität kann und sollte Auswirkungen auf die Arbeitslisten der betreffenden Bearbeiter haben. Denn das Ziel einer Priorisierung ist es, den Bearbeiter über dringende Aufgaben in Kenntnis zu setzen und so eine schnellere Abarbeitung der Aktivität zu erreichen.

Ist die Aktivität schon vor der ersten Bearbeiterzuordnung priorisiert, d. h. die Priorität besteht schon, wenn der aktivierte Prozessschritt an die ALV übergeben wird, dann wird die Priorisierung bei der Auslieferung an den Bearbeiter sofort kenntlich gemacht. Diese Situation tritt zum Beispiel bei Endterminen und Zeitfenstern auf.

Wird die Aktivität erst nach der Bearbeiterzuordnung priorisiert, d. h. die Aktivität ist schon in den Arbeitslisten vorhanden, dann muss die neue Priorität des Eintrags allen betroffenen Bearbeitern vermittelt werden. Eine entsprechende Kennzeichnung der Aktivität wird vorgenommen und über die Aktualisierung der Arbeitslisten mitgeteilt.

Eine Aktivität, die während der Ausführung priorisiert wird, d. h. die Aktivität befindet sich bereits bei einem Benutzer in Bearbeitung, steht nur noch bei dem verantwortlichen Bearbeiter in der Arbeitsliste. Diese muss aktualisiert werden, um den Bearbeiter auf die abgelaufene Zeit aufmerksam zu machen.

Kennzeichnung der Priorisierung in der ALV

Intern können die einzelnen Prioritäten über Attribute der Aktivitäten zugegriffen werden. Damit sie einheitlich verarbeitet werden können, sollten Prioritätskategorien festgelegt werden. Wie viele Kategorien es gibt bleibt den Entwicklern überlassen, sollte jedoch vorher auf Einsatztauglichkeit überprüft werden. Die Anzahl der Kategorien bestimmt sowohl die Menge der Informationen, die dargestellt werden können als auch die Übersichtlichkeit der Arbeitsliste beim Klienten.

Die einfachste Möglichkeit ist die Gruppierung der einzelnen Aktivitäten in 'priorisierte' und 'nicht priorisierte' Aktivitäten. Damit wird die Arbeitsliste nicht zu überladen mit Varianten

priorisierter Einträge und man behält den Überblick darüber, was wirklich wichtig ist. Allerdings gibt es dabei keine Möglichkeit, die Art der Priorisierung genauer zu beschreiben oder mehrere Stufen von Priorisierungen zu unterscheiden.

Eine weitere Unterteilung der Kategorien ermöglicht es, individuell auf einzelne Priorisierungen zu reagieren. Die Anzahl der Prioritätsstufen ist dabei nahezu unbegrenzt, sollte aber in einem vernünftigen Rahmen bleiben. Zu viele Kategorien erschweren die Darstellung in der Arbeitsliste und werden für den Bearbeiter zu unübersichtlich. Individuelle Reaktionen für jede einzelne Stufe sind nur mit entsprechend großem Aufwand der ALV zu erreichen.

Eine andere Möglichkeit ist, Prioritäten nach Art der Priorisierung festzulegen. In dem Fall gibt es dann z. B. Kategorien für eine überschrittene Bearbeitungsdauer, ein abgelaufenes Zeitfenster oder ein abgelaufener Endtermin. Das hat den Vorteil, dass individuell auf die verschiedenen Arten reagiert werden kann. Die Einbindung neuer Prioritäten wird dagegen schwieriger, weil dann ebenfalls neue Kategorien erstellt werden müssen. Außerdem müssen es die Zeitgeber der Arbeitslistenverwaltung erlauben, dass zwischen ihnen unterschieden werden kann, wodurch ihre Verwendung wiederum aufwändiger wird.

Ausgehend von diesen Betrachtungen ist die Verwendung von einigen wenigen Prioritätskategorien ratsam. Damit werden eine allgemeine Anwendung von Priorisierungen, ein flächendeckender Einsatz der Arbeitslistenverwaltung und eine leichte Anbindung von Arbeitslistenkunden ermöglicht. Es können verschiedene Priorisierungsarten eingesetzt und neu hinzugefügt werden, ohne die Prioritätsstufen selbst zu verändern.

Eine einfache und nicht zu überladene Einteilung der Prioritäten besteht z. B. aus vier Kategorien:

- niedrig – Aktivitäten ohne vorgegebene Zeitbeschränkungen
- normal – nicht priorisierte Aktivitäten
- hoch – Aktivitäten, die Zeitvorgaben überschritten haben
- kritisch – manuell priorisierte Aktivitäten

Damit ist es möglich wesentliche Informationen darzustellen und individuell zu reagieren, ohne die Arbeitsliste mit Informationen zu überladen.

Kennzeichnung der Priorisierung in den Arbeitslisten

Priorisierte Einträge müssen für die Besitzer der Arbeitslisten kenntlich gemacht werden. Dabei sollte die Priorisierung offensichtlich sein, genügend Informationen beinhalten, aber nicht zu überladen wirken. Hier gibt es zahlreiche Anwendungsmöglichkeiten, von denen einige nachfolgend erwähnt werden.

Zusätzliches Attribut: Eine Möglichkeit besteht darin, ein zusätzliches Attribut in der Arbeitsliste darzustellen – z. B. in Form einer extra Spalte – womit die Priorität kenntlich gemacht wird und welches in der Arbeitsliste sofort auffällt. Ein Nachteil dabei ist, dass bei einer großen Arbeitsliste priorisierte Elemente nicht sofort offensichtlich sind und erst durch das Lesen der gesamten Liste gesucht werden müssen. Außerdem muss die Anzahl der verschiedenen Etikettierungen beschränkt werden, sonst verschwimmen diese ineinander und die Bedeutung verblasst.



Abbildung 6-2: Kennzeichnung der priorisierten Aktivitäten durch zusätzliches Attribut

Farbkodierung der Aktivitäten, z. B. rot für priorisierte Einträge. Eventuell kann auch eine Unterscheidung der einzelnen Priorisierungstypen erfolgen. Das Farbschema ist gut erkennbar, hat aber den Nachteil, dass zu viele Farben eher störend und zu unübersichtlich wirken.



Abbildung 6-3: Farbkodierung der priorisierten Aktivitäten

Reihenfolge/Sortierung der Aktivitäten. In der Arbeitsliste des Bearbeiters werden die priorisierten Einträge ganz oben angezeigt und die restlichen, normalen Einträge folgen darunter. Innerhalb einzelner Kategorien kann man die Einträge nach Eingangsdatum oder Zeitpunkt der Priorisierung anordnen. Das ermöglicht dem Bearbeiter einen freien Blick auf die wichtigsten Aktivitäten, hat aber den Nachteil, dass die Grenze zwischen priorisierten und normalen Einträgen nicht sichtbar ist.



Abbildung 6-4: Sortierung der priorisierten Aktivitäten

Am besten ist eine Mischung aus verschiedenen Varianten, vor allem wenn mehrere Prioritätskategorien unterschieden werden sollen. Ein Vorschlag für die oben definierten Kategorien sieht folgendermaßen aus:

- niedrig: Einordnung am Ende der Arbeitsliste. Eventuell ist sogar eine Ausfilterung sinnvoll, so dass niedrig priorisierte Aktivitäten nur ab einer bestimmten (kleinen) Größe der Liste angezeigt werden.
- normal: keine Kennzeichnung
- hoch: Einordnung am Anfang der Arbeitsliste. Ein extra Attribut ist hier sinnvoll, um anzugeben, um welche Art von Priorisierung (Bearbeitungsdauer, usw.) es sich handelt.
- kritisch: Einordnung noch vor den hoch priorisierten Aktivitäten. Um die beiden Gruppen voneinander zu unterscheiden ist hier eine Farbkodierung der kritischen Aktivitäten angebracht.

Eine solche Kennzeichnung von Aktivitäten ermöglicht es, bei sinnvoller Ausnutzung der einzelnen Kategorien, Priorisierungen in den Arbeitslisten darzustellen, ohne dabei die Übersicht zu gefährden. Die normalen Aktivitäten ohne Priorisierung stellen im Normalfall den Großteil der Einträge und werden daher nicht gesondert gekennzeichnet.



Abbildung 6-5: optimale Kennzeichnung von priorisierten Aktivitäten

Die Darstellung der Prioritäten muss von den Arbeitslistenkunden selbst vorgenommen werden, da die ALV keinen Einfluss darauf hat, in welcher Weise die Arbeitsliste dort abgebildet wird. Zum Beispiel kann sie den Klienten nicht dazu zwingen, priorisierte Einträge rot zu kennzeichnen oder eine entsprechende zusätzliche Spalte mit Priorisierungsinformationen anzuzeigen. Die ALV kann den Klienten jedoch bei der Behandlung von Prioritäten unterstützen, indem sie die Informationen darüber zugänglich macht oder Arbeitslisten schon entsprechend filtert oder vorsortiert.

6.4 Priorisierungsinformationen

Um die ordnungsgemäße Priorisierung von Aktivitäten zu gewährleisten, benötigt die Arbeitslistenverwaltung einiges an Informationen.

Zum einen müssen die Priorität und die Priorisierungsinformationen als Eigenschaften einer Aktivität vorhanden sein. Da die Aktivitäten innerhalb der ALV ohnehin unabhängig von den eigentlichen Prozessschritten gespeichert werden, ist die Einführung zusätzlicher Attribute kein Problem. Bestehende Prioritäten oder Informationen, die zu Priorisierungen führen können und im Prozess hinterlegt sind, z. B. Endtermine oder Zeitfenster, müssen bei Aktivierung und Übergabe des Prozessschrittes an die ALV ausgeliefert und dort entsprechend verwaltet werden.

Wichtig sind ebenfalls Informationen über den vorgegebenen zeitlichen Verlauf von Aktivitäten, um die Zeitgeber zu initialisieren. Dafür müssen die maximalen Zeiten für die Bearbeitungsdauer und die Verweildauer von Aktivitäten in Arbeitslisten hinterlegt werden. Das kann sowohl in der Arbeitslistenverwaltung als auch im Prozessmodell geschehen.

Im Prozessmodell lohnt sich die Hinterlegung nur, wenn die Zeiten für jeden Prozessschritt verschieden sind und sich nicht ändern. Dann müssen diese Informationen aber bei Übergabe der Aktivität mitgegeben werden, damit die ALV die Zeitgeber entsprechend konfigurieren kann.

Gelten für alle Aktivitäten gleiche Zeiten, dann können diese permanent in der ALV hinterlegt werden. Damit ist ein schneller Zugriff möglich und der Platzbedarf wird verringert. Allerdings ist diese Lösung denkbar ungünstig, wenn man davon ausgeht, dass die einzelnen Aktivitäten verschieden komplex sind und die Abarbeitung dementsprechend verschieden lange dauert.

Eine dritte Möglichkeit besteht darin, die Informationen für jede Aktivität dynamisch in der ALV zu verwalten. Das nimmt zwar einiges an Speicherplatz in Anspruch, jedoch kann darauf schnell zugegriffen werden und die Zeiten können für jede Aktivität individuell bestimmt werden. Außerdem ist es dabei auch möglich, die Zeitvorgaben zur Laufzeit anzupassen.

6.5 Zusammenfassung

Im vorliegenden Kapitel wurde die Priorisierung von Aktivitäten innerhalb der Arbeitslistenverwaltung untersucht. Dabei wurde erörtert, was Priorisierung innerhalb der ALV bedeutet und welche Auswirkungen sie auf die Arbeitslisten der Bearbeiter hat. Außerdem wurde geprüft, welche Informationen die Arbeitslistenverwaltung benötigt, um Priorisierungen durchzuführen und wo diese Informationen hinterlegt werden.

Die Priorisierung von Aktivitäten ist wichtig und unerlässlich, um für die schnelle und zeitnahe Abarbeitung von Aktivitäten zu sorgen. Die Arbeitslistenverwaltung muss aber einiges an Logik und Hintergrundinformationen zu Verfügung haben, um Priorisierungen zu ermöglichen.

Eine automatische Priorisierung kann aufgrund ihres Zeitbezuges nur über Zeitgeber erfolgen. Die Arbeitslistenverwaltung muss diese mit den zur Verfügung stehenden Informationen starten und auf ihren Ablauf mit der Priorisierung der zugehörigen Aktivitäten reagieren.

Eine angemessene Reaktion auf und Darstellung von Prioritäten ist nur möglich, wenn zuvor Kategorien definiert wurden. Eine günstige Einteilung in die vier Prioritätsstufen 'niedrig', 'normal', 'hoch' und 'kritisch' wurde vorgestellt und eine optimale Kennzeichnung vorgeschlagen.

Die Kennzeichnung der priorisierten Aktivitäten fällt jedoch nur bedingt in den Zuständigkeitsbereich der ALV. Prioritäten müssen intern verwaltet und bei Bedarf an den Klienten ausgeliefert werden. Für die genaue Darstellung ist der Klient selbst verantwortlich. Die Arbeitslistenverwaltung kann diesen aber zumindest dahingehend unterstützen, dass sie eine Arbeitsliste entsprechend vorsortiert oder filtert.

Priorisierungsinformationen sind meist vom jeweiligen Prozessschritt abhängig und werden dann idealerweise auch im Prozessmodell hinterlegt. Das trifft vor allem zu, wenn sich die Informationen selten ändern. Auf der anderen Seite ist es sinnvoll, allgemeinere Priorisierungsinformationen, z. B. die Verweildauer von Prioritäten in Arbeitslisten, in der ALV selbst zu verwalten. Das ermöglicht auch eine dynamische Anpassung der Einstellungen an Umgebungsbedingungen, wie z. B. die Auslastung der Arbeitslisten.

7 Entwurf

In diesem Kapitel werden aufbauend auf der Untersuchung des Problembereiches (der Konzeption) konkrete Lösungsvorschläge gegeben, wie die gewünschte Arbeitslistenverwaltung implementiert werden kann. Dabei werden die Aspekte aus den vorangegangenen Kapiteln in einen Entwurf für eine Arbeitslistenverwaltung überführt.

Ein großes Thema innerhalb der Forschungsarbeit für WfMS ist die komponentenbasierte Entwicklung von Workflow-Anwendungen. Wie in Abschnitt 2.2 (Das Workflow-Referenzmodell) festgestellt wurde, besteht ein WfMS aus einer Kernkomponente, welche die Basisfunktionalität für die Abarbeitung von Prozessen enthält. Ausgehend davon gibt es verschiedene Komponenten, die erweiterte Funktionalität bereitstellen. Dazu gehören z. B. Prozessdefinitionswerkzeuge oder auch die Arbeitslistenverwaltung. Diese Komponenten sollen möglichst unabhängig sein, so dass man sie problemlos in eine bereits bestehende Umgebung integrieren kann.

Für die ALV bedeutet das im Wesentlichen, dass sie einmal eine Schnittstelle zum WfMS besitzen muss, die möglichst generisch ist. Die Schnittstelle wird vom WfMS vorgegeben und muss auf der Seite der ALV allgemein gehalten werden, um möglichst viele Workflow-Systeme unterschiedlicher Hersteller in Anspruch nehmen zu können. Das ist beispielsweise notwendig, wenn ein definierter Prozess sich über mehrere Unternehmen erstreckt [LeRo00]. Jedes dieser Unternehmen verwendet potentiell verschiedene WfMS. Auch innerhalb eines Unternehmens können verschiedene Workflow-Systeme im Einsatz sein, vor allem bei einer dezentralen Organisation, die global verteilt ist. Die Arbeitslistenverwaltung kann in diesen Fällen dazu verwendet werden die Aktivitäten der verschiedenen Workflow-Systeme an einem zentralen Punkt zu sammeln und zu verteilen.

Die Schnittstelle zum Arbeitslistenklienten liegt dagegen in der Verantwortung der ALV und muss vom Klienten implementiert werden. Um viele verschiedene Klienten versorgen zu können, die unterschiedliche Funktionalität verlangen, muss die ALV entsprechend mehrer Schnittstellen anbieten. Außerdem muss entschieden werden, welche der gewünschten Aufgaben von der ALV selbst übernommen werden und welche nach Bereitstellung der erforderlichen Informationen auch vom Klienten erfüllt werden können.

Ein dritter Aspekt sind die internen Schnittstellen der ALV. Sie stellen eine gewisse Grundfunktionalität hinsichtlich Bearbeiterzuordnung beziehungsweise Arbeitslisten- und Aktivitätenverwaltung zu Verfügung. Hier soll es möglich sein die Grundfunktionen durch eigene Komponenten zu erweitern, um so benutzerdefinierte Verteilungsverfahren oder Priorisierungsmechanismen dynamisch integrieren zu können.

Abbildung 7-1 zeigt ein abstraktes Architekturbild, welches einen Überblick über die in diesem Kapitel behandelten Schnittstellen darstellt und sie in den Rahmen der Arbeitslistenverwaltung eingliedert.

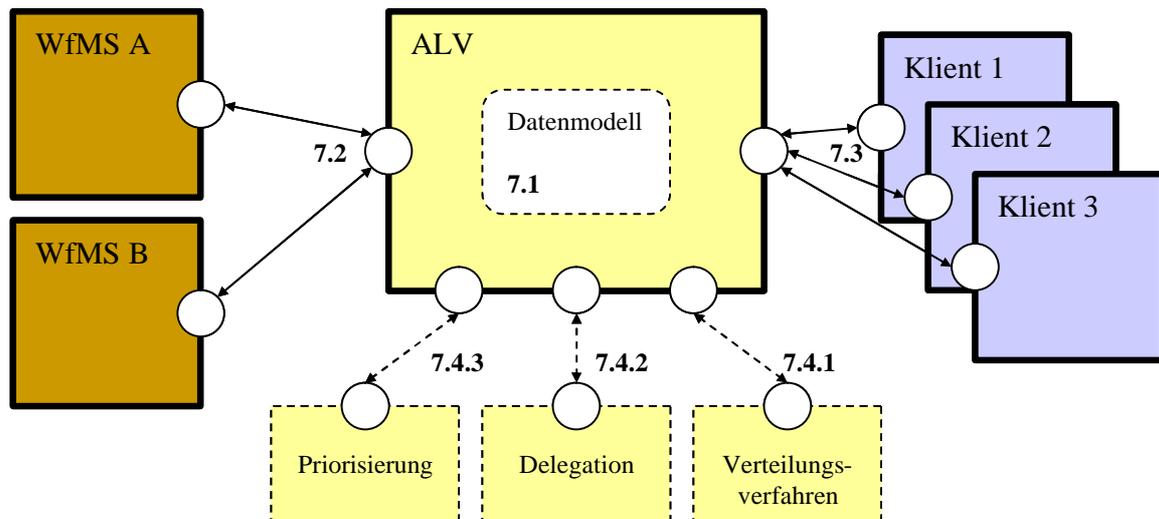


Abbildung 7-1: Schnittstellen und Datenmodell der Arbeitslistenverwaltung

In Abschnitt 7.1 wird das Datenmodell der Arbeitslistenverwaltung entworfen. Es enthält die Daten, die von der ALV benötigt und verwaltet werden, um ihre Aufgabe angemessen zu erledigen.

Als nächstes werden die Schnittstellen untersucht, die es der Arbeitslistenverwaltung erlauben mit verschiedenen WfMS und Arbeitslistenklienten zu interagieren. In Abschnitt 7.2 geht es um die Verbindung zu den WfMS. Dabei werden zuerst die Schnittstellen bereits existierender Systeme untersucht, um nachfolgend eine allgemein verwendbare Schnittstelle für die ALV zu entwerfen. Abschnitt 7.3 behandelt die Anbindung von Klientanwendungen an die Arbeitslistenverwaltung. Wichtig ist, dass die ALV in ihren Schnittstellen möglichst unabhängig ist und somit mit verschiedenen WfMS als auch unterschiedliche Klienten kommunizieren kann.

Zum Schluss werden in Abschnitt 7.4 interne Schnittstellen betrachtet, die sich um die in der Konzeption behandelten Aspekte der Verteilungsverfahren, Delegation und Priorisierung kümmern. Eine besondere Anforderung hierbei ist die Vorgabe, dass die Schnittstellen es erlauben müssen, externe Komponenten einfach anzubinden, um somit z. B. neue Verteilungsverfahren innerhalb der ALV verwenden zu können.

7.1 Datenmodell

Die Arbeitslistenverwaltung muss zur Erfüllung ihrer Aufgaben eine Reihe von Informationen vorrätig halten. Diese Informationen bekommt sie entweder vom WfMS zugeteilt oder hinterlegt sie selbst in ihrem Datenspeicher. Aus den Untersuchungen aus Abschnitt 3.3 – Verwaltung von Arbeitslisten und Aktivitäten – kann folgendes Datenmodell abgeleitet werden:

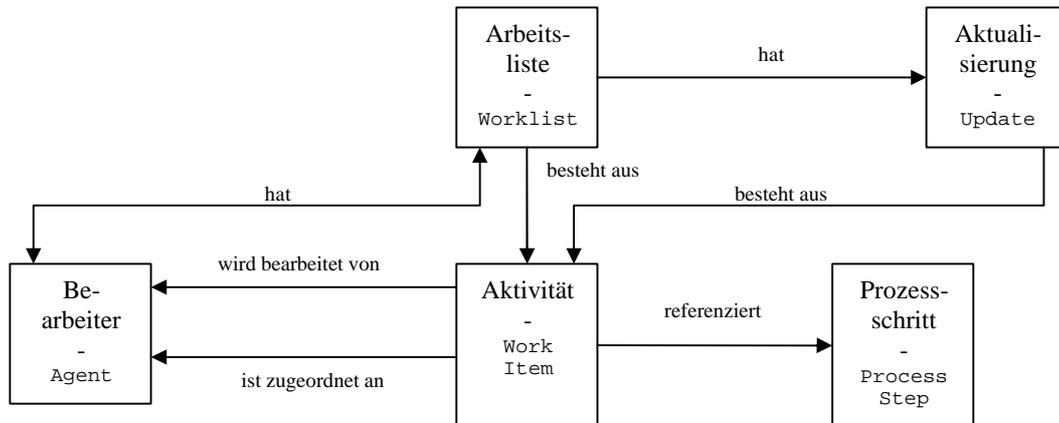


Abbildung 7-2: Das Datenmodell der Arbeitslistenverwaltung

Im Mittelpunkt der ALV steht die Arbeitsliste eines Bearbeiters. Neben den obligatorischen Arbeitslisteneinträgen in Form von Aktivitäten besitzt sie einen Behälter, welcher den Unterschied zwischen interner Arbeitsliste und zuletzt ausgelieferter Klientenarbeitsliste enthält. Dieser Aktualisierungsbehälter enthält Informationen über alle Aktivitäten, die seit der letzten Auslieferung neu hinzugekommen sind, geändert oder aus der Arbeitsliste entfernt wurden. Stellt der Klient nun eine Anfrage nach Aktualisierung seiner Arbeitsliste, wird nur der Inhalt dieses Behälters übermittelt.

Eine Arbeitsliste besteht aus mehreren eingetragenen Aktivitäten. Eine Aktivität in der Arbeitslistenverwaltung resultiert aus dem aktivierten Prozessschritt, der vom WfMS übergeben wurde. Einen Teil ihrer Attribute, wie z. B. die zugehörige Bearbeiterformel, Zustandsinformationen und eventuell bereits existierende Prioritäten erhält sie daher schon bei der Konstruktion. Weitere arbeitslistenspezifische Attribute werden der Aktivität im Laufe der Verarbeitung zugewiesen. Darunter befinden sich durch Verteilungsverfahren ausgewählte Bearbeiter, Statusinformationen bezüglich Delegation und Eskalation oder in der ALV erlangte Prioritäten. Jede Aktivität erhält außerdem noch eine Referenz auf den originalen Prozessschritt, damit Informationen zwischen ALV und WfMS ausgetauscht werden können.

Wichtig sind ebenfalls die verschiedenen Bearbeitermengen, die einer Aktivität zugewiesen werden. Das ist einmal die Bearbeiterformel, die vom WfMS mitgeliefert wird. Daneben gibt es die Menge der durch Verteilungsverfahren zugeordneten Bearbeiter. Zuletzt ist auch der Bearbeiter, welcher die Aktivität startet, gesondert hinterlegt.

Im Folgenden werden die Entitäten einzeln genauer betrachtet und die obligatorischen (+) und optionalen (–) Attribute angegeben, welche für die Arbeit der Arbeitslistenverwaltung nötig sind.

Arbeitsliste: Eine Arbeitsliste kann einem bestimmten Benutzer bzw. einer Gruppe von Benutzern zugewiesen sein oder einer bestimmten Stelle als Gruppenarbeitsliste dienen.

`Worklist(ID, agents, orgPos, workItems, capacity, updates, groupList)`

- (+) `ID` – eindeutiger Bezeichner der Arbeitsliste
- (+) `agents` – Benutzer, denen diese Arbeitsliste zugewiesen ist

- (+) orgPos – Stelle, der diese Arbeitsliste zugewiesen ist
- (+) workItems – Liste der Aktivitäten in der Arbeitsliste
- (-) capacity – Auslastungsgrenze/Leistungsindex der Arbeitsliste
- (+) updates – Behälter mit Aktualisierungsinformationen für jeden Bearbeiter
- (+) groupList – zeichnet die Arbeitsliste als Gruppen- oder persönliche Liste aus

Wie in Abschnitt 3.3 beschrieben ist es sinnvoll, dem Klienten eine eigene Schnittstelle für seine Arbeitsliste zur Verfügung zu stellen, so dass er nur lesenden Zugriff auf die für ihn wichtigen Informationen bekommt. Dazu gibt es zwei weitere Schnittstellen, die von `Worklist` erben: eine interne Arbeitsliste, die änderbar ist und der ALV zur Bearbeitung zur Verfügung steht (`InternalWorklist`) und eine Klienten-Arbeitsliste, die dem Arbeitslistenklienten zur Verfügung steht, um Informationen abzurufen (`ClientWorklist`). Die `ClientWorklist` ist nicht direkt manipulierbar und kann nur durch eine Aktualisierung geändert werden.

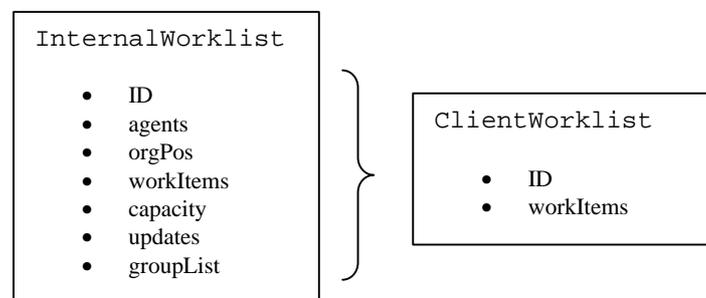


Abbildung 7-3: Unterschiedliche Ausprägungen von Arbeitslisten

Ob als persönliche oder Gruppenarbeitsliste, die Arbeitsliste ist immer einer bestimmten Stelle zugewiesen. Daher kann neben der Arbeitslistenkennung (`ID`) auch die Kennung der Stelle (`orgPos`) schon während der Konstruktion festgelegt werden und keine der drei Schnittstellen benötigt Methoden, um diese Attribute zu setzen. Bei einer persönlichen Arbeitsliste kann auch die Bearbeiterkennung (`agents`) bei der Erstellung bereits zugewiesen werden. Bei einer Gruppenarbeitsliste können die einzelnen Bearbeiter wechseln, daher ist hier eine vorgezogene Festlegung nicht möglich.

Aktualisierung: Die Aktualisierung einer Arbeitsliste enthält eine Liste der geänderten Aktivitäten, und wird bei Bedarf an den Klienten übertragen. Jede der Aktivitäten in der Aktualisierungsliste enthält einen Status, der besagt, ob die Aktivität hinzugefügt, gelöscht oder einfach nur geändert (z. B. priorisiert) wurde. Jede geänderte Aktivität wird eingetragen, sobald eine Änderung eintritt und bei einer Aktualisierung muss nur noch der Inhalt des Behälters (d. h. die Liste der Aktivitäten) dem Klienten übermittelt werden.

`Update(ID, worklist, agent, updatedWorkItems)`

- (+) ID – die eindeutige Kennung der Aktualisierung
- (+) worklist – die Arbeitsliste, zu der die Aktualisierung gehört
- (+) agent – der Bearbeiter, zu dem die Aktualisierung gehört
- (+) updatedWorkItem – Liste aus aktualisierten Aktivitäten und Status

Die Schnittstelle `Update` wird von der ALV implementiert und verwendet. Der Klient bekommt als Aktualisierung nur die Liste der aktualisierten Aktivitäten. Eine Aktualisierung bezieht sich immer auf die Arbeitsliste eines bestimmten Bearbeiters, daher können die Kennung (ID), die zugehörige Arbeitsliste (`worklist`) und der zugehörige Bearbeiter (`agent`) bereits bei der Konstruktion des `Update`-Objekts zugewiesen werden. Typischerweise wird solch ein Objekt erstellt, wenn sich der Bearbeiter an der ALV anmeldet.

Der Aktualisierungszustand einer Aktivität muss als Konstante in der ALV hinterlegt sein. Daraus ergibt sich folgende Einteilung:

`WorkItemUpdateState`:

ADDED: die Aktivität wurde der Arbeitsliste hinzugefügt
 REMOVED: die Aktivität wurde aus der Arbeitsliste entfernt
 CHANGED: die Aktivität wurde geändert (z. B. ihre Priorität)

Aktivität: kann von bestimmten Bearbeitern bearbeitet werden und wird ihnen deshalb in die Arbeitsliste gestellt. Die Aktivität wird, sobald sie im Prozessverlauf aktiviert wird, an die Arbeitslistenverwaltung übergeben. Sie unterscheidet sich von einem Prozessschritt im WfMS dahingehend, dass sie für die Dauer ihrer Bearbeitung zusätzliche Attribute besitzt, die nicht zum eigentlichen Prozessmodell gehören. Sie behält jedoch eine Referenz auf den ursprünglichen Prozessschritt, um Statusänderungen vom WfMS entgegennehmen zu können.

Eine Aktivität besitzt neben der zugeordneten Bearbeiterformel eine Bearbeitermenge, die aus der Anwendung eines Verteilungsverfahrens resultiert. Ein Bearbeiter dieser Menge startet die Aktivität und bekommt sie damit zur Bearbeitung zugewiesen. Des Weiteren besitzt jede Aktivität Informationen über Zeitvorgaben, Prioritäten, Auslastungen oder Delegationen.

`WorkItem(ID, activationDate, processStepID, assignedAgent, complexity, deadline, delegationAddressee, delegationLevel, delegationLimit, description, distributionDate, escalationAddressee, escalationLevel, escalationLimit, priority, selectedAgents, staffAssignment, state)`

- (+) ID – eindeutige Kennung der Aktivität
- (-) activationDate – Zeitpunkt des Startens der Aktivität
- (+) processStepID – Referenz auf den Prozessschritt aus dem WfMS
- (+) assignedAgent – für die Bearbeitung zugewiesener Bearbeiter
- (-) complexity – Komplexitätsindex der Aktivität für die Auslastung
- (-) deadline – vorgegebener Fertigstellungszeitpunkt der Aktivität
- (-) description – Beschreibung der Aktivität
- (-) delegationAddressee – Delegationsempfänger
- (-) delegationLevel – die Delegationsstufe der Aktivität
- (-) delegationLimit – die Obergrenze der Anzahl von Delegationen
- (-) distributionDate – Zeitpunkt der Erstverteilung der Aktivität
- (-) escalationAddressee – Eskalationsempfänger
- (-) escalationLevel – die Eskalationsstufe der Aktivität
- (-) escalationLimit – die Obergrenze der Anzahl von Eskalationen
- (-) priority – Priorität der Aktivität (Standard: Normal)

- (+) selectedAgents – durch Verteilungsverfahren gewählte mögliche Bearbeiter
- (+) staffAssignment – Bearbeiterformel für die Aktivität
- (+) state – Bearbeitungszustand der Aktivität

Wie bei der Arbeitsliste ist es auch beim Arbeitslisteneintrag sinnvoll zwei verschiedene Schnittstellen für ALV und Klienten zur Verfügung zu stellen: `InternalWorkItem` und `ClientWorkItem`. Beide werden von `WorkItem` abgeleitet.

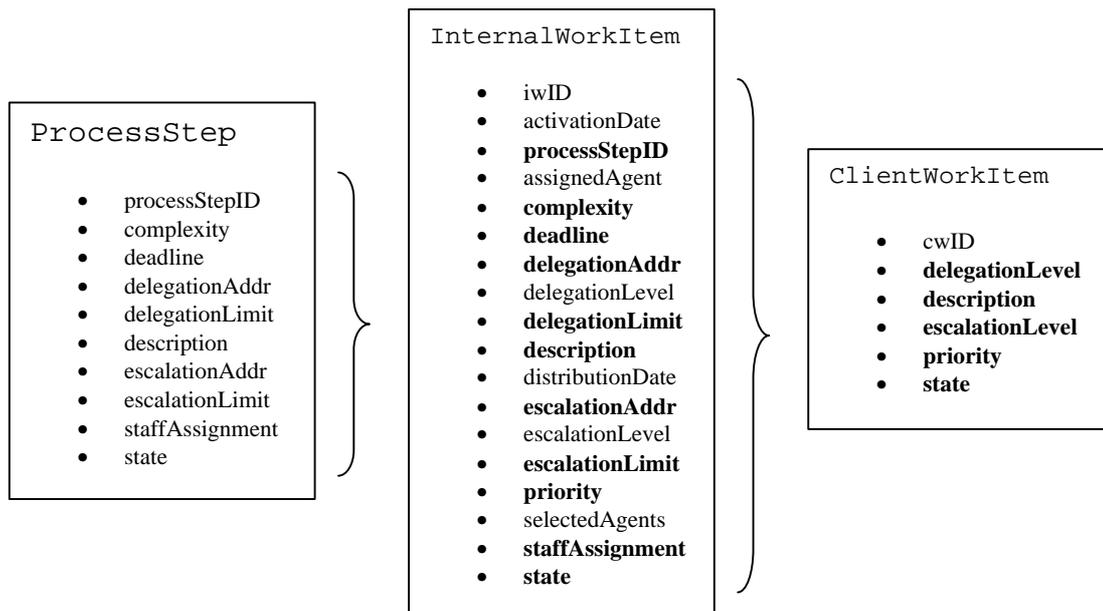


Abbildung 7-4: Unterschiedliche Ausprägungen von Arbeitslisteneinträgen

Die Referenz auf den Prozessschritt und die Bearbeiterformel werden vom WfMS übergeben, sobald der Prozessschritt aktiviert wird. Daher können neben der Kennung (ID) sowohl `processStepID` als auch `staffAssignment` über den Konstruktor festgelegt werden. Delegations- und Eskalationsempfänger bzw. -grenzen, Beschreibung, Priorität, Komplexität, Endtermin und Zustand werden ebenfalls im Prozessmodell hinterlegt bzw. stehen bei Aktivierung des Prozessschrittes bereits fest.

Prozessschritt: Eine Referenz auf den Prozessschritt, dem eine Aktivität entspricht. Sie wird benötigt, um mit dem WfMS zu kommunizieren und Informationen über Prozessschritte auszutauschen.

`ProcessStep(ID, workItemID)`

- (+) ID – die eindeutige Kennung des Prozessschrittes
- (+) workItemID – das zugeordnete `WorkItem`

Die Schnittstelle `ProcessStep` wird von der ALV verwendet, um die Attribute des `WorkItems` aktuell zu halten, wenn sich Änderungen am Prozessschritt im WfMS ergeben. `ID` und `workItemID` werden beim Konstruktoraufruf gesetzt und bleiben für die Lebensdauer der Aktivität unverändert bestehen.

Bearbeiter: Mitarbeiter, die eine oder mehrere Arbeitslisten besitzen (persönliche und Gruppenarbeitslisten).

`Agent(ID, orgPos, pWorklist, gWorklists)`

- (+) `ID` – eindeutige Kennung des Bearbeiters
- (+) `orgPos` – Stelle des Bearbeiters, an der er aktuell aktiv ist
- (+) `pWorklist` – persönliche Arbeitsliste des Bearbeiters
- (+) `gWorklists` – Liste der auf dem Bearbeiter laufenden Gruppenarbeitslisten

Die folgende Abbildung fasst noch einmal zusammen, auf welche Daten die Arbeitslistenverwaltung und der Arbeitslistenklient Zugriff haben. In den nächsten Abschnitten, während der Untersuchung der Schnittstellen, werden für die verschiedenen Elemente des Datenmodells die englischen Begriffe verwendet, sofern es sich dabei um Datenobjekte handelt.

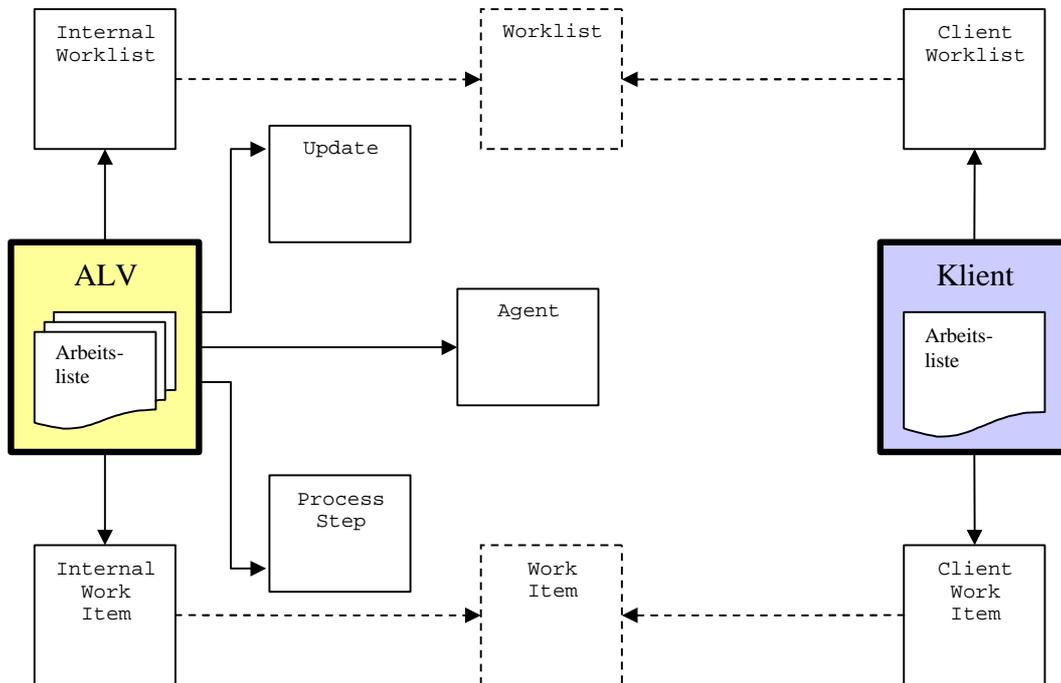


Abbildung 7-5: Zugriffe auf Daten im Bereich der ALV

7.2 Schnittstelle zum WfMS

Die Schnittstellen zwischen der Arbeitslistenverwaltung und den WfMS sind hauptsächlich dafür verantwortlich, Informationen über aktivierte Prozessinstanzen und ihren Bearbeitungsfortschritt auszutauschen. Im Abschnitt 7.2.1 wird zunächst überprüft, welche Anforderungen die ALV genau an eine Schnittstelle zum WfMS stellt.

Der Entwurf einer generischen Schnittstelle zwischen ALV und WfMS hängt in großem Maße davon ab, welche Schnittstellen bereits vom WfMS zur Verfügung gestellt werden. Dabei hat jeder Hersteller sein eigenes System, was die Entwicklung einer generischen Schnittstelle schwierig macht. Ob es dennoch machbar ist soll in den Abschnitten 7.2.2 bis 7.2.5 erörtert werden. Dabei werden drei bereits existierende, teils kommerzielle WfMS daraufhin untersucht, welche öffentlichen Schnittstellen von ihnen angeboten werden und wie man damit den Anforderungen der Arbeitslistenverwaltung genügen kann.

Die Ergebnisse der Untersuchungen werden in Abschnitt 7.2.6 zum Entwurf einer Schnittstelle für die ALV verwendet.

7.2.1 Anforderungen

Die Kommunikation zwischen WfMS und ALV bewegt sich in zwei Richtungen: das Workflow-System übermittelt der Arbeitslistenverwaltung neue Aktivitäten, die bearbeitet werden sollen, informiert sie über Zustandsänderungen dieser Aktivitäten und gibt vor, welche Bearbeiter für die Ausführung der Aktivität in Frage kommen. Die ALV ihrerseits stellt Anfragen an das Organisationsmodell innerhalb des WfMS, um Bearbeiterformeln aufzulösen und Informationen über Bearbeiter zu bekommen.

Daraus leiten sich folgende Anforderungen an eine Schnittstelle ab, die erfüllt sein müssen, um die ALV-Komponente erfolgreich an ein WfMS anzubinden.

Übermittlung einer neuen Aktivität an die ALV

Um ihre Arbeit überhaupt ausführen zu können muss die Arbeitslistenverwaltung über aktivierte Prozessschritte informiert werden. Diese Prozessschritte und alle damit verknüpften, relevanten Informationen werden der Arbeitslistenverwaltung zur weiteren Verarbeitung und zur Zuteilung an Bearbeiter übergeben.

Die erste Aufgabe nach Aktivierung eines Prozessschrittes ist, diesen in eine Aktivität, d. h. einen Arbeitslisteneintrag (`WorkItem`) umzuwandeln. Der Hauptunterschied dieser beiden Objekte besteht darin, dass ein Arbeitslisteneintrag mit einer Menge realer, zugewiesener Bearbeiter und einer spezifizierten Anwendung verknüpft ist, die zur Ausführung der Aufgabe des `WorkItems` gestartet wird. [LeRo00] Des Weiteren enthält ein `WorkItem` eine Reihe von zusätzlichen Attributen, welche im Prozessschritt nicht vorhanden sind und nach der Aktivierung oder während der Laufzeit gesetzt werden müssen. Dazu zählen beispielsweise das Verteilungsdatum und die Delegationsstufe.

Die Erstellung eines `WorkItems` kann vom WfMS oder von der ALV vorgenommen werden. Wird es von WfMS erstellt, muss das Workflow-System der ALV alle notwendigen Informationen übermitteln. Das hat den Vorteil, dass die Arbeitslistenverwaltung keine zusätzlichen Daten über den Prozessschritt speichern muss, um Informationen mit dem WfMS auszutauschen. Sie bekommt nur das erstellte `WorkItem` zugewiesen und kann damit ihre Arbeit ausführen. Die Referenz auf den Prozessschritt wird vom WfMS gehalten (`ProcessStep`). Außerdem liegen dort dann auch die Informationen über das verknüpfte Anwendungsprogramm, die vom Workflow-System benötigt werden, um `WorkItems` ordnungsgemäß zu starten.

Werden die `WorkItems` von der ALV erstellt, so muss diese vom WfMS den Prozessschritt übergeben bekommen und eine Referenz darauf hinterlegen. Der Vorteil ist, dass die ALV relativ unabhängig vom WfMS arbeiten kann, da keine Vorgaben gemacht werden, wie ein `WorkItem` aufgebaut sein muss und welche Attribute es besitzt. Probleme gibt es bei dieser Variante, wenn ein Bearbeiter ein `WorkItem` startet. In dem Fall sind die Informationen über das verknüpfte Anwendungsprogramm beim `WorkItem` in der ALV hinterlegt. Das WfMS, welches das Programm starten soll muss daher die Möglichkeit haben, die benötigten Informationen über den Prozessschritt abzuleiten.

Idealerweise wird das `WorkItem` von der ALV erstellt und hinterlegt. Das erlaubt es ihr unabhängig von den Vorgaben eines WfMS zu arbeiten und dadurch mehrere Workflow-Systeme zu unterstützen.

Generell ist es möglich ein eigenes `WorkItem` für jeden Bearbeiter zu erstellen oder ein einzelnes `WorkItem` anzulegen, welches den Bearbeitern als Referenz übergeben wird. Aus den Untersuchungen in Abschnitt 3.3 geht hervor, dass ein einzelnes `WorkItem` ressourcenschonender ist und daher bevorzugt wird.

Kompatibilität von Aktivitätszuständen zwischen ALV und WfMS

Ein `WorkItem` kann während seiner Lebenszeit verschiedene Zustände annehmen. Diese Zustände müssen zwischen WfMS und ALV übertragen werden. Die Übermittlung von Zustandsänderungen zwischen zwei unabhängigen Komponenten birgt jedoch ein großes Problem. Die Zustandsmodelle beider Systeme sind im Normalfall verschieden und müssen erst aufeinander abgebildet werden. Das ist Aufgabe der ALV. Sie muss ihr eigenes Zustandsmodell so abstrahieren, dass es mit dem des WfMS kompatibel ist. Dazu müssen sämtliche Zustände des Workflow-Systems entsprechend vor der Verarbeitung umgewandelt werden. Ein einfaches Zustandsmodell für `WorkItems` wird in der folgenden Abbildung dargestellt.

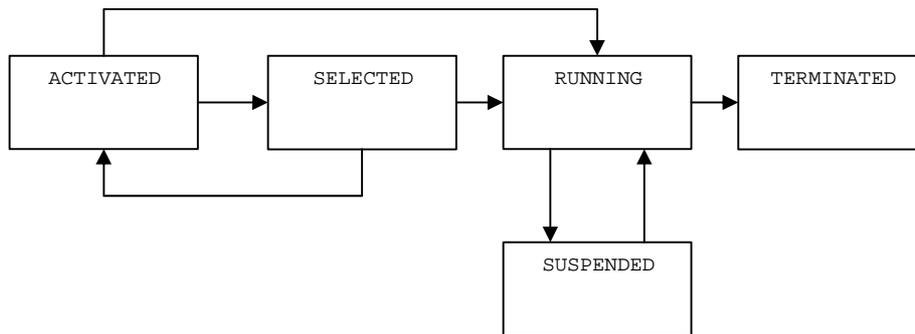


Abbildung 7-6: Zustandsmodell für WorkItems

Bei seiner Erstellung befindet sich das `WorkItem` im Zustand `ACTIVATED`, d. h. es wurde an Bearbeiter ausgeliefert und steht zur Bearbeitung zur Verfügung. Ein Bearbeiter kann nun ein `WorkItem` sofort starten oder erst reservieren und danach starten. Dabei gelangt es in die Zustände `SELECTED` bzw. `STARTED`. Eine Reservierung kann rückgängig gemacht werden und führt wieder in den Zustand `ACTIVATED`. Die Bearbeitung eines `WorkItems` kann unterbrochen (`SUSPENDED`) und später wieder aufgenommen werden. Wird die Abarbeitung eines `WorkItems` beendet tritt es in den Zustand `TERMINATED` und wird danach aus der Arbeitslistenverwaltung gelöscht.

Die einzelnen Zustände müssen als Konstanten in der ALV hinterlegt werden. Folgende Einteilung ergibt sich daraus:

`WorkItemState`:

<code>ACTIVATED</code> :	die Aktivität wurde an Bearbeiter ausgeliefert
<code>SELECTED</code> :	die Aktivität wurde von einem Bearbeiter reserviert
<code>RUNNING</code> :	die Aktivität befindet sich gerade in Bearbeitung
<code>TERMINATED</code> :	die Abarbeitung der Aktivität wurde beendet
<code>SUSPENDED</code> :	die Abarbeitung der Aktivität wurde unterbrochen

Änderung von Aktivitätszuständen

Über seine Arbeitsliste ist es dem Klienten möglich, einzelne Aktivitäten zu starten, zu beenden oder andere Zustandsänderungen vorzunehmen. Diese müssen dem WfMS übermittelt werden, damit es die notwendigen Daten und das verknüpfte Anwendungsprogramm zur Verfügung stellen kann. Dabei kann der Klient direkt mit dem WfMS oder über die Arbeitslistenverwaltung kommunizieren.

Werden Zustandsänderungen über die Arbeitslistenverwaltung vorgenommen, bedeutet das, dass der Klient die Aktivität über eine definierte Schnittstelle bei der ALV reserviert, startet, beendet, usw. Die ALV muss diese Aktionen dann an das WfMS weiterleiten und kann dann die Arbeitslisten der betroffenen Bearbeiter anpassen.

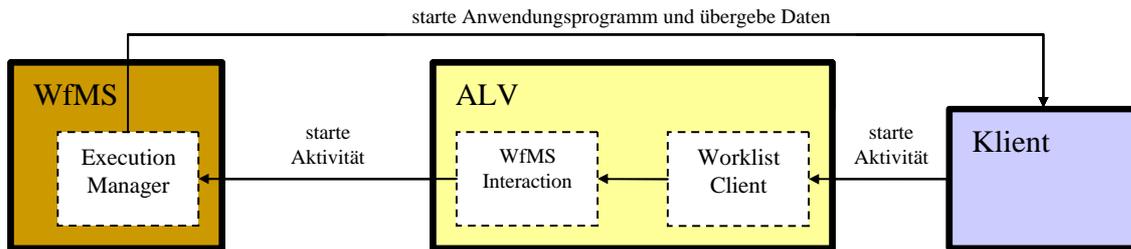


Abbildung 7-7: Übermittlung von Zustandsänderungen über die ALV

Der Vorteil dabei ist, dass die ALV direkt über Zustandsänderungen informiert wird und entsprechend schnell reagieren kann. Der Nachteil ist, dass redundante Aufrufe und Schnittstellen verwendet werden, wodurch die eigentliche Änderung des Zustands langsamer vorangeht. Außerdem liegt die Entscheidung, ob eine Zustandsänderung möglich ist beim WfMS und die ALV muss mit der Weiterverarbeitung warten, bis eine entsprechende Antwort gegeben wird.

Die zweite Möglichkeit besteht darin, dass der Klient direkt mit dem WfMS kommuniziert. Dazu muss eine Sitzung zwischen diesen beiden aufgebaut werden oder bereits vorhanden sein. Das Workflow-System übermittelt die Zustandsänderungen dann an die ALV zur Verarbeitung.

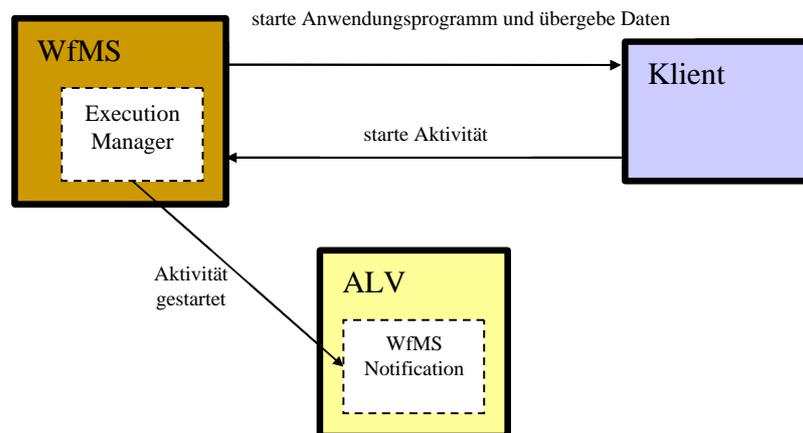


Abbildung 7-8: Übermittlung von Zustandsänderungen über das WfMS

Der Vorteil dieser Variante ist, dass die ALV nur über erfolgreiche Zustandsänderungen informiert wird und sofort reagieren kann, ohne erst auf weitere Benachrichtigungen des WfMS zu warten. Außerdem werden lange Befehlsketten so vermieden und die Arbeitslistenverwaltung hat weniger Aufwand.

Somit ist die günstigere Variante die direkte Kommunikation des Klienten mit dem WfMS. Das verringert insgesamt den Datenverkehr und den Kommunikationsaufwand. Außerdem wird der Aufwand bei der ALV reduziert, da nur noch relevante Informationen über Zustandsänderungen übermittelt und somit verarbeitet werden müssen.

Übermittlung geänderter Prozessschritte

Änderungen am Prozessmodell oder an den Attributen eines bereits aktivierten Prozessschrittes müssen der Arbeitslistenverwaltung übermittelt werden, damit diese darauf reagieren kann. Solche Änderungen umfassen z. B. Bearbeiterformeln, Delegationsempfänger, Prioritäten oder Zeitvorgaben. Die Schnittstelle für diese Übergabe muss vom WfMS gestellt werden. Prinzipiell können alle Attribute eines Prozessschrittes sich während der Laufzeit ändern. Daher muss abgeschätzt werden, ob die Schnittstelle bei jeder Änderung den gesamten Prozessschritt übergibt oder nur die geänderten Attribute.

Die Übergabe einzelner Attribute setzt voraus, dass eine mit dem Datenmodell kompatible Schnittstelle existiert, über die jedes Attribut eines Prozessschrittes übergeben werden kann. Das ist jedoch problematisch, da es die ALV zu sehr auf die Verwendung eines bestimmten Typs von Prozessschritt und damit eines bestimmten WfMS festlegt. Außerdem kann eine solche Schnittstelle unter Umständen sehr umfangreich werden, je nachdem wie viele Attribute ein Prozessschritt besitzt.

Die Übergabe des gesamten Prozessschrittes reduziert den Umfang der Schnittstelle auf ein Minimum und ermöglicht es, auch andere WfMS anzubinden, deren Prozessschritte sich in ihrer Struktur unterscheiden. Der Nachteil ist, dass die Übergabe eines ganzen Prozessschrittes mehr Datenverkehr erzeugt als die eines einzelnen Attributes. Außerdem muss dabei von der ALV überprüft werden, welche Attribute sich im Einzelnen geändert haben.

Dennoch ist es für eine möglichst generische Schnittstelle von Vorteil bei Änderungen jeweils den gesamten Prozessschritt zu übertragen.

Die Änderung von Prozessschritten zur Laufzeit wird derzeit nur von wenigen WfMS unterstützt und angeboten. Darunter befinden sich u. a. Ultimus BPM [Ultimus], Lexign Flow [Lexign] und ADEPT2 [ADEPT2]. Eine entsprechende Schnittstelle sollte aber für eine flächendeckende Unterstützung trotzdem von der ALV bereitgestellt werden.

Zugriff auf das Organisationsmodell

In verschiedenen Situationen kann es für die ALV wichtig sein, dass sie Zugriff auf das Organisationsmodell des WfMS bekommt, z. B. um Benutzer zu authentifizieren, Informationen über sie abzurufen oder Bearbeitermengen aufzulösen. Dazu muss eine entsprechende Schnittstelle von Organisationsmodell angeboten werden, die von der ALV in Anspruch genommen werden kann.

7.2.2 ADEPT2

ADEPT2 ist ein Workflow-Management-System der nächsten Generation und wird am Institut für Datenbanken und Informationssysteme der Universität Ulm entwickelt. Das ADEPT2-System befindet sich aktuell noch in der Entwicklungsphase und weist daher noch keine endgültig definierten Schnittstellen für die Verbindung zur Arbeitslistenverwaltung auf. Das hat den

Vorteil, dass beim Entwurf der Schnittstelle für die ALV einige Freiheiten bestehen, um selbst Vorgaben für das WfMS und den Aufbau seiner Schnittstelle zu machen. Ebenso gibt es noch keine endgültige technische Dokumentation. Die Beschreibung der Schnittstellen ist jedoch als Java-Dokumentation vorhanden. [ADEPT2]

Für die Interaktion mit einer Arbeitslistenverwaltung bietet ADEPT2 bereits zwei vordefinierte Schnittstellen: `WorklistInteraction` und `WorklistNotification`. Letztere sorgt dafür, dass die Arbeitslistenverwaltung asynchron Nachrichten und Daten vom WfMS empfangen kann und muss zur weiteren Verarbeitung von der ALV implementiert werden. `WorklistInteraction` erlaubt einer Arbeitslistenverwaltung sich beim System an- und abzumelden und selbstständig Anfragen nach Daten zu stellen.

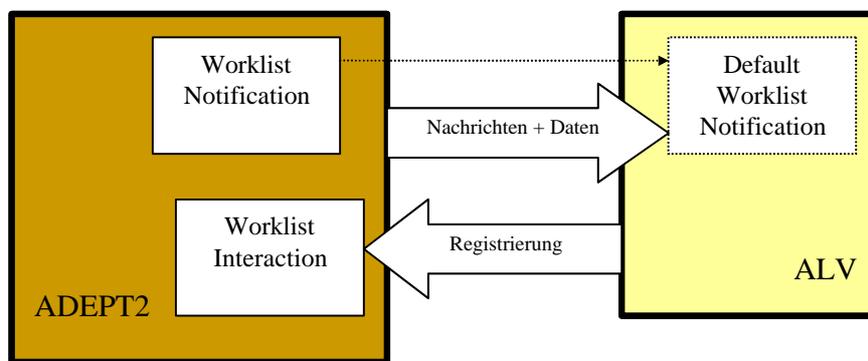


Abbildung 7-9: Interaktion zwischen ADEPT2 und der ALV

Übermittlung einer neuen Aktivität an die ALV

Eine entsprechende Methode (`addActivity`) ist in `WorklistNotification` hinterlegt. Sie wird vom WfMS mit den zur Verfügung stehenden Eingabeparametern aufgerufen und in der ALV ausgeführt. Die weitere Verarbeitung folgt dann in der Arbeitslistenverwaltung. Die Aktivität muss alle für die ALV wichtigen Attribute mitbringen, wie z. B. Prioritäten, Zeitvorgaben, Beschreibungen und eben auch die Bearbeiterformel.

Die Erstellung des `WorkItems` aus den Informationen des Prozessschrittes wird nicht vom WfMS angeboten und muss daher von der ALV vorgenommen werden.

Kompatibilität von Aktivitätszuständen zwischen ALV und WfMS

Das Zustandsmodell für Aktivitäten aus dem ADEPT2 – eine vereinfachte Darstellung zeigt die folgende Abbildung – enthält eine Reihe von Aktivitäten, die zu unterscheiden zwar für den Ablauf eines Prozesses wichtig sind, in der Arbeitslistenverwaltung aber keine Rolle spielen und daher abstrahiert werden können. Dabei werden sie auf die kleinstmögliche Menge reduziert.

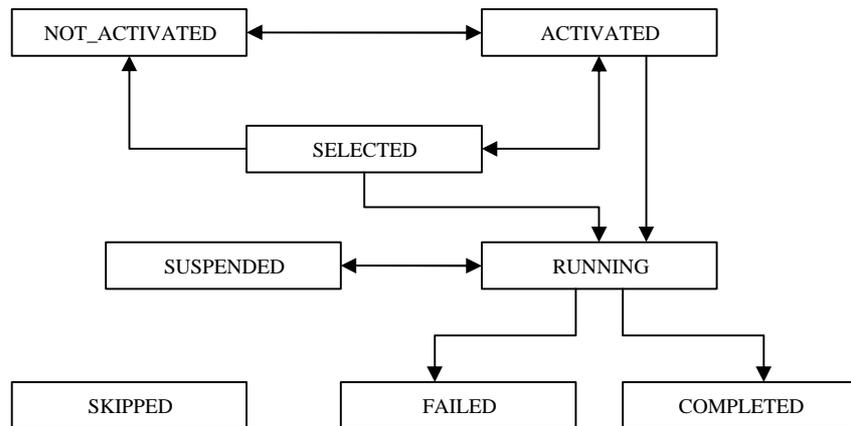


Abbildung 7-10: Die Zustände einer ADEPT2-Aktivität

Wird eine Aktivität in den Zustand `ACTIVATED` versetzt, wird sie vom WfMS an die Arbeitslistenverwaltung übergeben. Dort kann sie vom Klienten reserviert und wieder freigegeben (`SELECTED`), gestartet (`RUNNING`) und abgebrochen bzw. wieder aufgenommen werden (`SUSPENDED`). Wird eine Aktivität beendet, egal ob erfolgreich (`COMPLETED`) oder nicht (`FAILED`), zurückgesetzt (`NOT_ACTIVATED`) oder übersprungen (`SKIPPED`) muss das WfMS eine Nachricht an die ALV senden, damit diese die Aktivität aus ihrem System entfernen und die betroffenen Arbeitslisten bereinigen kann. Daraus lässt sich folgendes Zustandsmodell für die `WorkItems` der Arbeitslistenverwaltung modellieren. [Reic00]

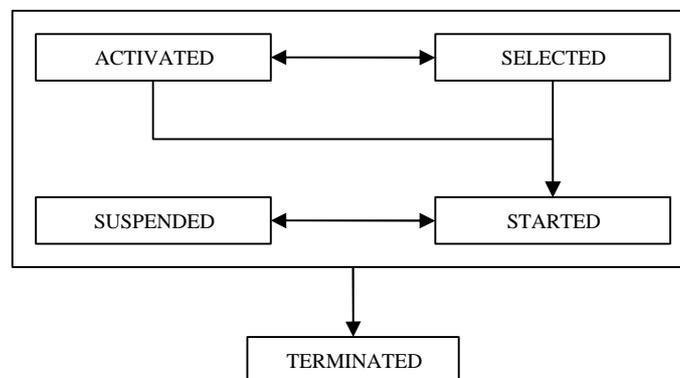


Abbildung 7-11: Zustände eines `WorkItems`

Die Schnittstelle `WorklistNotification` enthält eine Methode (`updateActivityState`), über die Zustandsänderungen einer Aktivität an die ALV übergeben werden können. Dafür müssen beide Zustandsmodelle in der ALV hinterlegt und vor der Übergabe oder Weiterverarbeitung miteinander abgeglichen werden.

Änderung von Aktivitätszuständen

Der Execution Manager von ADEPT2 bietet eine Schnittstelle (`ActivityStarting`), die es einem Klienten ermöglicht, Aktivitäten ohne Umwege über die ALV zu reservieren, zu starten, zu unterbrechen und zu beenden. Dafür muss vom Klienten die Referenz auf die betroffene Aktivität übergeben werden. Die ALV wird wiederum über `WorklistNotification` über diese Aktionen informiert.

Übermittlung geänderter Prozessschritte

ADEPT2 ist eines der wenigen WfMS, welches die Änderungen von Prozessschritten zur Laufzeit erlaubt und unterstützt.

Eine entsprechende Methode zur Übermittlung geänderter Prozessschritte muss in `WorklistNotification` hinterlegt und vom WfMS aufgerufen werden. ADEPT2 bietet jedoch bisher nur Methoden für die Übertragung geänderter Bearbeiterformeln (`reassignActivity`) und geänderter Zustände (`updateActivityState`).

Zugriff auf das Organisationsmodell

Für die Authentifizierung von Klienten und die Auflösung von Bearbeiterformeln stellt das Organisationsmodell von ADEPT2 zwei Schnittstellen zur Verfügung: `Authentication` und `PolicyResolution`. `Authentication` enthält die Methode zum authentifizieren eines Bearbeiters (`authenticate`) und über `PolicyResolution` kann eine gegebene Bearbeiterformel zu einer Menge von Bearbeitern inklusive ihren Stellen aufgelöst werden (`resolvePolicy`). [Berr05]

7.2.3 MQ Workflow

MQ Workflow ist ein Workflow-System des Herstellers IBM und Teil seiner WebSphere Produktlinie. [MQWf] Zur Untersuchung der Schnittstelle wurde der Programming Guide für die Version 3.6 verwendet. Er beschreibt die Verwendung der MQ Workflow Runtime API und enthält sowohl die Konzepte als auch die Aufrufmöglichkeiten der einzelnen, öffentlich zugänglichen Schnittstellen. [MQWF05]

Um die Schnittstellen von MQ Workflow benutzen zu können, muss zuerst eine Verbindung zwischen der ALV und dem Execution Server des WfMS aufgebaut werden. Dazu besorgt sich die ALV ein `ExecutionService`-Objekt durch Konstruktoraufruf, Zuweisung oder Lokalisierung und baut somit eine Sitzung zum dem Server auf. Ab dann kann die Anfrage nach Objekten stattfinden. Die folgende Abbildung zeigt den für die ALV relevanten Ausschnitt aus dem Datenmodell von MQ Workflow.

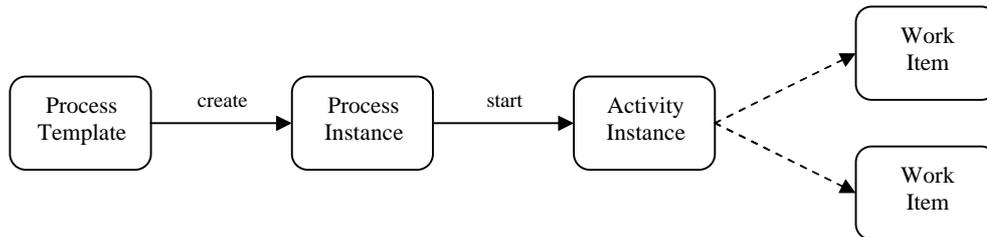


Abbildung 7-12: Teilausschnitt aus dem Datenmodell von MQ Workflow [MQWF05]

Der zeitliche Ablauf sieht so aus: die Prozessinstanz (`ProcessInstance`) startet einen Prozessschritt (`ActivityInstance`), der wiederum die Arbeitslistenaktivität (`WorkItem`) erzeugt.

Übermittlung einer neuen Aktivität an die ALV

`WorkItems` innerhalb von MQ Workflow sind Aktivitäten, die einem bestimmten Bearbeiter zugewiesen wurden, d. h. eine Aktivität mit 20 möglichen Bearbeitern resultiert in 20 `WorkItems`, die von dieser Aktivität abgeleitet werden. Diese Ableitung obliegt dem Workflow Server. `ActivityInstances` sind nicht ausführbar, d. h. um eine Aktivität abzuarbeiten muss das zugehörige `WorkItem` gestartet werden.

Es gibt mehrere Möglichkeiten über den `ExecutionService` Zugriff auf `WorkItems` zu erlangen. Mit `ExecutionService.createWorklist()` kann eine neue persistente Arbeitsliste auf dem Workflow Server angelegt werden, während mit `ExecutionService.queryWorklists()` alle Arbeitslisten eines bestimmten Bearbeiters angefordert werden. Ausgehend von einer vorliegenden Arbeitsliste ist es möglich, über `Worklist.queryWorkItems()` alle Aktivitäten einer Arbeitsliste zu extrahieren. Genauso kann man auch über `ExecutionService.queryWorkItems()` alle Aktivitäten eines vorgegebenen Bearbeiters erhalten.

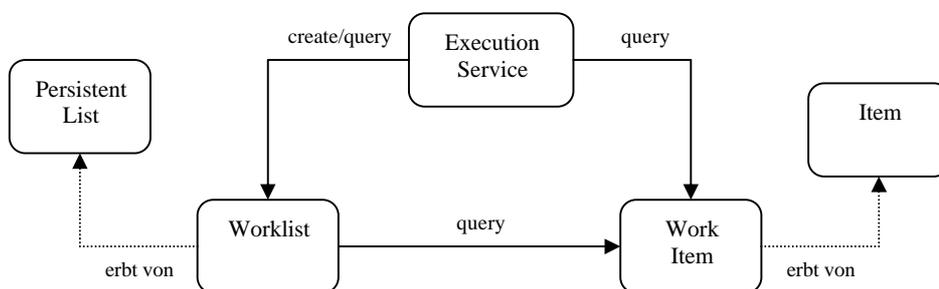


Abbildung 7-13: Zugriff auf `WorkItems` innerhalb von MQ Workflow [MQWF05]

Kompatibilität von Aktivitätszuständen zwischen ALV und WfMS

Das Zustandsmodell eines `WorkItems` in MQ Workflow besteht aus 14 verschiedenen Zuständen, deren Zustandsübergänge zusätzlich noch vom Zustand der zugehörigen Prozessinstanz abhängig sind. Eine reduzierte Übersicht über die für die ALV relevanten Zustände zeigt die folgende Abbildung.

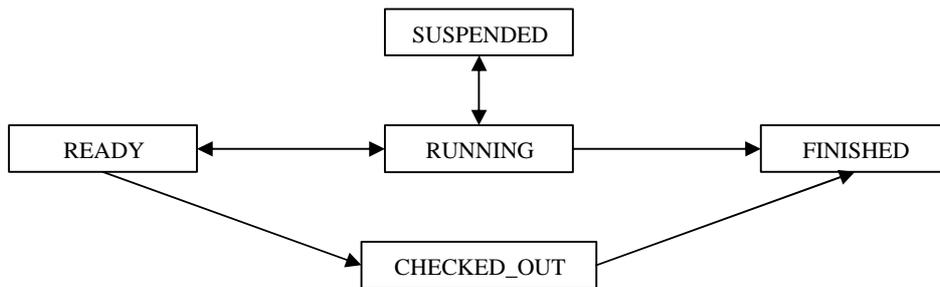


Abbildung 7-14: Zustandsmodell eines `WorkItems` in MQ Workflow [MQWF05]

Die für die Arbeitslistenverwaltung wichtigen Zustände beschränken sich auf `READY`, `RUNNING`, `SUSPENDED` und `FINISHED`. Die anderen Zustände sind entweder Variationen von 'beendet' oder haben mit der Arbeitslistenverwaltung direkt nichts zu tun. Eine Ausnahme bildet der Zustand `CHECKED_OUT`. Er beschreibt ein `WorkItem`, das zur manuellen oder externen Bearbeitung aus der ALV abgemeldet oder nach Beendigung wieder angemeldet wird. Diese fünf Zustände werden als einziges innerhalb der ALV unterschieden und abgesehen von `CHECKED_OUT` lassen sie sich gut mit dem eigenen Zustandsmodell der Arbeitslistenverwaltung verbinden.

Änderung von Aktivitätszuständen

Auf einem `WorkItem` können Operationen aufgerufen werden, die den Zustand des `WorkItems` ändern. Für den relevanten Ausschnitt des Zustandsmodells sind das folgende Methoden:

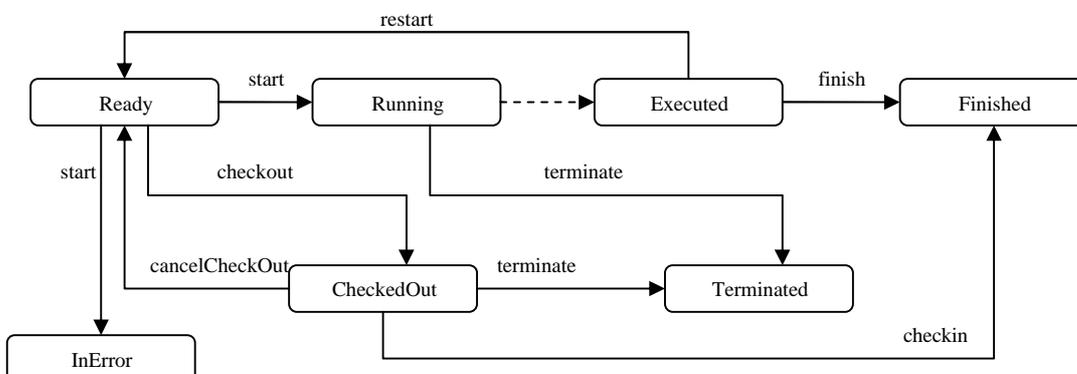


Abbildung 7-15: Zustandsänderungen in MQ Workflow [MQWF05]

Operationen, die auf einem `WorkItem` ausgeführt werden, haben auch Auswirkungen auf die anderen `WorkItems`, die von der gleichen Aktivität abgeleitet wurden. Dabei müssen die Zugriffe auf die einzelnen `WorkItems` untereinander abgeglichen werden, damit z. B. nicht zwei `WorkItems` der gleichen Aktivität zur gleichen Zeit von ihren Bearbeitern gestartet werden. Die Änderungen müssen vom Klienten direkt an das WfMS übermittelt werden. Dafür wird für jeden Klienten eine Sitzung mit dem Execution Server gestartet.

Übermittlung geänderter Prozessschritte

MQ Workflow ist eines der WfMS, die eine dynamische Änderung von Prozessschritten noch nicht unterstützen.

Zugriff auf das Organisationsmodell

Die Staff Administration API stellt Methoden bereit, um Anfragen und Änderungen auf von Organisationsdaten wie z. B. Mitarbeiter, Rollen oder Stellen auszuführen.

Für die Authentifizierung von Klienten kann die Schnittstelle `AuthenticationExit` verwendet werden. Durch sie kann man unabhängig von MQWorkflow einen anderen Dienst zur Authentifizierung, z. B. eine eigene Datenbank oder Datei verwenden.

7.2.4 TIBCO iProcess Engine

Die TIBCO iProcess Engine (ehemals TIBCO Staffware iProcessEngine) ist eine Komponente der TIBCO iProcessSuite, einem umfangreichen Prozess-Management-System, und ist zuständig für die Ausführung von Transaktionen und Geschäftsprozessen. [TIBCO] Für die Untersuchung der Schnittstellen wurden der TIBCO iProcess Object Programmer's Guide [TiPO05] und der TIBCO iProcess Server Objects Programmer's Guide [TiPSO05], jeweils in der Version 10.3 verwendet.

Die Schnittstellen der TIBCO iProcess Engine sind in so genannten Server Objects organisiert. Diese werden auf der Serverseite (im WfMS) gespeichert und können vom Klienten (in dem Fall der ALV) aufgerufen werden, um auf Daten zuzugreifen.

Die Server Objects sind in verschiedene Objekttypen eingeteilt, die beiden wichtigsten sind die `ServerObjects` selbst und die `ValueObjects`. `ServerObjects` enthalten selbst keine Daten und dienen nur dazu Methoden aus den `ValueObjects` abzurufen.

Die folgende Abbildung zeigt den für die ALV relevanten Ausschnitt aus dem Datenmodell der TIBCO iProcess Engine.

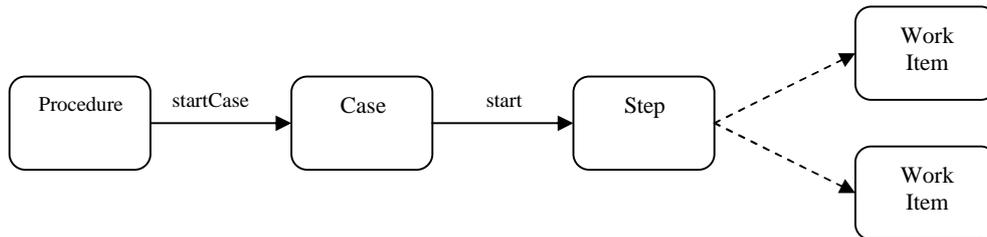


Abbildung 7-16: Teilausschnitt aus dem Datenmodell der TIBCO iProcess Engine [TiPO05]

Übermittlung einer neuen Aktivität an die ALV

In der TIBCO iProcess Engine bezieht sich ein `WorkItem` auf einen bestimmten Benutzer. Das bedeutet, wenn eine Aktivität 20 zugeordnete Benutzer hat, dann werden 20 `WorkItems` erstellt. Diese werden für den Benutzer in Arbeitslisten gruppiert. Jeder Benutzer kann mehrere Arbeitslisten haben, darunter seine persönliche sowie Arbeitslisten für Gruppen, Test und administrative Aufgaben.

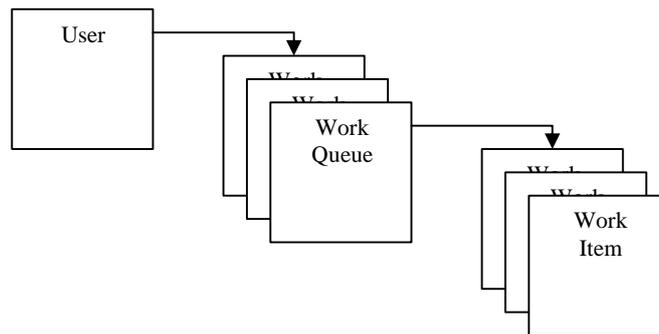


Abbildung 7-17: Die Zuordnung von `WorkItems` zu Benutzern [TiPO05]

`WorkItems` können ausgehend von einer gegebenen Arbeitsliste (`sWorkQ.getWorkItems`) oder einem Benutzer (`sUser.getWorkItems`) über spezielle Server Objects bezogen werden.

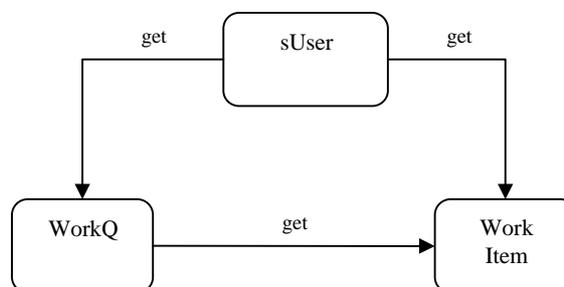


Abbildung 7-18: Zugriff auf `WorkItems` in der TIBCO iProcess Engine [TiPSO05]

Kompatibilität von Aktivitätszuständen zwischen ALV und WfMS

Die TIBCO Dokumentation stellt keine Übersicht zum Zustandsmodell eines `WorkItem`s zur Verfügung. Sie macht jedoch Angaben dazu, welche Operationen ein Benutzer ausführen kann, um eine Aktivität zu bearbeiten. Daraus lässt sich zumindest ein Teilausschnitt des Modells erstellen.

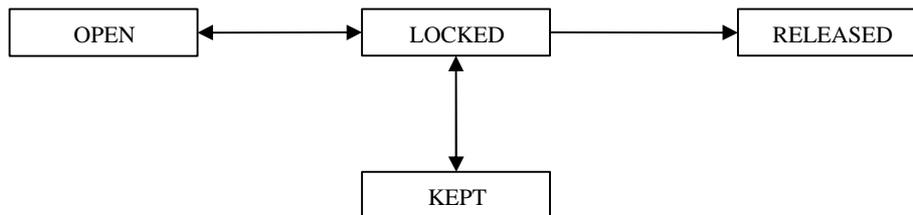


Abbildung 7-19: Zustandsmodell der `WorkItems` in der TIBCO iProcess Engine [TiPSO05]

Ein `WorkItem` befindet sich im Zustand `OPEN`, sobald der betreffende Prozessschritt aktiviert und das `WorkItem`-Objekt erstellt wird. Ein Bearbeiter kann das `WorkItem` nun zur Bearbeitung sperren (`LOCKED`) und bei Beendigung der Arbeit entweder behalten (`KEPT`) oder freigeben (`RELEASED`). Ein `WorkItem`, welches behalten wird, verbleibt im Prozessschritt und damit auch in der Arbeitsliste und kann von anderen Bearbeitern vervollständigt werden, bis es komplett abgearbeitet und freigegeben wird.

Änderung von Aktivitätszuständen

Über das Server Object `sWorkItem` können verschiedene Operationen auf einem `WorkItem` ausgeführt werden. Dabei kommuniziert der Klient direkt mit dem WfMS.

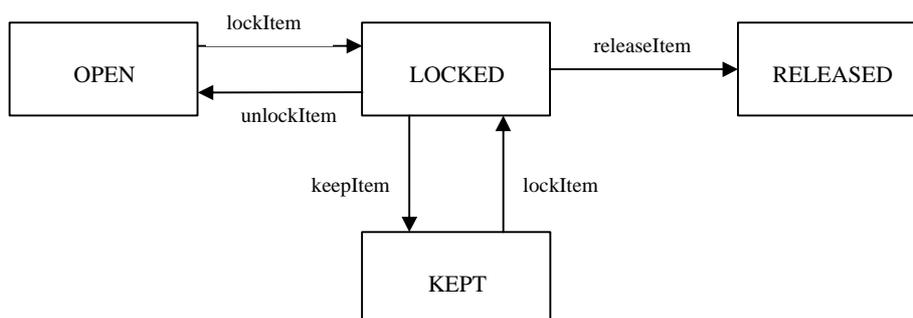


Abbildung 7-20: Zustandsänderungen bei TIBCO iProcessEngine [TiPSO05]

Änderungsoperationen, die auf einem `WorkItem` ausgeführt werden, haben auch Auswirkungen auf die anderen `WorkItems` der gleichen Aktivität. Dabei müssen die Zugriffe auf die einzelnen `WorkItems` untereinander abgeglichen werden, damit z. B. nicht zwei gleiche `WorkItems` zur gleichen Zeit von ihren Bearbeitern gestartet werden.

Übermittlung geänderter Prozessschritte

Die TIBCO iProcess Engine unterstützt bisher noch keine dynamische Änderung von Prozessschritten.

Zugriff auf das Organisationsmodell

Sowohl `WorkItems` als auch `Steps` (Prozessschritte) haben Methoden, um die ihnen zugewiesenen Bearbeiter abzufragen. Dabei liefert das `WorkItem` nur den Besitzer zurück, während `Step` sämtliche für die Ausführung in Frage kommenden Bearbeiter zurückliefert.

Des Weiteren können über das `Server Object AddressUserRef` alle Nutzer ermittelt werden, die administrativen Zugriff auf einen Prozessschritt haben, einer bestimmten Gruppe angehören oder Zugriff auf eine bestimmte Arbeitsliste haben.

7.2.5 Vergleich der Systeme

Der Vergleich der drei WfMS hat gezeigt, dass verschiedene Workflow-Systeme eine unterschiedliche Struktur hinsichtlich der Schnittstellen und des Datenmodells aufweisen.

ADEPT2 erfüllt die meisten der in Abschnitt 7.2.1 gestellten Anforderungen. Es ermöglicht der ALV `WorkItems` selbstständig von Aktivitäten abzuleiten und für die einzelnen Bearbeiter zu referenzieren. Das Zustandsmodell ist vollständig kompatibel mit den auftretenden Zuständen eines `WorkItems`. Operationen auf den einzelnen `WorkItems`, wie z. B. das Starten einer Aktivität, können über das WfMS vorgenommen werden. ADEPT2 ist zudem das einzige der drei WfMS, welches dynamische Änderungen von Prozessschritten zulässt. Für den Zugriff auf das Organisationsmodell, vor allem zur Auflösung von Bearbeiterformeln, stellt ADEPT2 eine eigene Schnittstelle, den `OrgModelManager` zur Verfügung.

In MQ Workflow ist jedem Bearbeiter einer Aktivität ein eigenes `WorkItem` zugewiesen, welches vom Workflow-System selbst abgeleitet wird. Die ALV muss daher die übermittelten `WorkItems` verwalten, ohne Änderungen vornehmen zu können. Das Zustandsmodell von MQ Workflow ist umfangreich, kann jedoch so weit reduziert werden, dass es mit dem Modell der ALV kompatibel ist. Der Klient kann direkt mit dem WfMS kommunizieren, um Operationen auf `WorkItems` auszuführen. Für Zugriffe auf das Organisationsmodell werden zwei Schnittstellen bereitgestellt. Eine Änderung von Prozessschritten zur Laufzeit unterstützt MQ Workflow derzeit noch nicht.

In der TIBCO iProcess Engine sind – ähnlich wie bei MQ Workflow – die `WorkItems` einzelnen Bearbeitern zugeordnet. Sie werden ebenfalls vom Workflow-System erzeugt, ohne dass die ALV eingreifen kann. Das Zustandsmodell weist einige Unterschiede zu dem der ALV auf, wodurch eine Anpassung schwierig zu erreichen ist. Der Klient übermittelt Aktionen auf den einzelnen `WorkItems` direkt an das WfMS. Über verschiedene Schnittstellen können Informationen über Bearbeiter abgerufen werden. Auch die TIBCO iProcess Engine unterstützt noch keine dynamischen Änderungen von Prozessschritten.

Die Schnittstelle des WfMS ADEPT2 erfüllt die Anforderungen der ALV daher am besten. Im folgenden Abschnitt wird die Schnittstelle der ALV zu diesem WfMS entworfen und die Schnittstellen von ADEPT2 dabei geringfügig entsprechend angepasst.

7.2.6 Entwurf einer Schnittstelle zu ADEPT2

Aus den vorangegangenen Erkenntnissen lässt sich nun die Verbindung zwischen Arbeitslistenverwaltung und ADEPT2 spezifizieren. Das Konzept der zwei Schnittstellen – `WorklistInteraction` für Nachrichtenübertragung zum WfMS und `WorklistNotification` für die Benachrichtigung der ALV – wird dabei übernommen und die notwendigen Änderungen eingefügt.

Die folgende Abbildung gibt einen Überblick über die verschiedenen Schnittstellen zwischen der ALV und dem WfMS. Um die Anforderungen der Arbeitslistenverwaltung zu erfüllen genügen drei Schnittstellen. Über `WorklistInteraction` kann sich die ALV beim WfMS registrieren und kann dadurch Nachrichten und Daten zur Weiterverarbeitung empfangen. Die Übermittlung neuer oder geänderter Prozessschritte wird durch `WorklistNotification` bzw. `WorklistRequest` sichergestellt. Den Zugriff auf das Organisationsmodell des WfMS (`OrgModelManager`) erhält die Arbeitslistenverwaltung über eine eigene Schnittstelle (`OrgModelConnection`).

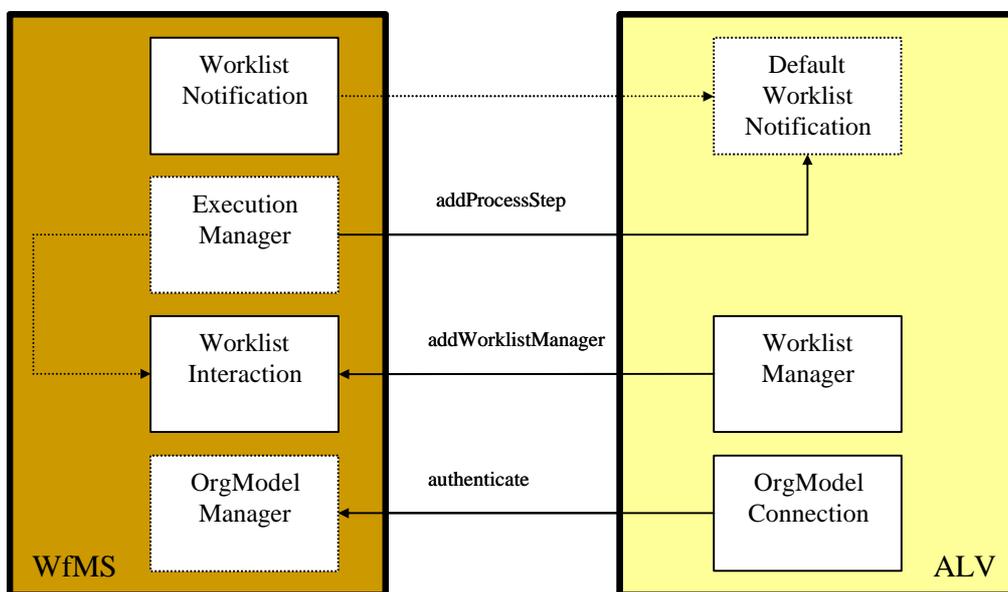


Abbildung 7-21: Zusammenarbeit von WfMS und ALV

WorklistInteraction: Enthält Operationen, die es der ALV erlauben, sich beim WfMS zu registrieren. Erst danach können Daten über `WorklistNotification` empfangen werden. Interface und Implementierung liegen beim WfMS.

- `void addWorklistManager(WorklistNotification)`
Registriert eine Arbeitslistenverwaltung am WfMS.
- `void removeWorklistManager(WorklistNotification)`
Entfernt die Registrierung einer Arbeitslistenverwaltung am WfMS.
- `void requestActiveActivities(WorklistNotification)`
Sendet alle aktiven Prozessschritte an die Arbeitslistenverwaltung, indem wiederholt die Funktion `addActivity` aus `WorklistNotification` aufgerufen wird. Diese Funktion ist notwendig, wenn sich eine ALV neu am WfMS anmeldet und über den aktuellen Stand der Arbeit informiert werden will.

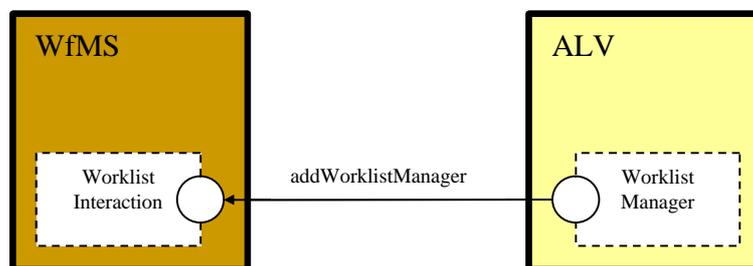


Abbildung 7-22: Die Schnittstelle WorklistInteraction

OrgModelConnection: Diese Schnittstelle verwendet den `OrgModelManager`, um Instanzen der `PolicyResolution`-Schnittstelle und der `Authenticate`-Schnittstelle zu erhalten. Darüber können dann Bearbeiterformeln aufgelöst und Bearbeiter authentifiziert werden.

- `boolean authenticate(user, pwd)`
Authentifiziert einen Benutzer über die gleichnamige Methode des `OrgModelManagers`.
- `List<Agent> resolvePolicy(staffAssignmentRule)`
Löst die gegebene Bearbeiterformel über das Organisationsmodell und die dort hinterlegte gleichnamige Methode auf.

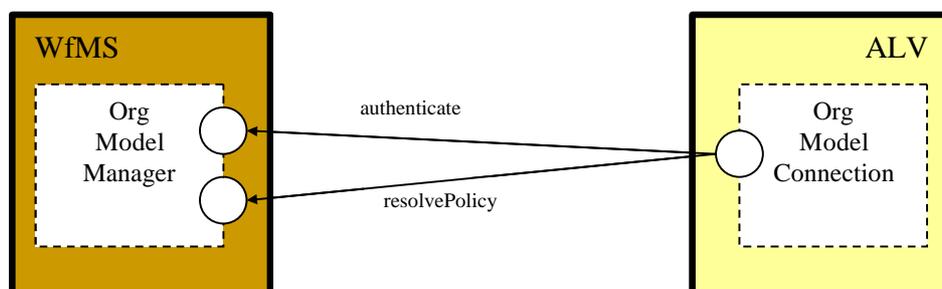


Abbildung 7-23: Die Schnittstelle OrgModelConnection

WorklistNotification: Die Schnittstelle wird vom WfMS bereitgestellt, um Informationen an die ALV zu senden, ohne erst auf eine entsprechende Anfrage warten zu müssen. Die ALV muss diese Schnittstelle selbst implementieren und schließlich über `WorklistInteraction` beim WfMS registrieren. Bei Eintritt eines entsprechenden Ereignisses (z. B. möchte das WfMS einen neuen Prozessschritt übermitteln), löst das WfMS bei allen registrierten `WorklistManagern` die entsprechende Methode aus und überträgt somit die zu verarbeitenden Informationen.

- `void addProcessStep(processStep, staffAssignmentRule)`
Übergibt einen neuen Prozessschritt samt Bearbeiterformel an die Arbeitslistenverwaltung.
- `void removeProcessStep(processStepID)`
Entfernt einen Prozessschritt aus der Arbeitslistenverwaltung, weil er auf Seiten des WfMS in einen der Zustände COMPLETED, FAILED oder SKIPPED gewechselt ist.
- `void reassignProcessStep(processStepID, staffAssignmentRule)`
Meldet der Arbeitslistenverwaltung, dass ein Prozessschritt eine neue Bearbeiterzuordnung erhält, die in der ALV umgesetzt werden muss.
- `void changeProcessStep(processStep)`
Überträgt einen geänderten Prozessschritt an die Arbeitslistenverwaltung. Diese muss die Änderungen ermitteln und auf das zugehörige `WorkItem` übertragen.

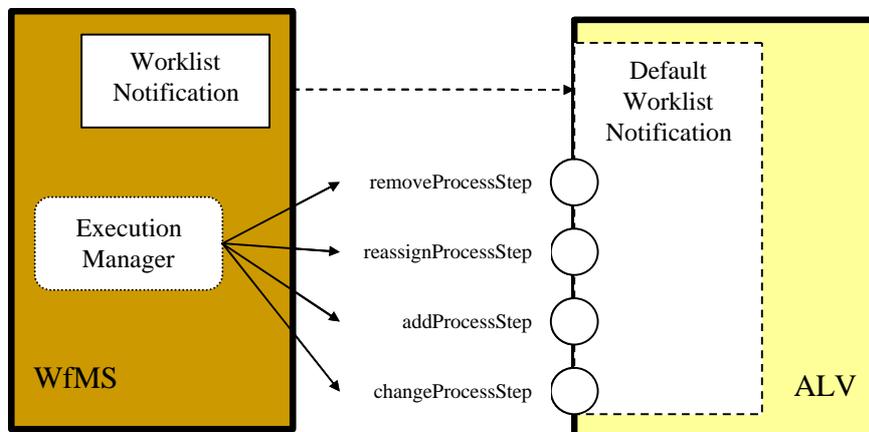


Abbildung 7-24: Die Schnittstelle `WorklistNotification`

WorklistRequest: Eine Alternative zu `WorklistNotification`, die Operationen enthält, um synchrone Anfragen über das Pull-Verfahren nach genau den oben genannten Informationen selbst zu stellen. Diese Schnittstelle muss vom WfMS zur Verfügung gestellt und von der ALV periodisch angefragt werden.

Beispielsweise stellt die ALV, um neue Aktivitäten zu erhalten, in vordefinierten Intervallen eine Anfrage an das WfMS und erhält als Antwort eine Liste aller neu aktivierten Aktivitäten. Diese Art der Kommunikation erzeugt allerdings unter Umständen zu viel Kommunikation oder führt dazu, dass die ALV auf weniger aktuellen Daten arbeitet.

- `List<ProcessStep> getNewProcessSteps()`
Stellt eine Anfrage nach neuen Prozessschritten. `List<ProcessStep>` muss in diesem Fall die Prozessschritte und ihre Bearbeiterformeln enthalten.
- `List<ProcessStep> getTerminatedProcessSteps()`
Stellt eine Anfrage nach allen Prozessschritten, die entfernt werden müssen, weil sie auf Seite des WfMS in einen der Zustände COMPLETED, FAILED oder SKIPPED gewechselt ist. `List<ProcessStep>` besteht in diesem Fall aus Aktivitätskennungen.
- `List<ProcessStep> getReassignedProcessSteps()`
Stellt eine Anfrage nach neuen Bearbeiterzuordnungen für Prozessschritte. `List<ProcessStep>` enthält Prozessschritte und die neuen Bearbeiterformeln.
- `List<ProcessStep> getChangedProcessStep()`
Stellt eine Anfrage nach geänderten Prozessschritten.

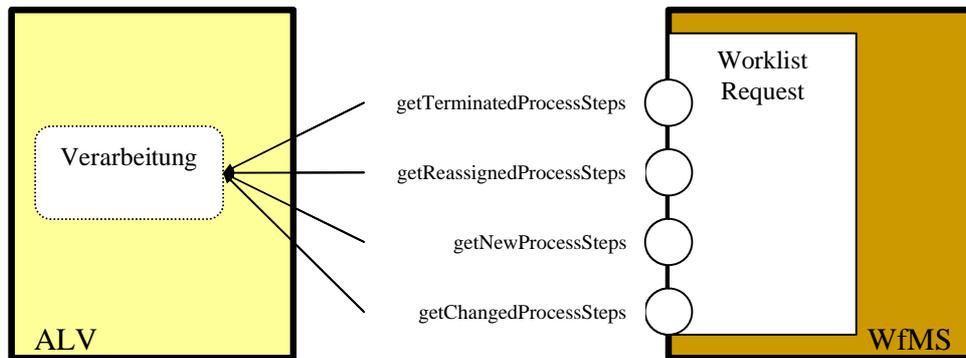


Abbildung 7-25: Die Schnittstelle `WorklistRequest`

7.3 Schnittstelle zum Arbeitslistenklienten

Ein zentraler Aspekt der Arbeitslistenverwaltung ist die Kommunikation mit dem Arbeitslistenklienten, der für die Darstellung der Arbeitslisten beim Benutzer zuständig ist. Eine wichtige Anforderung hierbei ist die Unabhängigkeit der Komponente. Es soll möglich sein über die Schnittstelle verschiedenste Klienten zu bedienen, sowohl einfache *HTML-Klienten* als auch so genannte *Rich Clients* – komplexere, intelligente Anwendungen, die selbst einen Teil der Verarbeitungslogik implementieren. Hierbei muss vor allem abgeschätzt werden, inwieweit die Arbeitslistenverwaltung Anwendungslogik für den Klienten bereitstellen muss und welche er selbst beinhaltet.

Um möglichst unabhängig zu sein von der Art des Arbeitslistenklienten werden mehrere Schnittstellen angeboten, deren Benutzung dem Klienten freigestellt wird. Nur die Basis-Schnittstelle (`WorklistClient`) mit den Grundfunktionen und die administrative Schnittstelle (`WorklistManager`) zur Registrierung der Klienten bei der ALV müssen verwendet werden.

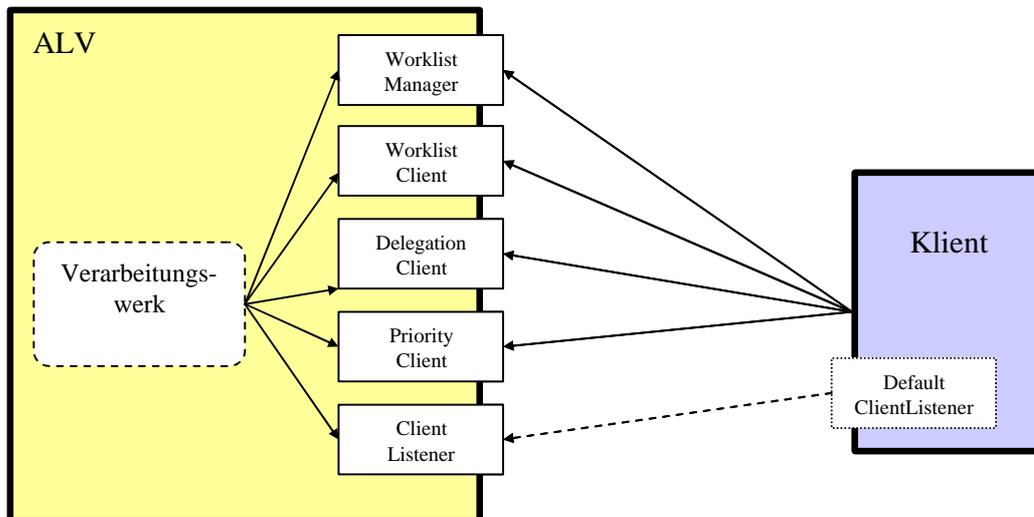


Abbildung 7-26: Schnittstellen der ALV zum Arbeitslistenklienten

WorklistManager: Die Schnittstelle für administrative Aufgaben, z. B. das An- und Abmelden von Bearbeitern, was ein Erstellen bzw. Löschen und eventuell Neuverteilungen von Aktivitäten nach sich zieht. Außerdem muss dem Klienten ermöglicht werden, seinen ClientListener bei der Arbeitslistenverwaltung anzumelden.

- `void loginUser(agent, orgPosition)`
Meldet einen neuen Bearbeiter an der Arbeitslistenverwaltung an und erstellt ihm eine Arbeitsliste bzw. weist ihm eine Gruppenarbeitsliste zu.
- `void logoutUser(agent, orgPosition, tempLogout)`
Meldet den gegebenen Nutzer an der angegebenen Stelle ab. Je nachdem, ob er sich kurzfristig oder längerfristig abmeldet (`tempLogout`) wird seine Arbeitsliste verworfen. Eventuell wird vorher noch eine Neuverteilung der Aktivitäten in dieser Liste veranlasst.
- `void addClientListener(clientListener)`
Registriert für jeden Arbeitslistenklienten einen Listener bei der Arbeitslistenverwaltung, der ihn über Push-Aktualisierungen informiert.
- `void removeClientListener(clientListener)`
Entfernt den Listener des Arbeitslistenklienten aus der Arbeitslistenverwaltung

WorklistClient: Diese Schnittstelle muss vom Klienten verwendet werden, um die ordnungsgemäße Zuweisung von Aktivitäten zu gewährleisten. Sie stellt die Grundfunktionalität, die jede Arbeitslistenverwaltung für den Klienten anbieten muss, zur Verfügung.

-
- `Update updateWorklist(worklist, agent)`
Sucht die zur angegebenen Arbeitsliste gehörende Aktualisierung für den Klienten und liefert den Inhalt als Liste von Aktivitäten an den Klienten aus.
 - `void rejectWorkItem(workItem, agent)`
Weist den Start einer Aktivität aufgrund unzureichender Ressourcen zurück. Die ALV hat dann die Aufgabe die Aktivität neu zu verteilen. Der angegebene Bearbeiter ist von dieser Neuverteilung jedoch ausgeschlossen

ClientListener: Die Schnittstelle muss vom Klienten implementiert und bei der ALV registriert werden für den Fall, dass eine Arbeitsliste mittels Push aktualisiert werden soll.

- `Update updateWorklist(worklist)`
Muss vom Klienten so implementiert werden, so dass die gleichnamige Funktion aus dem `WorklistClient` aufgerufen wird.

DelegationClient: Dies ist die Benutzerschnittstelle, über die ein Bearbeiter Aktivitäten delegieren oder eskalieren kann. Sie arbeitet weitestgehend mit dem `DelegationManager` zusammen, welcher die eigentliche Neuverteilung der Arbeitslisteneinträge vornimmt. Aus Sicherheitsgründen wird dem Klienten kein Zugriff auf den `DelegationManager` eingeräumt, da dort unter Umständen noch weitere, ALV-interne Methoden angeboten werden.

- `List<Agent> getDelegationRecipients(workItem, agent)`
Ruft die gleichnamige Funktion aus dem `DelegationManager` auf. `agent` bezeichnet hier den Delegationsauslöser.
- `void delegateWorkItem(workItem, agents)`
Ruft die gleichnamige Funktion aus dem `DelegationManager` auf. `agents` ist die Liste der Delegationsempfänger.
- `List<Agent> getEscalationRecipients(workItem, agent)`
Ruft die gleichnamige Funktion aus dem `DelegationManager` auf. `agent` ist der Eskalationsauslöser.
- `void escalateWorkItem(workItemID, agents)`
Ruft die gleichnamige Funktion aus dem `DelegationManager` auf. `agents` sind die Eskalationsempfänger.

PriorityClient: Über diese Schnittstelle wird dem Klienten die Möglichkeit gegeben, einen Teil der Funktionalität des `PriorityManagers` einzusetzen, um die Priorisierung einzelner Aktivitäten zu ändern.

- `void setPriorityForWorkItem(workItem, Constants.Priority)`
Ruft die gleichnamige Funktion aus dem `PriorityManager` auf.

7.4 Schnittstellen für die Verarbeitung von Arbeitslisten

In den Kapiteln 3 bis 6 wurden Konzepte für einzelne Aspekte einer erweiterten Funktionalität der Arbeitslistenverwaltung vorgelegt. Im Rahmen einer komponentenbasierten Entwicklung der ALV soll überprüft werden, ob diese Funktionalität auch über eigenständige Komponenten eingebunden und sinnvoll genutzt werden kann oder ob es günstiger ist, die Funktionalität direkt in der Arbeitslistenverwaltung zu implementieren. In den folgenden Abschnitten wird deshalb untersucht, welche Anforderungen an die Schnittstellen für Verteilungsverfahren (7.4.1), Delegation (7.4.2) und Priorisierung (7.4.3) gestellt werden. Aufbauend auf den Erkenntnissen wird jeweils ein entsprechender Entwurf ausgearbeitet.

7.4.1 Bearbeiterzuordnung

Aus den Erkenntnissen der Kapitel 3 (Bearbeiterzuordnung), 4 (Verteilungsverfahren) und 5 (Dynamische Neuverteilung) lassen sich für die Arbeitslistenverwaltung die folgenden Anforderungen ableiten:

- Die Arbeitslistenverwaltung muss die vom WfMS übergebene Bearbeiterformel einer Aktivität auflösen.
- Über das vorgegebene Verteilungsverfahren kann die aufgelöste Bearbeitermenge weiter eingeschränkt werden.
- Beim Abmelden von Bearbeitern muss angemessen mit einer Neuverteilung von Aktivitäten reagiert werden.

Es gibt zwei grundsätzliche Möglichkeiten die Bearbeiterzuordnung innerhalb der Arbeitslistenverwaltung einzubinden: durch komplette Eingliederung in die ALV oder durch zumindest teilweise Auslagerung der Funktionalität aus der ALV in eine externe Komponente.

Eine komplette Eingliederung der Bearbeiterzuordnung und der Verteilungsverfahren in die ALV bedeutet, dass alle Funktionen und Daten in die Arbeitslistenverwaltung integriert beziehungsweise dort hinterlegt werden müssen. Das hat den Vorteil, dass alle erforderlichen Daten für die ALV schnell zugreifbar sind und kein Kommunikationsaufwand und Datentransfer zu externen Komponenten besteht. Dadurch kann auch die Überwachung von Umgebungsbedingungen, die bei Änderung eine Neuverteilung nach sich ziehen, einfach von der ALV vorgenommen werden. Allerdings kann die ALV nicht oder nur mit sehr viel Aufwand auf die Anforderungen einzelner Verteilungsverfahren reagieren. Außerdem müssen dann innerhalb der ALV für jedes Verfahren spezifische Daten vorrätig gehalten werden, die aber bei Anwendung eines anderen Algorithmus nicht notwendig sind. Zuletzt ist es bei dieser von Schnittstelle auch nicht möglich ohne weiteres neue Verteilungsverfahren an die Arbeitslistenverwaltung anzubinden.

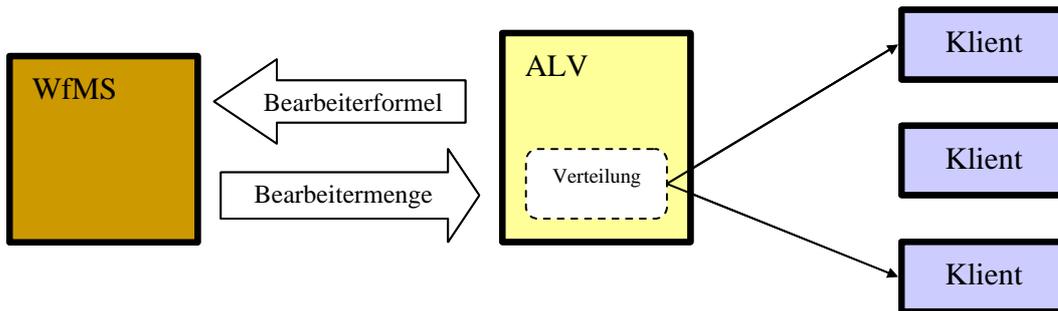


Abbildung 7-27: Die Mitarbeiterzuordnung eingegliedert in die ALV

Die zweite Möglichkeit sieht vor, dass ein konkretes Verteilungsverfahren in einer Art Dienstleistungskomponente untergebracht ist, die über eine Schnittstelle an die ALV angekoppelt wird. Dabei werden der Verteilungskomponente sämtliche Daten zur Verfügung gestellt, die diese nicht selbst mit angemessenem Aufwand hinterlegen und aktuell halten kann, darunter beispielsweise die Komplexität von Aktivitäten und Auslastungsgrade von Arbeitslisten. Dafür erhält die Verteilungskomponente Zugriff auf ausgewählte Datensätze des Datenmodells der ALV. Damit kann die Komponente die dynamischen Umgebungsbedingungen aller bisher untersuchten Verfahren berechnen und zur Festlegung oder Neuberechnung der Mitarbeiterzuordnung verwenden.

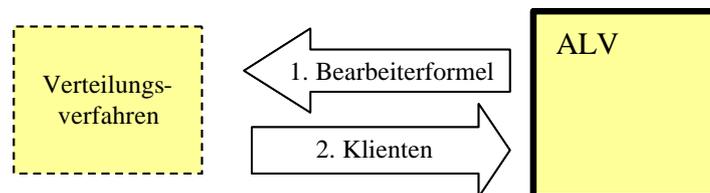


Abbildung 7-28: Die Mitarbeiterzuordnung ausgegliedert aus der ALV

Demnach sind eine wohldefinierte Schnittstelle der ALV und benutzerdefinierte Komponenten für die Verteilungsverfahren die günstigste Lösung für eine Umsetzung der Mitarbeiterzuordnung. Verfahrensspezifische Daten, die ohne viel Aufwand aktuell gehalten werden können, werden in der Komponente hinterlegt, während alle von sich ständig ändernden Umgebungsbedingungen abhängigen Daten über die ALV zur Verfügung gestellt werden.

Bei den dynamischen Neuverteilungen von Aktivitäten, die nach der Änderung von Umgebungsbedingungen auftreten können, wird im folgenden Entwurf nur die Neuverteilung nach dem Abmelden eines Bearbeiters in Betracht gezogen. Diese Methode ist notwendig um sicherzugehen, dass eine Aktivität auch dann noch abgearbeitet wird, wenn sich der einzige zugewiesene Bearbeiter vom System abmeldet. Andere Formen, z. B. die Überwachung und konsequente Anpassung der Auslastung werden nicht näher untersucht, da ihre Anwendung sehr zeit- und ressourcenintensiv ist und ihre Notwendigkeit bei einem stabil laufenden Verfahren nicht besteht.

Schnittstelle für Verteilungsverfahren

Um mehrere verschiedene Verteilungsverfahren an die Arbeitslistenverwaltung anbinden zu können, muss deren Schnittstelle möglichst generisch sein. Das heißt, es müssen Methoden angeboten werden, die als Ein- und Ausgabeparameter mit elementaren Datentypen arbeiten, die dann individuell interpretiert werden müssen.

Innerhalb der ALV teilt sich die Bearbeiterzuordnung auf zwei Schnittstellen auf: der `DistributionManager` ist die erste Anlaufstelle innerhalb der Arbeitslistenverwaltung. Er trifft alle nötigen Vorbereitungen und ruft dann die Verteilungskomponente auf. Diese verteilt die gegebene Aktivität auf Bearbeiter und gibt das Ergebnis in Form einer Liste zurück an den `DistributionManager`.

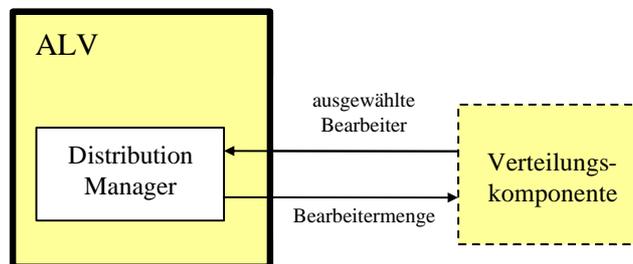


Abbildung 7-29: die Schnittstellen für Verteilungsverfahren

DistributionManager: Diese Schnittstelle enthält Methoden zur Erstverteilung einer Aktivität nach einem vorgegebenen Verfahren bzw. zur Neuverteilung von Aktivitäten bei Abmeldung eines Bearbeiters. Der `DistributionManager` selbst stellt nur die Schnittstelle zu einer Verteilungskomponente dar. Er wird von der ALV aufgerufen sobald eine neue Aktivität vom WfMS übergeben wird oder sich irgendwelche Umgebungsbedingungen ändern und leitet die Anfrage nach kurzer Vorarbeit an die aktive Verteilungskomponente weiter.

Allen Methoden gemeinsam ist, dass die ALV die Bearbeiterformel vor der Übergabe an die Verteilungskomponente auflösen und auf die angemeldeten Bearbeiter einschränken muss, denn die Komponente hat weder Zugriff auf das Organisationsmodell des WfMS noch darauf, welche Bearbeiter an- oder abgemeldet sind.

- `void distribute(workItem, staffAssignmentRule)`

Löst die angegebene Bearbeiterformel auf, ruft dann die gleichnamige Methode der Verteilungskomponente auf, um die Bearbeitermenge für eine Aktivität zu bestimmen und verteilt die Aktivität dann auf die Arbeitslisten der Bearbeiter in der Ergebnismenge.

- `void redistributeLogout(agent)`

Sucht die Arbeitsliste des angegebenen Bearbeiters und ruft für jede darin enthaltene Aktivität die Methode `distribute` (siehe oben) auf, um eine Neuverteilung der Arbeitsliste des Klienten nach dessen Logout zu veranlassen.

- `void redistributeReject(workItem,staffAssignmentRule,agent)`
Wird aufgerufen, nachdem ein Klient eine Aktivität aufgrund mangelnder Ressourcen zurückgewiesen hat. Löst dann die gegebene Bearbeiterformel auf und übermittelt die Bearbeitermenge (ohne agent, d. h. den Bearbeiter, der die Aktivität zurückgewiesen hat) an die Verteilungskomponente. Dort wird, wie bei einer normalen Erstverteilung die Methode `distribute` aufgerufen.

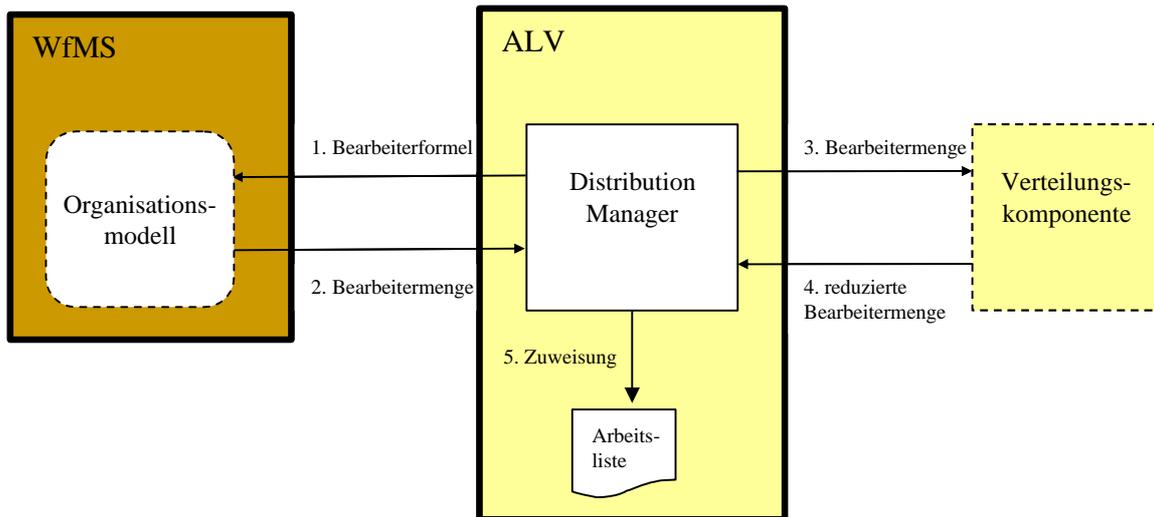


Abbildung 7-30: Ablauf einer Verteilung

DistributionRoundRobin: Die Verteilungskomponente für das Round Robin Verfahren. Wichtig ist, dass eine geordnete Mitarbeiterliste in der Komponente vorrätig gehalten wird, über die das Verfahren angewendet werden kann. Da die übergebene Bearbeitermenge nur angemeldete Bearbeiter enthält, genügt es eine Liste aller Mitarbeiter zu hinterlegen, ohne Anpassungen bei Abmeldung und Anmeldung von Bearbeitern vornehmen zu müssen.

- `List<Agent> distribute(List<Agent>)`
Übernimmt eine Liste von Bearbeitern und wendet das Verteilungsverfahren auf diese an. Die Rückgabe ist eine Bearbeiterliste mit nur einem Element (da der `DistributionManager` allgemein eine Liste von Bearbeitern als Rückgabewert erwartet).

DistributionLoadBalancing: Die Verteilungskomponente für das Load Balancing Verfahren. Um die Auslastung der Arbeitslisten der einzelnen Bearbeiter zu bestimmen, benötigt die Verteilungskomponente Zugriff auf das Datenmodell der Arbeitslistenverwaltung, um die Kapazität der Arbeitslisten und die Komplexität der Aktivitäten zu erhalten. Über die einzelnen Bearbeiter können ihre persönlichen Arbeitslisten abgefragt werden.

- `List<Agent> distribute(List<Agent>)`
Übernimmt eine Liste von Bearbeitern und wendet das Verteilungsverfahren auf diese an. Die Rückgabe ist eine Liste von einem oder mehreren Bearbeitern.

7.4.2 Delegation

Die Erkenntnisse aus der Untersuchung der Delegation (Abschnitt 5.2.1) stellen folgende Anforderungen an die Arbeitslistenverwaltung fest:

- Unterstützung automatischer und manueller Delegation
- zur Bestimmung der Delegationsempfänger werden Bearbeiterformeln verwendet
- die Delegationsstufe muss berechnet und hinterlegt werden

Automatische und manuelle Delegation sind sich in ihren Ausprägungen sehr ähnlich. Der einzige Unterschied besteht darin, dass automatische Delegationen auf bestimmte Ereignisse reagieren und von der ALV selbst ausgelöst werden, während manuelle Delegationen von einem Bearbeiter angestoßen werden.

Um zu verhindern, dass Aktivitäten zu häufig delegiert und damit an ihrer Abarbeitung gehindert werden, muss die ALV eine Delegationsstufe verwalten, die besagt, wie oft eine Aktivität bisher delegiert wurde. Über den Abgleich mit einer Delegationsgrenze kann überprüft werden, ob die Aktivität noch einmal delegiert werden darf.

Da sich die Empfängermengen bei Eskalation und Delegation ein und derselben Aktivität unterscheiden und diese innerhalb ihrer Lebenszeit sowohl eskaliert als auch delegiert werden können, muss hier bei den Operationen wieder zwischen diesen beiden Formen unterschieden werden, obwohl der Ablauf nahezu identisch ist.

Innerhalb der Arbeitslistenverwaltung sind zwei Schnittstellen für die Delegation vorgesehen: der `DelegationClient` und der `DelegationManager`, wobei die Methoden des Klienten keine eigene Logik haben, sondern selbst nur auf die Manager-Schnittstelle zurückgreifen. Die Lösung wurde gewählt, da automatische und manuelle Delegation gleich ablaufen, aber dem Klienten kein Zugriff auf interne Verwaltungsschnittstellen der ALV gewährt werden soll.

Eine Implementierung der Delegation als eigenständige Komponente ist im vorliegenden Fall nicht sinnvoll. Es findet keine aufwändige oder sich wiederholende Bearbeitung statt, mit der die Verlagerung in eine eigene Komponente gerechtfertigt werden könnte.

DelegationManager: Dies ist die Schnittstelle, über die Aktivitäten delegiert oder eskaliert werden. Sie enthält Methoden für die eigentliche Delegation und die Bestimmung der Empfängermengen. Die Bearbeiterformeln für die Empfänger können bei der Aktivität oder in der ALV hinterlegt sein.

- `List<Agent> getDelegationRecipients(workItem, agent)`

Ausgehend von einer hinterlegten, generischen Bearbeiterformel werden die potentiellen Empfänger der Delegation abhängig vom delegierenden Bearbeiter (`agent`) oder der zu delegierenden Aktivität (`workItem`) bestimmt. Das wird vor allem dazu verwendet, dem Delegierenden die mögliche Empfängermenge zur Auswahl anzubieten.

- `void delegateWorkItem(workItem, agents)`

Diese Funktion delegiert die angegebene Aktivität (`workItem`) an die angegebenen Bearbeiter (`agents`) und stellt sie ihnen in die Arbeitsliste. Aus den übrigen Arbeitslisten wird der Eintrag danach entfernt. Zum Schluss wird die Delegationsstufe geändert.

- `List<Agent> getEscalationRecipients(workItem, agent)`

Ermittelt ausgehend vom angegebenen Bearbeiter (`agent`) oder der Aktivität (`workItem`) die Empfängermenge der Eskalation.

- `void escalateWorkItem(workItem, agents)`

Eskaliert die betroffene Aktivität (`workItem`) an die angegebenen Bearbeiter (`agents`) und aktualisiert dabei die Arbeitslisten aller Beteiligten. Danach wird die Eskalationsstufe angepasst.

7.4.3 Priorisierung

Die Untersuchungen aus Kapitel 6 ergaben folgende Anforderungen an die ALV:

- die automatische Priorisierung verlangt nach dem Einsatz von Timern
- zur Unterstützung des Arbeitslistenklienten können Arbeitslisten vorsortiert werden
- Priorisierungsinformationen stammen sowohl aus dem WfMS als auch aus der ALV

Im Folgenden werden drei verschiedene Möglichkeiten diskutiert, wie man die Priorisierung in die Arbeitslistenverwaltung einbinden kann: durch komplette Eingliederung in die ALV, durch komplette Auslagerung in einen externen Dienst oder durch teilweise Auslagerung der Funktionalität in entsprechende Komponenten.

Die direkte Integration der Priorisierung in die Arbeitslistenverwaltung bedeutet, dass sämtliche Informationen und die gesamte Funktionalität an einer Stelle verwaltet werden. Die ALV übernimmt dabei die Überprüfung der Aktivität auf bereits bestehende Prioritäten und startet verschiedene Timer, die den zeitlichen Ablauf der Bearbeitung überwachen. Läuft während des Aufenthalts der Aktivität in der Arbeitslistenverwaltung einer der Timer ab, wird die Aktivität von der ALV priorisiert und der oder die Bearbeiter benachrichtigt. Eine manuelle Priorisierung wird über eine gegebene Benutzerschnittstelle ausgelöst.

Der Vorteil dabei ist, dass die Arbeitslistenverwaltung direkten Zugriff auf alle Funktionen und Daten hat, wodurch die Kommunikation und der Datenverkehr auf ein Minimum reduziert werden. Die ALV hat ihre eigene Datenbank und kann Priorisierungsinformationen selbst hinterlegen und individuell an Aktivitäten oder Umgebungsbedingungen anpassen. Nachteilig ist, dass die Arbeitslistenverwaltung dadurch sehr viel mehr Aufwand hat und das Einfügen neuer Priorisierungsdienste schwierig ist. Außerdem ist es möglich, dass ein Priorisierungsdienst gar nicht erwünscht ist und durch unnötige Funktionalität Kosten verursacht werden.

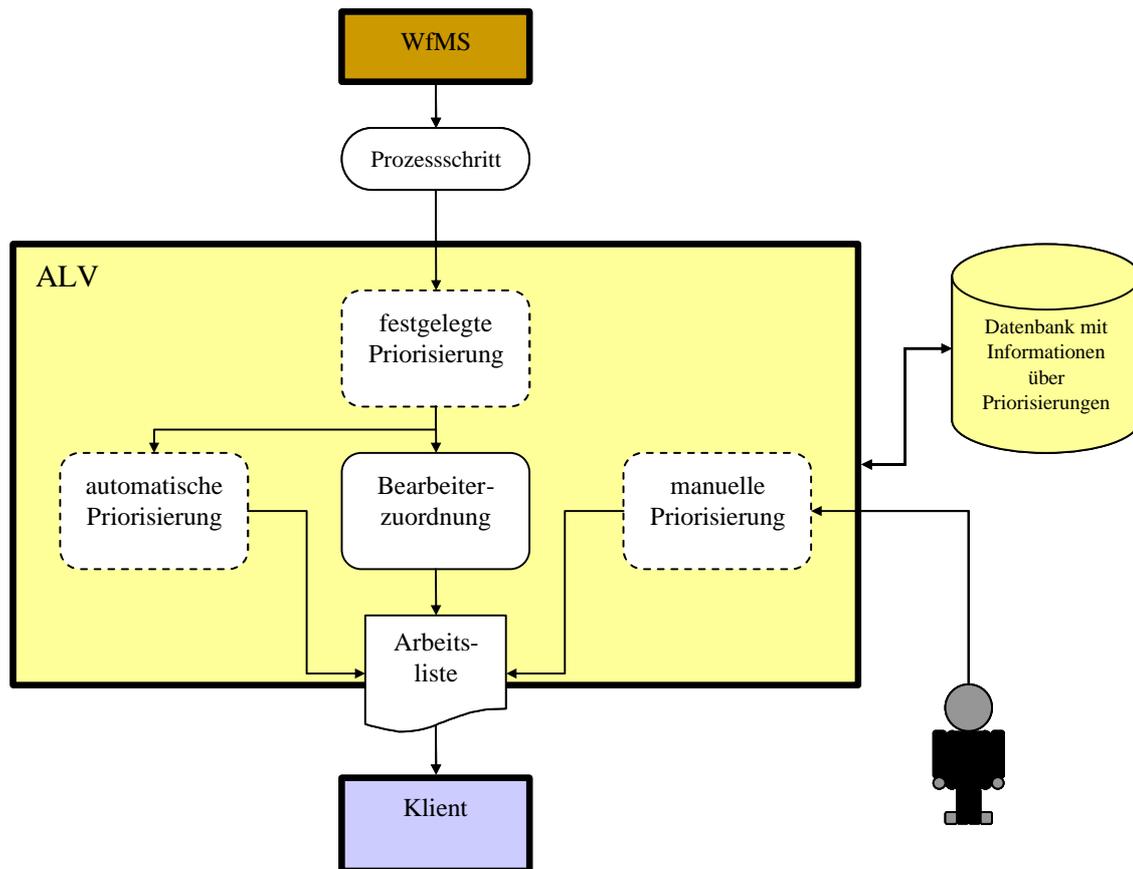


Abbildung 7-31: Priorisierung integriert in die ALV

Im Rahmen einer Komponentenarchitektur kann man die Priorisierung auch als eigenen, von der Arbeitslistenverwaltung unabhängigen Dienst implementieren. Dabei werden sämtliche Informationen über priorisierte Aktivitäten in der komponenteneigenen Datenbank hinterlegt. Die ALV fragt dann beim Eintreffen einer neuen Aktivität beim Priorisierungsdienst nach bereits existierenden Prioritäten. Im Anschluss daran werden wieder über die Priorisierungskomponente Timer für die Aktivität gestartet und überwacht. Läuft einer der Timer ab, wird die ALV informiert, welche die Nachricht an den Klienten weiterleitet. Die Benutzerschnittstelle für die manuelle Priorisierung läuft ebenfalls über den Priorisierungsdienst.

Diese Art der Umsetzung einer Priorisierung erfüllt die Anforderungen einer komponentenbasierten Entwicklung: sie ist weitgehend unabhängig von der Arbeitslistenverwaltung, entlastet diese und macht den Priorisierungsdienst optional, d. h. die ALV und der Klient können selbst entscheiden, in welchem Maß die den Dienst in Anspruch nehmen wollen. Genauso einfach kann neue Funktionalität in die Priorisierung integriert werden. Nachteilig ist, dass ein hohes Maß an Kommunikation zwischen Arbeitslistenverwaltung und Priorisierungskomponente nötig ist. Außerdem braucht der Klient neben der Verbindung zur ALV dann auch noch eine Verbindung zur Priorisierungskomponente.

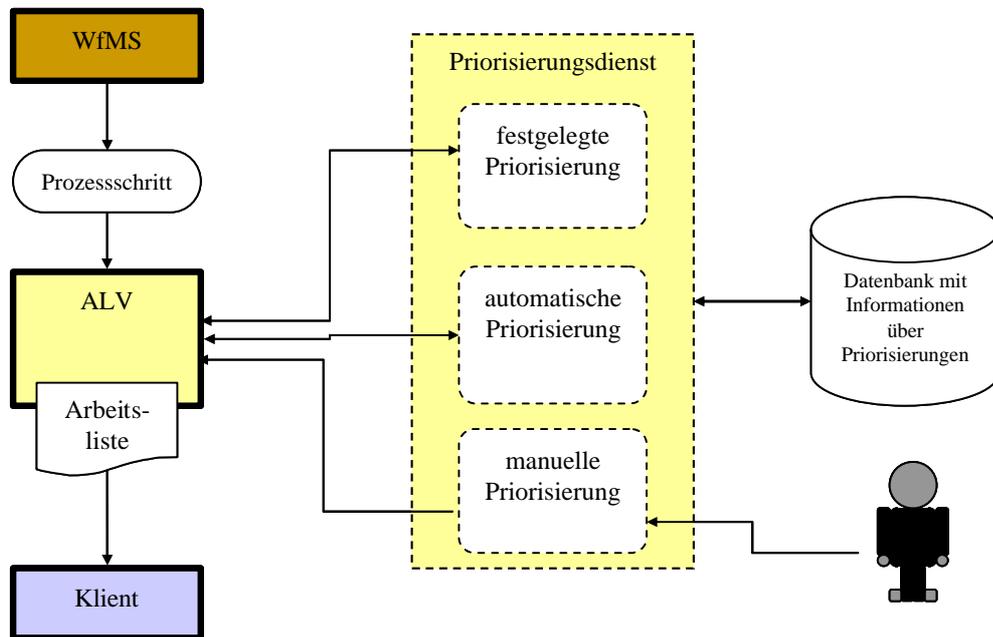


Abbildung 7-32: Priorisierung als eigene Komponente

Anstatt die Priorisierung gänzlich in eine eigene Komponente auszulagern, lohnt es sich abzuwägen, welche Aufgaben der Arbeitslistenverwaltung ohne großen Aufwand zugemutet werden können.

Ein Aspekt betrifft die Priorisierung von Aktivitäten wenn sie vom WfMS übergeben werden. Eventuell bestehende Prioritäten werden zusammen mit dem Prozessschritt übermittelt und können, wie auch andere Daten, gleich mit in die Aktivität übernommen werden. Das betrifft sämtliche Informationen, die Priorisierungen beeinflussen können und im Prozessmodell hinterlegt werden: vorgegebene Endtermine, die maximale Bearbeitungsdauer, Zeitfenster und die Priorität an sich. Die Übernahme dieser Informationen kann in der ALV ohne Aufwand vorgenommen werden.

Mithilfe der bereits existierenden Informationen können als nächstes die Timer für jede einzelne Aktivität gesetzt werden. Welche Arten von Timern zur Verfügung stehen und ob sie überhaupt eingesetzt werden kann individuell durch Verantwortliche entschieden werden und sollte nicht in der ALV festgelegt sein. Eine eigene Komponente für die automatische Priorisierung bringt den Vorteil, dass Timer nach eigenem Anwendungsbedarf eingesetzt und sogar selbst erstellt werden können.

Die Benutzerschnittstelle für die manuelle Priorisierung sollte ebenfalls optional sein, muss aber nicht unbedingt in einer eigenen Komponente umgesetzt werden, denn dadurch entsteht nur eine größere Vernetzung zwischen den Beteiligten. Viel besser ist es, dem Klienten über die Arbeitslistenverwaltung eine Schnittstelle zur Priorisierung einzuräumen. Das birgt für die ALV keine nennenswerte Aufwandserhöhung und hat den Vorteil, dass ein Klient sich nur einmal an einer Komponente anmelden muss und dort alle Funktionalität bekommt, die er benötigt.

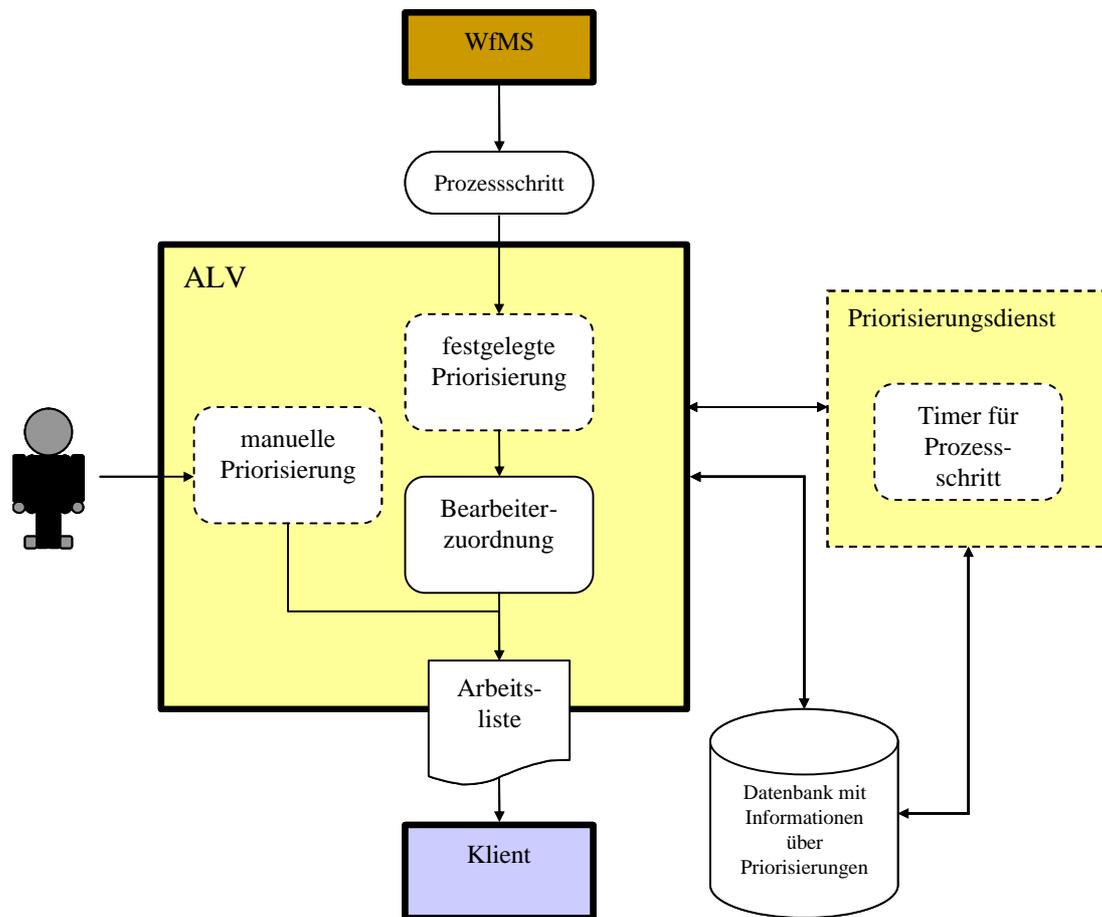


Abbildung 7-33: teilweise Auslagerung des Priorisierungsdienstes

Die Erkenntnis, die sich aus der Untersuchung der Priorisierungskomponente ziehen lässt ist, dass die Implementierung der Priorisierung teils in der Arbeitslistenverwaltung und teils als eigenständige Komponente die beste Lösung darstellt. Es belässt die weniger aufwendigen Priorisierungsfunktionen in der Arbeitslistenverwaltung und verlegt nur die Verwaltung der automatischen Priorisierung in eine eigene Komponente.

Die Schnittstellen der Priorisierung

Aus den oben durchgeführten Betrachtungen geht hervor, dass an der Priorisierung von Aktivitäten drei Schnittstellen beteiligt sind: der `PriorityClient` ermöglicht einem verantwortlichen Benutzer die manuelle Priorisierung und wurde bereits in der Benutzerschnittstelle in Abschnitt 7.3 diskutiert. Die anderen beiden Schnittstellen sind der `PriorityManager` und der `PriorityTimer`.

PriorityManager: Der `PriorityManager` ist die interne Priorisierungskomponente der Arbeitslistenverwaltung und sowohl Bezugspunkt für den `PriorityClient` als auch die dritte Schnittstelle, den `PriorityTimer`.

- `void setPriorityForWorkItem(workItem, Constants.Priority)`
Setzt die angegebene Priorität für die betroffene Aktivität.

PriorityTimer: Diese Schnittstelle gehört zu einer externen Komponente und verwaltet die auf den einzelnen Aktivitäten laufenden Timer. Die Methode `setPriority` ruft über den `PriorityManager` `setPriorityForWorkItem` auf, da der `PriorityTimer` selbst keinen Zugriff auf die Daten innerhalb der Arbeitslistenverwaltung besitzt.

- `void setPriority(workItem, Constants.Priority)`
Setzt für die angegebene Aktivität die Priorität nach Timerablauf.
- `void setTimerForWorkItem(workItemID)`
Setzt einen Timer für die angegebene Aktivität. Eine Aktivität kann mehrere Timer haben, auf die unterschiedlich reagiert wird. Die Informationen über die Zeitspanne der Timer werden in der Schnittstelle verwaltet sofern sie nicht über das Prozessmodell festgelegt und in der Aktivität hinterlegt wurden.

PriorityConstants: Aufbauend auf der Empfehlung aus Abschnitt 6.3 (Auswirkung der Priorisierung) werden für die Priorisierung folgende Konstanten gewählt und in der ALV implementiert:

CRITICAL: kritische Priorität
HIGH: hohe Priorität
NORMAL: keine Priorität
LOW: niedrige Priorität

7.5 Überblick über die Schnittstellen

Zum Abschluss des Schnittstellentwurfs werden hier noch einmal alle Schnittstellen und ihre Interaktionen untereinander im Überblick dargestellt.

Es wurde eine Schnittstelle zum Arbeitslistenklienten entworfen, die unabhängig ist dadurch, dass sie unterschiedliche Funktionalität anbietet. Es gibt eine Hauptschnittstelle für die grundlegende Versorgung des Klienten mit Arbeitslisten und Aktivitäten. Daneben gibt es eine Reihe erweiterter Funktionalitäten, deren Einbindung für den Arbeitslistenklienten optional ist.

Ein Teil der erweiterten Funktionalität wird über Schnittstellen aus externen Komponenten bezogen. Das ermöglicht es, diese Komponenten beliebig auszutauschen oder zu pflegen, ohne die Arbeitslistenverwaltung selbst zu beeinflussen.

Die Schnittstelle zum WfMS wurde auf der Grundlage untersuchter Anforderungen entworfen.

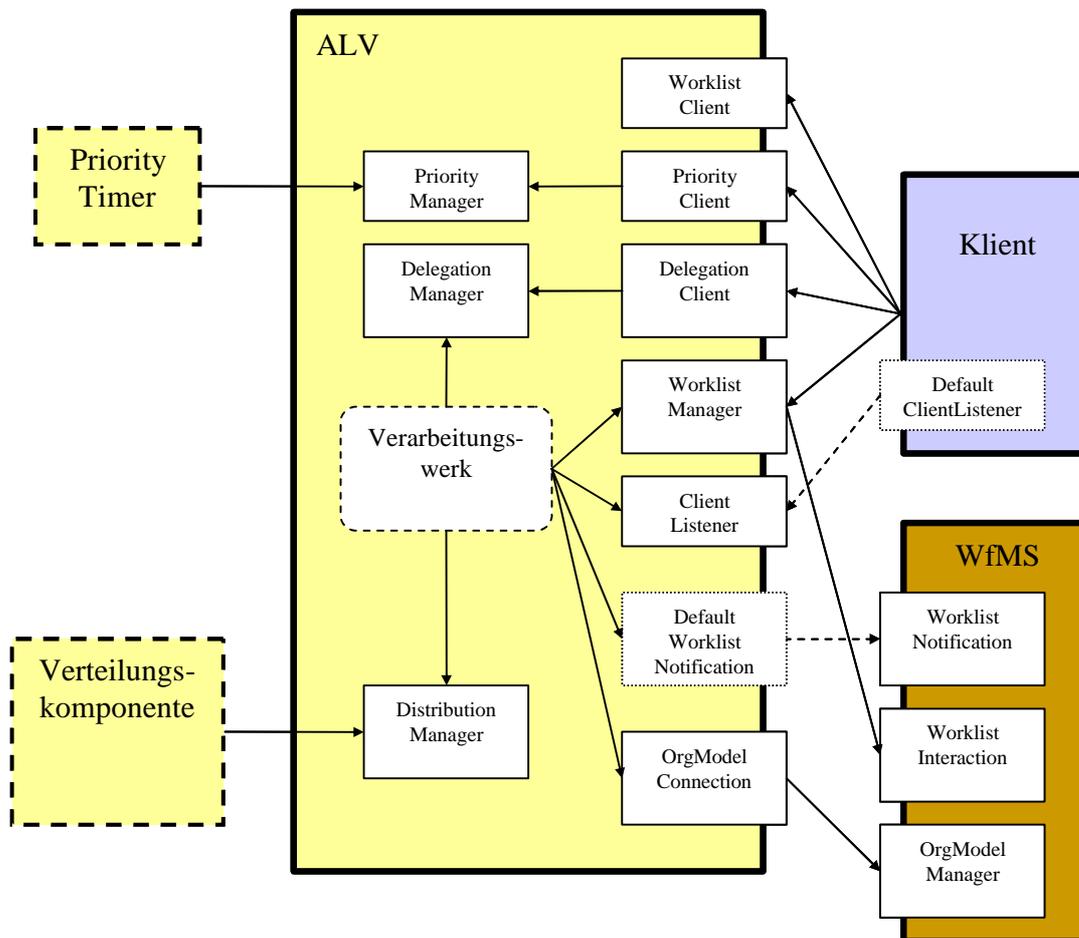


Abbildung 7-34: Überblick über die Schnittstellen der ALV

In Abschnitt 2.2 wurde das Workflow-Referenzmodell vorgestellt, ein Vorschlag der WfMC, wie ein Workflow-System standardmäßig aussehen sollte. Es sieht für die ALV zwei Schnittstellen vor: eine zum WfMS und eine zum Arbeitslistenklienten.

Die Schnittstelle der ALV zum WfMS weicht teilweise von der Vorgabe der WfMC ab. Der ALV werden eigene Zugriffsrechte auf dem Organisationsmodell eingeräumt. Außerdem werden Arbeitslisten nicht vom WfMS übergeben, sondern von der ALV zusammengestellt und verwaltet. Das Workflow-System übermittelt dabei nur die aktivierten Prozessschritte. Anwendungen, die mit den Aktivitäten verknüpft sind, werden nicht von der ALV gestartet, sondern über das WfMS.

Die Schnittstelle der ALV zum Arbeitslistenklienten genügt weitestgehend den Anforderungen der WfMC. Der Klient wird nur verwendet, um die Arbeitslisten beim Bearbeiter darzustellen. Dazu werden Informationen über die enthaltenen Aktivitäten von der ALV übermittelt.

8 Ausblick

Im Verlauf der Erstellung dieser Arbeit sind bei Untersuchungen und Diskussionen einige Aspekte aufgekommen, die aus Zeitgründen oder weil sie den Rahmen dieser Arbeit sprengen würden, nicht weiter erläutert wurden. Die folgenden Themenbereiche sind ein Ausblick darauf, über welches Potential die Arbeitslistenverwaltung noch verfügt und wie sie durch zusätzliche Aspekte gewinnbringend erweitert und ihr Anwendungsgebiet dadurch beachtlich ausgedehnt werden kann.

Groupware

Computer Supported Cooperative Work (CSCW) beschreibt das Prinzip rechnergestützter Gruppenarbeit. In vielen Tätigkeitsbereichen ist es heute üblich, Aufgaben interaktiv auszuführen. Das kann innerbetrieblich passieren, aber auch unternehmensübergreifend auf globaler Ebene. Diese Art der gemeinsamen Arbeit verlangt ein hohes Maß an Koordination, Kooperation und Kommunikation zwischen den beteiligten Mitarbeitern. Groupware unterstützt diese Aspekte auf softwaretechnischer Ebene. Sie besteht aus mehreren verschiedenen Anwendungen, die zusammen verwendet werden, um Gruppenarbeit zu organisieren und zu ermöglichen. Dazu gehören beispielsweise Email-Klienten, Mehrbenutzereditoren oder Content-Management-Systeme.

Ein WfMS übernimmt die Koordination innerhalb eines CSCW-Systems, d. h. die Planung von Terminen, die Zuteilung von Mitarbeitern zu einzelnen Aktivitäten sowie die Verknüpfung und das Starten von Anwendungen zur Ausführung der vorgegebenen Aufgaben.

Es gibt mehrere Möglichkeiten Groupware-Anwendungen in ein WfMS einzubinden. Die einfachste Methode besteht darin, sie als externe Applikationen mit einzelnen Prozessschritten zu verknüpfen und bei deren Start aufzurufen. Das hat den Vorteil, dass Groupware nach Bedarf eingesetzt werden kann, macht jedoch keine Aussagen darüber, welche Bearbeiter an der Zusammenarbeit teilnehmen sollen oder mit welchen anderen Aktivitäten der jeweilige Prozessschritt eventuell zusammenhängt. Eine zweite Möglichkeit wäre, das WfMS selbst zu einem CSCW-System umzugestalten und Groupware-Anwendungen zu einem Teil des Workflow-Systems zu machen. Hier ist es wiederum schwierig einzelne Aktivitäten durch spezielle externe Groupware-Anwendungen zu unterstützen, da deren Einbindung Kompatibilitätsprobleme und Redundanzen nach sich ziehen kann. [BSKa96]

Die Anforderungen, die sich durch die Verwendung von Groupware für WfMS ergeben, variieren abhängig davon, wie Groupware integriert wird und liegen mehrheitlich nicht im Rahmen der Arbeitslistenverwaltung. Es lassen sich jedoch einige allgemeine Anforderungen ableiten, die von der ALV erfüllt werden müssen, um zur Unterstützung von CSCW beizutragen. Erstens stellt eine Groupware-Aktivität besondere Bedingungen an die Bearbeiterzuordnung und das eingesetzte Verteilungsverfahren. So muss eine solche Aktivität naturgemäß an mehr als einen Bearbeiter verteilt oder eine entsprechende Gruppenarbeitsliste bereitgestellt werden. Die Aktivitäten müssen in den einzelnen Arbeitslisten besonders gekennzeichnet werden. Es ergeben sich Einschränkungen für Delegation und Eskalation der Arbeitslisteneinträge. Und nicht zuletzt

erfordert eine gemeinsame Bearbeitung auch ein gemeinsames Starten der Aktivität und sukzessive eine Übermittlung der Bearbeitermenge zur Weiterverarbeitung an die ALV.

Jobbildung

In manchen Situationen kann es hilfreich oder gar angebracht sein, zwei gleiche oder ähnliche Aktivitäten in einem gemeinsamen Arbeitsschritt (Job) zu erledigen. Zum Beispiel bietet es sich an, die Fertigung zweier Bauteile in einem Arbeitsgang durchzuführen, weil damit Betriebsstoffe gespart werden können oder die Kapazität der Maschine es hergibt. Oder man fasst die Bestellung mehrerer verschiedener Waren zu einem Auftrag zusammen, weil damit die Versandkosten günstiger werden. [Hein00, AAE96]

Diese Art der Abarbeitung von Aktivitäten stellt besondere Anforderungen an die Behandlung von Arbeitslisteneinträgen. Erst einmal müssen Aktivitäten daraufhin abgeglichen werden, ob sie gemeinsam bearbeitet werden können. Das beschränkt sich nicht nur auf gleiche Prozessschritte zweier Prozessinstanzen, sondern kann auch über die verknüpften Anwendungen (Webanwendung eines Zulieferers) oder bestimmte Prozessschrittvorlagen (Typ 'Bestellung') geschehen. Die passenden Aktivitäten müssen dann in den Arbeitslisten entsprechend kenntlich gemacht und dem Bearbeiter bei Auswahl einer einzelnen Aktivität zusätzlich angeboten werden. Auch die Möglichkeit Aktivitäten zu einem späteren Zeitpunkt zu einem bereits bestehenden Job hinzuzufügen, sollte überprüft und bei Bedarf angeboten werden. Eine Beschränkung der Anzahl der gemeinsam bearbeitbaren Aufgaben ist je nach Kapazität oder vorgegebener Höchstgrenze eventuell notwendig. Zuletzt muss auch durch eine geeignete Bearbeiterzuordnung gesichert sein, dass ähnliche Aktivitäten an dieselben Bearbeiter zugewiesen werden, um so eine gemeinsame Abarbeitung überhaupt erst zu ermöglichen.

Verteilungsverfahren bezogen auf eine spezielle Aktivität

Anstatt ein Verteilungsverfahren wie Round Robin oder das Load Balancing global in der Arbeitslistenverwaltung festzulegen und anzuwenden, kann man auch Situationen spezifizieren, in denen ein Verteilungsverfahren abhängig von einer speziellen Aktivität verwendet wird. Eine Möglichkeit ist beispielsweise, sehr aufwendige Arbeiten nicht auslastungsabhängig zu verteilen, sondern nach Round Robin, weil so jeder Bearbeiter wenigstens einmal eine komplexe Aufgabe zugewiesen bekommt.

Voraussetzung für diese Art der Verteilung ist, dass spezifiziert wird, bei welcher Aktivität und unter welchen Bedingungen welches Verfahren angewendet werden soll. Die ALV muss diese Bedingungen abgleichen und entsprechend reagieren. Eine solche Verteilung kann unter Umständen auch die Umgebungsbedingungen eines aktivierten globalen Verfahrens beeinflussen, so zum Beispiel die gleichmäßige Verteilung der Auslastung oder die Reihenfolge der Zuordnung bei Round Robin. Diese Situationen müssen entsprechend überwacht und die Umgebungsbedingungen angepasst werden.

Filterung

Um die Belastung der Arbeitsliste beim Klienten zu verringern und die Performanz zu steigern, ist es möglich ein Filterungsverfahren anzubieten, welches dem Klienten erlaubt, eigenhändig zu bestimmen, welche oder wie viele Aktivitäten ihm in seiner Arbeitsliste angeboten werden. So kann man beispielsweise eine Höchstgrenze von Arbeitslisteneinträgen definieren oder nur Aktivitäten eines bestimmten Typs anzeigen.

Die Aufgabe der Arbeitslistenverwaltung ist es dann, die Filterung auf die Arbeitsliste anzuwenden und die reduzierte Liste an den Klienten auszuliefern. Dabei entsteht ein großer Aufwand, da zwei unterschiedliche Listen vorrätig gehalten werden müssen. Jede Änderung der originalen Arbeitsliste kann eine Änderung der reduzierten Arbeitsliste nach sich ziehen oder nicht. Das Wechseln von einem Filter auf einen anderen ist zeitaufwändig und erfordert unter Umständen viel Kommunikation und Datenverkehr zwischen der ALV und dem Klienten. Vorgegebene Höchstgrenzen müssen ständig überwacht und der Inhalt einer Arbeitsliste bei Bedarf angepasst werden. Des Weiteren gibt es Aktivitäten, die eine Filterung umgehen können oder sogar müssen, z. B. Priorisierungen, Delegationen oder Eskalationen. Diese Aktivitäten müssen geeignet gekennzeichnet und vor allem bei einem Wechseln zwischen verschiedenen Filtern gesondert behandelt werden.

Suchfunktion

Bei großen unübersichtlichen Arbeitslisten kann es sinnvoll sein, dem Klienten eine Suchfunktion anzubieten, über die er einzelne Aktivitäten finden kann. Möglich wäre die Suche nach einer Aktivität mit bestimmten Eigenschaften, die nicht vom Filter erfasst werden oder die Suche nach einem Prozessschritt einer bestimmten Prozessinstanz.

Die Arbeitslistenverwaltung bestimmt dabei, welche Suchparameter für den Klienten zur Verfügung stehen. Außerdem muss ein Algorithmus gefunden werden, der die Suche effizient auf der Arbeitsliste ausführen kann und nach Bedarf auf Umgebungsbedingungen, wie z. B. gefilterte oder ungefilterte Arbeitslisten, Rücksicht nimmt.

9 Zusammenfassung

Ziel dieser Masterarbeit war die Konzeption und der Entwurf einer zentralen Komponente für die Arbeitslistenverwaltung. Neben dem Entwurf der Arbeitslistenverwaltung im Sinne einer komponentenbasierten Architektur, sollten auch erweiterte Funktionen und Aspekte in die Betrachtung mit einbezogen und diskutiert werden. Dazu gehörten die Bearbeiterzuordnung, Verteilungsverfahren, Priorisierung, Delegation und Vertreterregeln.

Als erstes wurde in Kapitel 3 die Bearbeiterzuordnung innerhalb der Arbeitslistenverwaltung untersucht. Einzelne Teilaspekte betrafen die Auflösung und Hinterlegung von Bearbeiterformeln, die Verwaltung von Arbeitslisten und Aktivitäten und die Kommunikation mit den Arbeitslistenklienten. Dabei wurde festgestellt, dass die Arbeitslistenverwaltung durch eine Schnittstelle zum Organisationsmodell des WfMS in der Lage ist, Bearbeiterformeln selbstständig aufzulösen. Das hat den Vorteil, dass die Einsatzmöglichkeiten der Bearbeiterformeln besser ausgeschöpft werden können, z. B. durch die allgemeine Definition und Platz sparende Hinterlegung von Delegationsempfängern. Als nächstes wurden die Aspekte der Verwaltung von Arbeitslisten diskutiert. Dabei wurde neben der effizienten Aktualisierung von Klientenarbeitslisten auch die Verteilung und Aufbewahrung von Aktivitäten untersucht. Die Ergebnisse flossen in den Entwurf eines Datenmodells für die ALV ein. Zum Schluss wurde die Kommunikation zwischen der ALV und dem Arbeitslistenklienten genauer betrachtet. Dabei wurde darlegt, welche Kommunikationsarten es gibt, welche Vor- und Nachteile sie haben und in welchen Situationen sie sinnvoll eingesetzt werden können.

Zur Steuerung und Reduktion der Auslastung der einzelnen Mitarbeiter ist die Anwendung von Verteilungsverfahren dringend notwendig. Zwei davon – Round Robin und die lastabhängige Verteilung – wurden in Kapitel 4 untersucht. Sie verbessern nicht nur die Übersichtlichkeit der Arbeitsliste beim Klienten, sondern bewirken zusätzlich, dass einzelne Aktivitäten durch die günstigere Zuordnung schneller abgearbeitet werden. Die Untersuchungen wurden in den Entwurf der Arbeitslistenverwaltung übernommen und resultierten in eigenständigen Komponenten für jedes Verteilungsverfahren. Damit können Verfahren individuell entwickelt und zur Laufzeit nach Bedarf eingesetzt werden.

In Kapitel 5 wurde die dynamische Neuverteilung von Aktivitäten diskutiert. Diese kann in bestimmten Situationen angemessen oder sogar notwendig sein. Beispielsweise kann durch eine Neuverteilung sichergestellt werden, dass die Arbeitslisteneinträge eines Bearbeiters, welcher sich vom System abmeldet, neu verteilt werden. Ein wichtiger Aspekt war die Neuverteilung einer Aktivität durch eine dynamische Änderung ihrer Bearbeiterzuordnung. Dabei wurden zwei Varianten der dynamischen Neuordnung unterschieden und miteinander verglichen: Delegationen und Vertreterregeln.

Im Bereich der Delegation wurde erörtert, wie die Empfänger einer Neuordnung abhängig vom delegierenden Bearbeiter oder der delegierten Aktivität selbst ermittelt werden können und wie sich eine Delegation auf die Arbeitslisten der betroffenen Bearbeiter auswirkt. Dabei wurde erkannt, dass sich die Spezifizierung von Delegationsempfängern sinnvoll durch den Einsatz von Bearbeiterformeln realisieren lässt. Diese können generisch hinterlegt und zum Zeitpunkt der Delegation an den Delegationsauslöser angepasst und über das Organisationsmodell aufgelöst

werden. Für den Entwurf wurde die Delegation als integrierte Funktionalität der Arbeitslistenverwaltung übernommen.

Bei der Untersuchung der Vertreterregelung wurde speziell darauf geachtet, wie und unter welchen Umständen sie eingesetzt und aufgelöst wird. Dabei wurde festgestellt, dass Vertreterregeln im Organisationsmodell hinterlegt sind und ihre Anwendung und Auflösung abseits der ALV während der Auflösung von Bearbeiterformeln geschieht. Für eine entsprechende Kennzeichnung der Vertretungen in den einzelnen Arbeitslisten ist es jedoch sinnvoll, die ALV über eine existierende Vertretung einer Aktivität zu informieren.

Zum Abschluss der Konzeption wurden in Kapitel 6 die Anforderungen an eine Priorisierung von Aktivitäten innerhalb der Arbeitslistenverwaltung diskutiert. Neben der manuellen Priorisierung lag besondere Aufmerksamkeit auf der automatischen Priorisierung nach Zeitvorgaben. Dazu muss die Arbeitslistenverwaltung für jede Aktivität die Zeitvorgaben überwachen und gegebenenfalls auf deren Überschreitung mit einer Priorisierung reagieren. Außerdem müssen die zeitlichen Informationen und Prioritäten bei der Aktivität verfügbar und zugreifbar sein. Für den Entwurf war es daher sinnvoll die Priorisierung in einer eigenen Komponente unterzubringen, die über eine dafür spezifizierte Schnittstelle von der Arbeitslistenverwaltung aus zugegriffen werden kann. Das hat den Vorteil, dass die Verwaltung der Überwachung jeder einzelnen Aktivität nicht aufwendig von der Arbeitslistenverwaltung vorgenommen werden muss und potentiell auch andere Priorisierungsverfahren verwendet werden können.

Kapitel 7 beschäftigte sich mit dem Entwurf der Komponente für die Arbeitslistenverwaltung. Dafür wurden Anforderungen an eine ideale Schnittstelle untersucht und festgelegt. Danach wurden die Schnittstellen von drei bekannten WfMS – ADEPT2, MQ Workflow und TIBCO iProcess Engine – daraufhin untersucht, welche Anforderungen und Voraussetzungen sie selbst für die Anbindung einer eigenen Arbeitslistenverwaltung mitbringen und ob sich diese mit den zuvor ermittelten Anforderungen vereinen lassen. Dabei wurde festgestellt, dass das WfMS ADEPT2 als einziges die Anforderungen ausreichend erfüllt. Dadurch konnte ein Entwurf für die Schnittstelle zu ADEPT2 vorgenommen werden.

Zuletzt wurde die Schnittstelle zwischen Arbeitslistenverwaltung und Arbeitslistenklient entworfen. Die besondere Anforderung hierbei war, die Schnittstelle so zu gestalten, dass verschiedene Klienten mit unterschiedlichen Ansprüchen an die Funktionalität darüber versorgt werden können. Einfache HTML-Klienten sollen einfach und schnell Zugriff auf die von ihnen benötigten Daten bekommen während Rich Clients eine weitaus größere Funktionalität und Datenmenge in Anspruch nehmen können. Das Ergebnis des Entwurfes ist die Spezifikation mehrerer Schnittstellen. Eine davon enthält die Grundfunktionalität der Arbeitslistenverwaltung und muss zwingend vom Klienten benutzt werden. Die Verwendung der anderen Schnittstellen erlaubt es dem Klienten erweiterte Funktionalität, wie z. B. die Delegation von Aktivitäten, in Anspruch zu nehmen und ist daher optional.

Literaturverzeichnis

- [AAE96] Alonso, G.; Agrawal, D.; El Abbadi, A.: *Process Synchronization in Workflow Management Systems*, 8th IEEE Symposium on Parallel and Distributed Processing (SPDS'97), Oktober 1996
- [ADEPT2] ADEPT2 – Next Generation Workflow Technology, <http://www.informatik.uni-ulm.de/dbis/index01.htm?01/forschung/projects/adept/adept.htm> (Stand: 07.05.2007)
- [AlHa97] Alanyali, M.; Hajek, B.: *Analysis of Simple Algorithms for Dynamic Load Balancing*, Mathematics of Operations Research, Volume 22 , Issue 4, 840 - 871, November 1997
- [ABLZ00] Atkinson, C.; Bayer, J.; Laitenberger, O.; Zettel, J.: *Component-Based Software Engineering. The Kobra Approach*, 3rd ICSE Workshop on Component-Based Software Engineering. Reflection on Practice. Workshop Proceedings Limerick, 2000
- [Atkin03] Atkinson, C.; Bär, H.; Bayer, J.; Bunse, Ch.; Girard, J.-F.; Gross, H.-G.; Kettemann, S.; Kolb, R.; Kühne, Th.; Romberg, T.; Seng, O.; Sody, P.; Tolzmann, E.: *Handbuch zur komponentenbasierten Softwareentwicklung*. Fraunhofer IESE, FZI; Version 1.0, 15. Januar 2003
- [Bach00] Bachmann, F.; Bass, L.; Buhman, C.; Comella-Dorda, S.; Long, F.; Robert, J.; Seacord, R.; Wallnau, K.: *Volume II: Technical Concepts of Component-Based Software Engineering, 2nd Edition*, Technical Report CMU/SEI-2000-TR-008, Mai 2000
- [Berr05] Berroth, Marco: *Konzeption und Entwurf einer Komponente für Organisationsmodelle*, Diplomarbeit an der Universität Ulm, Fakultät für Informatik, Juni 2005
- [BoRi01] Bon, M., Ritter, N.: *Interoperable Workflows*, TU Kaiserslautern, FB Informatik, Lehrgebiet Informationssysteme, Interner Bericht, April 2001
- [BRZ00] Bon, M.; Ritter, N.; Zimmermann, J.: *Interoperabilität heterogener Workflows*, Grundlagen von Datenbanken, 11-15, 2000
- [BSCW] BSCW – Basic Support for Cooperative Work, <http://www.bscw.de/produkt.html>, (Stand: 28.05.2007)
- [BSKa96] Ben-Shaul, I.Z.; Kaiser, G.E.: *Integrating groupware activities into workflow management systems*, Proceedings of the 7th Israeli Conference on Computer-Based Systems and Software Engineering, 1996

-
- [Buss94] Bussler, C.: *Policy Resolution in Workflow Management Systems*, Digital Technical Journal 6, No. 4, 1994
- [Casa99] Casati, F.: *Semantic Interoperability in Interorganizational Workflows*. Cross-Organisational Workflow Management and Co-ordination 1999
- [CCPP96] Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G.: *Semantic workflow interoperability*. In Proc. of the 5th Conf. on Extending Database Technology (EDBT), 443-462, 1996
- [CLWK00] Cai, X.; Lyu, M.R.; Wong, K.; Ko, R.: *Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes*, Proceedings of the 7th APSEC, 2000
- [COSA] COSA Business Process Management, <http://www.cosa.de/Workflow.html> (Stand: 07.05.2007)
- [Crnk01] Crnkovic, I.: *Component-based Software Engineering – New Challenges in Software Development*, Mälardalen University, August 2001
- [Dada06a] Dadam, P.; Acker, H.; Göser, K.; Jurisch, M.; Kreher, U.; Lauer, M.; Rinderle, S.; Reichert, M.: *ADEPT2 – Ein adaptives Prozess-Management-System der nächsten Generation*, Tagungsband Stuttgarter Softwaretechnik Forum 2006, November 2006
- [DaSc81] Daduna, H.; Schassberger, R.: *A discrete-time round-robin queue with bernoulli input and general arithmetic service time distributions*, Acta Informatica 15, 251-263, Juni 1981
- [DeHa01] Desmond, P.; Hancock, P.: *Stress, Workload, and Fatigue*, Lawrence Erlbaum Associates Inc, Januar 2001
- [DRRA05] Dadam, P.; Reichert, M.; Rinderle, S.; Atkinson, C.: *Auf dem Weg zu prozessorientierten Informationssystemen der nächsten Generation – Herausforderungen und Lösungskonzepte*. Tagungsband doIT-Forschungstag, Juni 2005
- [Emme02] Emmerich, W.: *Distributed Component Technologies and their Software Engineering Implications*, Proceedings of the 24th International Conference on Software Engineering, 537-546, 2002
- [EPPR99] Eder, J.; Panagos, E.; Pozewaunig, H.; Rabinovich, M.: *Time Management in Workflow Systems*, 3rd International Conference on Business Information Systems, 265-280, 1999
- [FileNet] FileNet Content Manager
http://www.filenet.com/English/Products/Content_Manager/,
(Stand 28.05.2007)

-
- [FSP00] Faissstnauer, C.; Schmalstieg, D.; Purgathofer, W.: *Priority round-robin scheduling for very large virtual environments*, Virtual Reality 2000. Proceedings. IEEE, S. 135-142, März 2000
- [GaEd97] Garbarino, E.; Edell, J.: *Cognitive Effort, Affect and Choice*, Journal of Consumer Research, Sep 1997
- [GWWK00] Gillmann, M.; Weissenfels, J.; Weikum, G.; Kraiss, A.: *Performance and Availability Assessment for the Configuration of Distributed Workflow Management Systems*, Advances in Database Technology - EDBT 2000: 7th International Conference on Extending Database Technology, März 2000
- [Hast99] Hastedt-Marckwardt, C.: *Workflow-Management-Systeme: Ein Beitrag der IT zur Geschäftsprozeß-Orientierung & -Optimierung – Grundlagen, Standards und Trends*, Informatik Spektrum 22: 99–109, 1999
- [HBCM98] Harchol-Balter, M.; Crovella, M.; Murta, C.: *On Choosing a Task Assignment Policy for a Distributed Server System*, Computer Performance Evaluation: 10th International Conference, Tools'98, LNCS 1469, pp. 231-242, 1998
- [Hein00] Heinlein, Christian: *Workflow- und Prozeßsynchronisation mit Interaktionsausdrücken und Graphen – Konzeption und Realisierung eines Formalismus zur Spezifikation und Implementierung von Synchronisationsbedingungen*, Dissertation an der Universität Ulm, Fakultät für Informatik, Mai 2000
- [JCSS01] Jin, L.; Casati, F.; Sayal, M.; Shan, M.: *Load balancing in distributed workflow management system*, ACM Symposium on Applied Computing (SAC 2001) 11-14, März 2001
- [KVV01] Kumar, A.; Van Der Aalst, W.; Verbeek, E.: *Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance*, Journal of Management Information Systems, Vol. 18, No. 3, 157-193. 2002
- [LeRo00] Leymann, F.; Roller, D.: *Production Workflow – Concepts and Techniques*, Prentice Hall PTR, New Jersey, 2000
- [Lexign] Lexign Flow, <http://www.keyproducts.de/Keyflow/index2.html>, (Stand 28.05.2007)
- [Lotus] IBM Lotus Notes, <http://www-142.ibm.com/software/sw-lotus/products/product4.nsf/wdocs/noteshomepage> (Stand: 28.05.2007)
- [LZC05] Liao, X.; Zhang, L.; Chan, S.: *A Task-Oriented Access Control Model for WfMS*, Lecture Notes in Computer Science, Volume 3439/2005, 168-177, 2005

-
- [METEOR] METEOR – Managing End-To-End OpeRations
<http://lstdis.cs.uga.edu/projects/past/METEOR/> (Stand: 07.05.2007)
- [MQWf] WebSphere MQ Workflow
<http://www-306.ibm.com/software/integration/wmqwf/> (Stand: 07.05.2007)
- [MQWF03] *IBM WebSphere MQ Workflow Best Practices Guide*, Version 1.0, Juli 2003
- [MQWF05] *IBM WebSphere MQ Workflow Programming Guide*, Version 3.6, Eleventh Edition, Juni 2005
- [MWiki] MediaWiki, *<http://www.mediawiki.org/wiki/MediaWiki>*, (Stand: 28.05.2007)
- [PaRa97] Panagos, E.; Rabinovich, M.: *Reducing Escalation-Related Costs in WFMSs, NATO Advanced Study Institute (ASI) on Workflow Management Systems and Interoperability*, August 1997
- [Reic00] Reichert, Manfred: *Dynamische Ablaufänderungen in Workflow-Management-Systemen*, Dissertation an der Universität Ulm, Fakultät für Informatik, Mai 2000
- [RRD03c] Reichert, M.; Rinderle, S.; Dadam, P.: *ADEPT Workflow Management System: Flexible Support For Enterprise-wide Business Processes (Tool Presentation)*. Proc. Int'l Conf. on Business Process Management (BPM '03), Eindhoven, Netherlands, June 2003, LNCS 2678, pp. 371-379
- [RRKD05] Reichert, M.; Rinderle, S.; Kreher, U.; Dadam, P.: *Adaptive Process Management with ADEPT2*. Proc. Int'l Conf. on Data Engineering, ICDE 2005, Tokyo, April 2005, Demo Session
- [Scha81] Schassberger, R.: *On the response time distribution in a discrete round-robin queue*, Acta Informatica 16, 57-62, August 1981
- [ShVa95] Shreedhar, M.; Varghese, G.: *Efficient fair queueing using deficit round-robin*, SIGCOMM, S. 231-242, October 1995
- [SiKa93] Sinha, A.B.; Kale, L.V.: *A load balancing strategy for prioritized execution of tasks*, Parallel Processing Symposium, 1993., Proceedings of Seventh International, April 1993
- [SoKi99] Son, J.H.; Kim, M.H.: *Improving the performance of time-constrained workflow processing*, Journal of Systems and Software, Volume 58, Issue 3, 211-219, September 2001
- [Stellent] Stellent Universal Content Management, *<http://www.stellent.com/en/index.htm>*, (Stand 28.05.2007)

-
- [STO99] Shepherdson, J.; Thompson, S.; Odgers, B.: *Decentralised Workflows and Software Agents*, BT Technology Journal, Volume 17, Number 4, 65-71, Oktober 1999
- [TIBCO] TIBCO iProcess Suite
http://www.tibco.com/software/business_process_management/iprocess_suite/default.jsp (Stand: 07.05.2007)
- [TiPO05] *TIBCO iProcess Objects Programmer's Guide*, Version 10.3, Oktober 2005
- [TiPSO05] *TIBCO iProcess Server Object (Java) Programmer's Guide*, Version 10.3, Oktober 2005
- [TYPO3] TYPO3 Open Source Content Management System, <http://typo3.com/>, (Stand 28.05.2007)
- [Ultimus] Ultimus Business Process Management Software, <http://www.ultimus.com/>, (Stand: 03.06.2007)
- [Vand05] Vanderfeesten, Irene: *Workflow Management As You Like It*, BPTrends.com, April 2005
- [VdA03] Van der Aalst, W.: *Don't got with the flow: Web services composition standards exposed*, Trends & Controversies, IEEE Intelligent Systems, Februar 2003
- [WaTa97] Watts, J.; Taylor, S.: A Practical Approach to Dynamic Load Balancing, IEEE Transactions on Parallel and Distributed Systems, Dezember 1997
- [WCW06] Wang, J.-W.; Chang, C.-H.; Wang, F.-J.: *An Analysis of Delegation Mechanism in Workflow Management System*, in NCS03, Juni 2006
- [WfMC] Workflow Management Coalition Homepage
<http://www.wfmc.org/> (Stand: 08.05.2007)
- [WfRM95] Hollingsworth, David: *The Workflow Reference Model*, Document Number TC00-1003, Januar 1995
- [WfRM98] *Workflow Management Application Programming Interface (Interface 2&3) Specification*, Document Number WFMC-TC-1009, Version 2.0, Juli 98
- [ZhTo93] Zhen, L.; Towsley, D.: *Optimality of the Round Robin Routing Policy*, Technical Report: UM-CS-1992-055, Februar 1993 (Revised Version)

Abbildungsverzeichnis

Abbildung 1-1: Einteilung von unterstützenden Softwaresystemen [LeRo00]	7
Abbildung 1-2: Arbeitsumgebung eines Workflow-Management-Systems	8
Abbildung 1-3: Darstellung einer Arbeitsliste in einem einfachen HTML-Klient	9
Abbildung 1-4: Darstellung einer Arbeitsliste in einem Rich Client	9
Abbildung 1-5: Die Arbeitslistenverwaltung als Verbindung zwischen WfMS und Klient	10
Abbildung 1-6: Verteilung der Aktivitäten auf die Arbeitslisten der Benutzer	11
Abbildung 1-7: Starten und Entfernen von Aktivitäten aus den Arbeitslisten	11
Abbildung 1-8: erweiterte Verteilungsverfahren	12
Abbildung 1-9: Die ALV als unabhängige Komponente	13
Abbildung 1-10: Ausschnitt aus dem Organisationsmodell einer Klinik	14
Abbildung 1-11: Workflow einer Untersuchung innerhalb der Abteilung für Innere Medizin	14
Abbildung 1-12: Komponenten zur Erweiterung der Funktionalität	15
Abbildung 2-1: Workflow-Referenzmodell der WfMC [WfRM95]	18
Abbildung 2-2: Die Verbindung der ALV zu WfMS und Klient	19
Abbildung 3-1: Auflösung von Eskalationsempfängern über eine Bearbeiterformel	22
Abbildung 3-2: persönliche und gruppenbezogene Arbeitslisten	23
Abbildung 3-3: Verwendung alternativer Bearbeiterformeln	24
Abbildung 3-4: Auflösung der Bearbeiterformeln über die ALV	25
Abbildung 3-5: Auflösung der Bearbeiterformeln über das WfMS	26
Abbildung 3-6: Hinterlegung der Bearbeiterformeln im Prozessmodell	27
Abbildung 3-7: Hinterlegung der Bearbeiterformeln in der ALV	28
Abbildung 3-8: Persönliche und Gruppenarbeitslisten	30
Abbildung 3-9: interne und Klient-Arbeitslisten	31
Abbildung 3-10: Übermittlung einer kompletten Arbeitsliste	32
Abbildung 3-11: Erstellen und Übermitteln einer Aktualisierung	33
Abbildung 3-12: Verwaltung und Übermittlung einer Aktualisierung	34
Abbildung 4-1: Round-Robin-Prinzip	37
Abbildung 4-2: Zusammenhang von Bearbeitermengen	38
Abbildung 4-3: Mitarbeiterliste vor Round-Robin-Zuweisung	38
Abbildung 4-4: Mitarbeiterliste nach 1. Round-Robin-Zuweisung	39
Abbildung 4-5: Mitarbeiterliste nach 2. Round-Robin-Zuweisung	39
Abbildung 4-6: Bestimmung des nächsten Bearbeiters nach Round Robin	39
Abbildung 4-7: Auslastungsverteilung nach Auslastungsgrad	41
Abbildung 5-1: Delegation einer Aktivität	48
Abbildung 5-2: Eskalation einer Aktivität	48
Abbildung 5-3: kombinierte Eskalation und Delegation	49
Abbildung 5-4: Delegationstiefe	50
Abbildung 5-5: Änderung der Arbeitslisten nach einer Delegation	51
Abbildung 5-6: Vertreterregel für eine Stelle	53
Abbildung 5-7: Beispiel einer Vertreterregel für eine Stelle	54
Abbildung 5-8: Vertreterregel für einen Bearbeiter	54
Abbildung 6-1: Zeitvorgaben für die Priorisierung in der ALV	60
Abbildung 6-2: Kennzeichnung der priorisierten Aktivitäten durch zusätzliches Attribut	63
Abbildung 6-3: Farbkodierung der priorisierten Aktivitäten	63

Abbildung 6-4: Sortierung der priorisierten Aktivitäten.....	64
Abbildung 6-5: optimale Kennzeichnung von priorisierten Aktivitäten.....	65
Abbildung 7-1: Schnittstellen und Datenmodell der Arbeitslistenverwaltung	68
Abbildung 7-2: Das Datenmodell der Arbeitslistenverwaltung.....	69
Abbildung 7-3: Unterschiedliche Ausprägungen von Arbeitslisten	70
Abbildung 7-4: Unterschiedliche Ausprägungen von Arbeitslisteneinträgen.....	72
Abbildung 7-5: Zugriffe auf Daten im Bereich der ALV	73
Abbildung 7-6: Zustandsmodell für WorkItems.....	76
Abbildung 7-7: Übermittlung von Zustandsänderungen über die ALV	77
Abbildung 7-8: Übermittlung von Zustandsänderungen über das WfMS	77
Abbildung 7-9: Interaktion zwischen ADEPT2 und der ALV	79
Abbildung 7-10: Die Zustände einer ADEPT2-Aktivität	80
Abbildung 7-11: Zustände eines WorkItems	80
Abbildung 7-12: Teilausschnitt aus dem Datenmodell von MQ Workflow [MQWF05]	82
Abbildung 7-13: Zugriff auf WorkItems innerhalb von MQ Workflow [MQWF05].....	82
Abbildung 7-14: Zustandsmodell eines WorkItems in MQ Workflow [MQWF05].....	83
Abbildung 7-15: Zustandsänderungen in MQ Workflow [MQWF05]	83
Abbildung 7-16: Teilausschnitt aus dem Datenmodell der TIBCO iProcess Engine [TiPO05]....	85
Abbildung 7-17: Die Zuordnung von WorkItems zu Benutzern [TiPO05].....	85
Abbildung 7-18: Zugriff auf WorkItems in der TIBCO iProcess Engine [TiPSO05].....	85
Abbildung 7-19: Zustandsmodell der WorkItems in der TIBCO iProcess Engine [TiPSO05]..	86
Abbildung 7-20: Zustandsänderungen bei TIBCO iProcessEngine [TiPSO05]	86
Abbildung 7-21: Zusammenarbeit von WfMS und ALV	88
Abbildung 7-22: Die Schnittstelle WorklistInteraction	89
Abbildung 7-23: Die Schnittstelle OrgModelConnection.....	89
Abbildung 7-24: Die Schnittstelle WorklistNotification.....	90
Abbildung 7-25: Die Schnittstelle WorklistRequest.....	91
Abbildung 7-26: Schnittstellen der ALV zum Arbeitslistenklienten	92
Abbildung 7-27: Die Bearbeiterzuordnung eingegliedert in die ALV	95
Abbildung 7-28: Die Bearbeiterzuordnung ausgegliedert aus der ALV	95
Abbildung 7-29: die Schnittstellen für Verteilungsverfahren	96
Abbildung 7-30: Ablauf einer Verteilung	97
Abbildung 7-31: Priorisierung integriert in die ALV	100
Abbildung 7-32: Priorisierung als eigene Komponente	101
Abbildung 7-33: teilweise Auslagerung des Priorisierungsdienstes	102
Abbildung 7-34: Überblick über die Schnittstellen der ALV	104

Glossar

ADEPT2	Zweiter Prototyp des an der Universität Ulm entwickelten fortschrittlichen und adaptiven WfMS. Es unterstützt u. a. flexible Änderungen von Prozessinstanzen, dynamische Prozessänderungen und ihre Übertragung auf laufende Instanzen sowie die Analyse und Verifizierung von Prozessvorlagen. [ADEPT2]
ALV	Arbeitslistenverwaltung, verwaltet Arbeitslisten und Aktivitäten für jeden Bearbeiter und stellt erweiterte Funktionalität dafür zur Verfügung. Sie bildet die Vermittlungsstelle zwischen WfMS und Klient.
Aktivität	Ein für die Abarbeitung aktivierter Prozessschritt, welcher der ALV übergeben wurde.
Arbeitsliste	Enthält eine Liste aller Aktivitäten, die ein bestimmter Bearbeiter erledigen kann. Arbeitslisten können persönlich für einen Bearbeiter oder als Gruppenarbeitsliste für eine bestimmte Organisationseinheit gehalten werden.
Arbeitslistenklient	Eine Anwendung, die für die Darstellung einer Arbeitsliste beim Bearbeiter verantwortlich ist.
Ausführungskomponente	Ein Modul innerhalb eines WfMS, welches dafür verantwortlich ist Prozessinstanzen auszuführen, zu überwachen und weiterzuschalten.
Bearbeiter	Ein Anwender, der Arbeitslisten besitzt und darin befindliche Aktivitäten erledigen kann.
Bearbeiterformel	Eine Zuordnungsvorschrift, die angibt, welche Bearbeiter eine Aktivität bearbeiten können.
CMS	Content-Management-System, ermöglicht und organisiert die gemeinsame Bearbeitung von Dokumenten oder Multimediadateien. Beispiele für CMS sind das FileNet CMS von IBM [FileNet], Stellent von Oracle [Stellent] und das Open Source Projekt TYPO3 [TYPO3].
CSCW	Computer Supported Cooperative Work – rechnergestützte Gruppenarbeit – ein Oberbegriff für alle Anwendungen, welche die gemeinsame Erledigung von Aufgaben ermöglichen. Bekannte CSCW-Systeme sind u. a. IBM's Lotus Notes [Lotus], BSCW [BSCW] und MediaWiki [MWiki].

Delegation	Die Verschiebung einer Aktivität zu einem anderen Bearbeiter, der sich in der gleichen oder einer tieferen Hierarchieebene befindet.
Erstverteilung	Die Verteilung einer Aktivität nach einer Bearbeiterformel. Sie wird vorgenommen, sobald die Aktivität vom WfMS übergeben wird.
Eskalation	Die Verschiebung einer Aktivität zu einem anderen Bearbeiter, der sich in einer höheren Hierarchieebene befindet.
Execution Manager	Die Ausführungskomponente eines WfMS. Sie ist für die Ausführung und Weiterschaltung von Prozessen innerhalb eines WfMS verantwortlich.
Fast User Switching	Verfahren, bei dem sich ein Bearbeiter schnell und kurzzeitig von der ALV abmelden kann, ohne dass seine Arbeitsliste verworfen wird.
Groupware	Die Gesamtheit aller Systeme, die dazu beitragen CSCW zu ermöglichen.
HTML-Klient	Eine Klientenanwendung, deren Verarbeitungslogik auf die Ein- und Ausgabe von Informationen beschränkt ist.
Load Balancing	Ein Verteilungsverfahren, bei dem Aktivitäten auf die am wenigsten ausgelasteten Bearbeiter verteilt werden.
MQ Workflow	Ein WfMS der Firma IBM. Es gehört zur WebSphere MQ Serie, deren Teilprodukte auf dem Prinzip der Warteschlangen (Message Queue) basieren. [MQWf]
Neuverteilung (dynamisch)	Die Umverteilung einer Aktivität, um die Bearbeiterzuordnung an geänderte Umgebungsbedingungen anzupassen.
Organisationseinheit	Eine Einheit im Organisationsmodell. Dazu gehören Stelle und Rollen genauso wie Mitarbeiter.
Organisationsmodell	Komponente des WfMS, die Informationen über Bearbeiter, Stellen und Rollen einer Organisation enthält.
Priorisierung	Versehen eines Prozessschrittes mit Prioritäten, um die Abarbeitung seitens des Bearbeiters zu beschleunigen.
Prozess	Ein Arbeitsablauf, der aus mehreren Prozessschritten besteht, die in einer vorgegebenen Reihenfolge abgearbeitet werden müssen.

Prozessinstanz	Ein Prozess, der nach einer Prozessvorlage abgeleitet wurde und jetzt ausgeführt wird.
Prozessmodell	Eine Komponente innerhalb eines WfMS, in der Prozessvorlagen hinterlegt werden und welche die Definition, Dokumentation und Analyse von Prozessen unterstützt.
Prozessschritt	Ein Knoten innerhalb eines Prozesses. Er kann eine ausführbare Aktivität darstellen oder einen Unterprozess.
Prozessvorlage	Definierter Prozess, bei dem neben den einzelnen Schritten auch Reihenfolge und Datenbereitstellung festgelegt wurden.
Pull	Verfahren, bei dem die Aktualisierung der Arbeitsliste des Klienten manuell angestoßen werden muss.
Push	Verfahren, bei dem die Aktualisierung der Arbeitsliste des Klienten von der ALV angestoßen wird.
Ressourcenverwaltung	Überwacht Ressourcen und ihre Verfügbarkeit innerhalb eines WfMS.
Rich Client	Klientenanwendung, die nicht nur die grafische Benutzerschnittstelle enthält, sondern auch große Teile der Anwendungs- und Verarbeitungslogik implementiert.
Rolle	Eine bestimmte Position innerhalb einer Stelle, die von einem oder mehreren Mitarbeitern besetzt werden kann. Eine Rolle umfasst eine Menge von Fähigkeiten, die zur Erledigung von bestimmten Aufgabentypen innerhalb der Stelle nötig sind. Sie ist Teil des Organisationsmodells.
Round Robin	Verteilungsverfahren, das eine Aktivität sequentiell über eine geordnete Bearbeitermenge verteilt.
Stelle	Eine Organisationseinheit, die sich aus einer oder mehreren Rollen zusammensetzt und verschiedene Mitarbeiter verwaltet. Ist Teil des Organisationsmodells.
TIBCO iProcess Engine	Teil der TIBCO iProcess Suite, einem umfangreichen WfMS, und dort verantwortlich für die Ausführung von Geschäftsprozessen. [TIBCO]
Verteilungsverfahren	Wählt aus einer gegebenen Bearbeitermenge eine bestimmte Teilmenge von Bearbeitern aus, denen eine Aktivität in die Arbeitsliste gestellt wird.

Vertreterregel	Eine Zuordnungsvorschrift, die für einen abwesenden Bearbeiter oder eine unterbesetzte Stelle eine Vertretung festlegt.
WfMC	Workflow-Management-Coalition, ein Gremium zur Standardisierung von Workflow-Komponenten. [WfMC]
WfMS	Workflow-Management-System, ein System zur Prozessunterstützung
Workflow Referenzmodell	Modell eines standardisierten WfMS nach der WfMC [WfRM95]

Eidesstattliche Erklärung

Name: Romy Opitz

Matrikelnummer: 609025

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den _____

(Unterschrift)