

Studienarbeit

**Unterstützung kooperativer Arbeit
mit Hilfe eines Transportsystems**

Mirko Freund
Vor der Mühle 12
21640 Horneburg
3freund@informatik.uni-hamburg.de

14.Juni 1998

Betreuung:

Prof. Heinz Züllighoven
Fachbereich Informatik
Arbeitsbereich Softwaretechnik
Universität Hamburg
Vogt-Kölln-Str. 30
22527 Hamburg

Inhaltsverzeichnis

1	EINLEITUNG	5
2	ANWENDUNGSFÄLLE KOOPERATIVER ARBEIT.....	9
2.1	BEARBEITUNG EINES TODESFALLVORGANGES IN EINER BANK.....	9
2.2	SZENARIO EINER BLUTGRUPPENBESTIMMUNG	10
2.3	UNTERSCHIEDE UND GEMEINSAMKEITEN.....	12
2.4	ZUSAMMENFASSUNG.....	13
3	KONZEPTION DES TRANSPORTSYSTEMS	15
3.1	AUFBAU	15
3.2	FACHLICHE KOMPONENTEN	17
3.2.1	<i>Adresse</i>	18
3.2.2	<i>Material</i>	19
3.2.3	<i>Transportmappe</i>	20
3.2.4	<i>Postkorb</i>	22
3.2.5	<i>Versandautomat</i>	25
3.3	ZUSAMMENSPIEL DER KOMPONENTEN	27
4	ARCHITEKTUR DER VERTEILUNGSKOMPONENTE	31
4.1	DIENSTLEISTUNGSPRINZIP.....	31
4.2	KONZEPTION DER VERTEILUNGSKOMPONENTE	32
4.3	REMOTE METHODE INVOCATION	34
4.4	NACHRICHTENVERMITTLUNG.....	36
4.4.1	<i>Aufbau</i>	37
4.4.2	<i>Komponenten</i>	38
4.4.2.1	Nachricht.....	38
4.4.2.2	Klausel.....	39
4.4.2.3	MessageBroker.....	39
4.4.2.4	MessageBrokerServer	40
4.5	REALISIERUNG DER VERTEILUNGSKOMPONENTE	40
5	ZUSAMMENFASSUNG UND AUSBLICK	45
6	LITERATURVERZEICHNIS	47

1 Einleitung

Am Arbeitsbereich Softwaretechnik der Universität Hamburg gibt es seit wenigen Jahren Bestrebungen, der wachsenden Bedeutung kooperativer Arbeitsprozesse gerecht zu werden. Dazu wurde die Softwareentwicklungsmethode WAM weiterentwickelt, die von einer Einzelarbeitsplatzumgebung ausging. Mit meiner Arbeit möchte ich einen weiteren Beitrag in dieser Forschungsrichtung leisten. Um den Ausgangspunkt, das Vorgehen sowie das Ziel meiner Studienarbeit deutlich darzustellen, wird auf die folgenden Fragen im einzelnen kurz eingegangen.

- Welche Thematik wird bearbeitet?
- In welchem Kontext steht die Studienarbeit?
- Was ist der gegenwärtige Wissensstand auf diesem Gebiet?
- Wie soll die Unterstützung kooperativer Arbeit aussehen?
- Was ist das Ziel dieser Arbeit?
- Wie ist der Aufbau dieser Arbeit?

Welche Thematik wird bearbeitet?

In der heutigen Zeit gewinnen kooperative Arbeitsprozesse immer mehr an Bedeutung. Zu deren Unterstützung haben sich mehrere verschiedene Kooperationsmedien, wie z.B. Postfächer, Rohrpost und Postversandsystem (vgl. [WAM98], Kapitel: „Unterstützung kooperativer Arbeit“), in der fachlichen Anwendungswelt etabliert.

In dieser Arbeit möchte ich mich genau dieser Thematik widmen und einen softwaretechnischen Beitrag dazu leisten. Dabei werde ich mich in meinen weiteren Ausführungen auf ein Kooperationsmedium, dem Transportsystem (in [WAM98] als Postversandsystem bezeichnet) beschränken. Zuerst werden anhand von zwei Beispielen aus einem Projekt des Arbeitsbereiches Softwaretechnik der Universität Hamburg die Kooperationsformen und somit die Anforderungen an eine computergestützte Kooperation ausgearbeitet. Anschließend wird ein fachliches Modell eines Transportsystems entwickelt, das den skizzierten Anforderungen entspricht. Zum Schluß wird dieses mit Hilfe bereits vorhandener technischer Mittel realisiert.

In welchem Kontext steht die Studienarbeit?

Die gesamte entsteht Arbeit vor dem Hintergrund der Softwareentwicklungsmethode WAM, welche am Arbeitsbereich Softwaretechnik der Universität Hamburg vertreten wird. Hierbei steht die Abkürzung WAM für **W**erkzeug, **A**utomat und **M**aterial. Diese Begriffe sind Metaphern der Methode, die zur Entwicklung von interaktiven Anwendungssystemen dient. Grundlage der Methode ist der Arbeitsplatz für eigenverantwortliche qualifizierte Tätigkeit. Dem Anwender wird ein Expertenwissen vorausgesetzt, das ihm ein eigenständiges Handeln erlaubt.

Was ist der gegenwärtige Wissensstand auf diesem Gebiet?

Die WAM-Methode ging ursprünglich von einer *Einzelarbeitsplatzumgebung* aus. Doch sehr häufig steht nicht die Arbeit eines Einzelnen im Vordergrund, sondern es muß noch mit weiteren involvierten Personen kooperiert werden. Aber zu der Problematik der Kooperation war kein Ansatz in der Methode vorhanden. So kam es in den letzten Jahren verstärkt zu Vorhaben, Erkenntnisse auf diesem Gebiet zu erweitern. Ergebnisse dieser Vorhaben sind u.a. die Arbeiten von G. Grycan (vgl. [Gry95]) und M. Wulf (vgl. [Wulf95]). Letztgenannte dient als wichtigste Grundlage dieser Arbeit.

Wie soll die Unterstützung kooperativer Arbeit aussehen?

In den gesammelten Erkenntnissen aus den oben genannten Arbeiten diene ein Transportsystem zur Unterstützung kooperativer Arbeitsprozesse. Dieses besteht aus den folgenden Komponenten:

- Transportmappe, zur Aufnahme aller zu sendenden Materialien
- Posteingangskorb, zur Aufnahme der empfangenen Transportmappen
- Postausgangskorb, zur Aufnahme der zu sendenden Transportmappen
- Versandautomat, zum automatischen Versand aller Transportmappen

In meiner Arbeit baue ich auf diesem Leitbild auf und konkretisiere es.

Was ist das Ziel dieser Arbeit?

Das Ziel dieser Arbeit ist, ein Konzept des Transportsystems sowie dessen Realisierung vorzustellen. Dabei ist ein Schwerpunkt, fachlich motiviert und detailliert das Transportsystem mit seinen Komponenten vorzustellen sowie dessen Funktionsweise. Weiterhin wird aus softwaretechnischer Sicht die Architektur dargestellt, um die Möglichkeiten der Integration in einer Softwareumgebung bzw. dessen Verwendung deutlich werden zu lassen. Ein weiterer Schwerpunkt ist die Realisierung dieses Konzeptes unter Nutzung bereits vorhandener technischer Hilfsmittel in JAVA. Dabei baue ich auf das Framework JWAM, das am Arbeitsbereich entwickelt wurde, auf.

Wie ist der Aufbau dieser Arbeit?

Kapitel 1 ist die Einleitung und beschreibt den Ausgangspunkt, Motivation sowie Ziel dieser Arbeit.

Kapitel 2 befaßt sich exemplarisch mit zwei Anwendungsfällen kooperativer Arbeit. Anhand derer werden Unterschiede und Gemeinsamkeiten erörtert, um ein fachliches Modell des Transportsystems zu entwickeln.

Kapitel 3 schildert die Entwicklung eines Konzeptes für das Transportsystems, das dem zuvor aufgestellten fachlichen Modell gerecht wird. Weiterhin werden die einzelnen Komponenten des aufgestellten Konzeptes im einzelnen erörtert, deren Umgangsformen ausgearbeitet sowie das Zusammenspiel im ganzen erklärt.

Kapitel 4 stellt die Umsetzung des Konzeptes in JAVA dar. Dabei werden hier auf die bereits vorhandenen Hilfsmittel, wie der RMI-Mechanismus von Java sowie die

Nachrichtenvermittlung des JWAM, eingegangen.

Kapitel 5 gibt einen Ausblick auf die weiteren Entwicklungs- sowie Verwendungsmöglichkeiten des Transportsystems.

Kapitel 6 beinhaltet das Literaturverzeichnis.

Danksagung:

An dieser Stelle möchte ich Carola Lilienthal für ihre umfangreiche Hilfe und Betreuung danken. Sie gab mir viele neue Ideen und war unermüdlich in der Korrektur der vorliegenden Arbeit. Weiterhin danke ich allen Mitstudenten insbesondere Martin Lippert, die mir reichlich Ratschläge sowie Feedback aus der Verwendung der erstellten Software gaben.

2 Anwendungsfälle kooperativer Arbeit

In diesem Kapitel werden zwei Szenarios vorgestellt, anhand derer mögliche Ausprägungen kooperativer Arbeit deutlich werden. Weiterhin dienen sie als Grundlage für die Entwicklung eines fachlichen Modells des Transportsystems. Dazu werden die Unterschiede und Gemeinsamkeiten der verschiedenen Anwendungsfälle aufgedeckt und ein Begriffsgerüst für das Transportsystem erstellt. Der Übersichtlichkeit halber sind die für das Transportsystem fachlich relevanten Begriffe im Text hervorgehoben.

Die folgenden beiden Szenarios sind Zusammenfassungen aus einer Projektdokumentation, die am Arbeitsbereich Softwaretechnik erstellt wurde. Sie beschränken den Informationsgehalt auf den für das Transportsystem relevanten Teil. Die vollständigen Szenarios sind im Anhang ersichtlich.

2.1 Bearbeitung eines Todesfallvorganges in einer Bank

Die Bank kann aus verschiedenen Quellen vom Tod eines Kunden Kenntnis erlangen. Solche wären z.B. Mitteilung durch einen Angehörigen, Informationsbeschaffung aus Tageszeitungen und amtlichen Mitteilungen, und weiterer informeller Kanäle.

Nach erfolgreicher Prüfung, ob es sich bei dieser Person um einen Kunden der Bank handelt, wird ein Servicevorgang angelegt. Dieser wird anschließend an die Abteilung bzw. Mitarbeiter für die Nachlaßbearbeitung weitergeleitet. Ein Vorgang besteht im wesentlichen aus einer **Mappe** mit einer Checkliste und weiterer zugehöriger **Dokumente**. Die Checkliste wird benutzt, um sicherzustellen, daß keine notwendigen Aktionen zur Vorgangsbearbeitung ausgelassen werden. Weiterhin wird sie verwendet, um evtl. Abhängigkeiten zwischen den einzelnen Aktivitäten zu berücksichtigen. Ein Beispiel hierfür wäre, daß erst ein Erbschein und eine Weisung der Erben vorliegen müssen, bevor ein Konto aufgelöst werden darf.

Der nun mit diesem Todesfall beauftragte Mitarbeiter der Bank beginnt seinen Teil am gesamten Vorgang entsprechend der Checkliste zu bearbeiten. Danach wird der Vorgang an den für den nächsten Teil zuständigen Mitarbeiter weitergeleitet. Während der Mitarbeiter gemäß der Checkliste seine Aktionen ausführt, beschafft er sich notwendige Informationen und leitet verschiedenen Untervorgänge ein, die außerhalb seines Tätigkeitsfeldes liegen.

Die Beschaffung der zusätzlichen Informationen kann er durch eine mündliche oder schriftliche Anforderung erfolgen. Dazu müssen meistens weitere bereits vorhandene Informationen bzw. Dokumente den Befragten zugesandt werden. Ein Beispiel dafür wäre, daß der Mitarbeiter wissen möchte, ob der Verstorbene fest angelegtes Kapital hatte. Doch um diese Information zu erhalten, sollte der Befragte wenigsten den Namen des Kunden übermittelt bekommen.

Wie zu erkennen ist, ist bei der Beschaffung von Informationen ein Informationsfluß vom anfragenden zum befragten Mitarbeiter vorhanden.

Ähnlich sieht es bei der Initiierung weiterer Untervorgänge aus. So kann z.B. ein Mitarbeiter veranlaßt werden, das Konto des Verstorbenen aufzulösen. Doch dafür benötigt er die Kontonummer sowie den Namen des Kontoinhabers.

Damit wird deutlich, daß auch bei der Initiierung eines Untervorganges ein Informationsfluß vom Initiator zum Ausführenden vorhanden ist.

Bisher ist lediglich deutlich geworden, daß Informationen vom anfragenden zum ausführenden Mitarbeiter gesendet werden. Aber auch in umgekehrter Richtung kann es zum Informationsaustausch kommen. So ist der Mitarbeiter der Nachlaßbearbeitung an den Ergebnissen seiner Anfragen bzw. eingeleiteten Untervorgänge interessiert. So möchte er z.B. bei seiner Anfrage auf evtl. Kapitalanlagen wissen, ob und welche Anlagen auf den verschiedenen Depots des Kunden existieren. Genauso möchte er auch den erfolgreichen Abschluß eines Untervorganges, z.B. einer Kontoauflösung, wissen.

Doch wie wird nun dieser Informationsaustausch unterstützt? Dazu bedient sich die Bank der **Hauspost** als Transportmechanismus. Dabei werden sowohl einzelne Dokumente bzw. Nachrichten als auch der gesamte Vorgang versendet. Die Entscheidung, was versendet werden soll, ist vom Kontext und dem aktuellen Bearbeitungszustand abhängig. Somit liegt es in der Entscheidungsgewalt des Mitarbeiters, was versendet wird.

Die zu sendenden Dokumente werden mit der **Adresse** des **Empfängers** beschriftet. Häufig wird zusätzlich ein **Absender** angegeben, um Rückläufe wie z.B. Ergebnisse von Teilaktivitäten zu ermöglichen. Anschließend wird die Sendung in den **Postausgangskorb** gelegt, aus dem sie später von einem **Postboten** abgeholt wird. Dieser befördert sie zum Empfänger und legt sie dort in den **Posteingangskorb**. Zu einem späteren Zeitpunkt kann der Sachbearbeiter die erhaltene Sendung, z.B. ein Vorgang, aus dem Posteingangskorb entnehmen und bearbeiten. Nachdem er sie bearbeitet hat, sendet er sie zurück oder leitet sie weiter.

Bei der gewöhnlichen Vorgehensweise zur Versendung von Dokumenten kann es zu zwei möglichen Abweichungen kommen.

1. In einigen Bankfilialen gibt es sogenannte „intelligente“ Postboten, die bei der Versendung von gesamten Vorgängen anhand der Checkliste selbständig erkennen können, wer der Empfänger ist.
2. Mitunter wird die Hauspost umgangen und der Mitarbeiter legt selbst die Sendung, z.B. einen Vorgang, in das Postfach des Empfängers.

2.2 Szenario einer Blutgruppenbestimmung

Aus den verschiedensten Gründen kann es erforderlich sein, die Blutgruppe eines Patienten zu bestimmen. Der Ausgangspunkt kann sowohl eine Station in einem Krankenhaus als auch eine Ambulanz sein. Doch bevor der genauere Ablauf geschildert wird, soll Abbildung 1 einen ersten Überblick vermitteln.

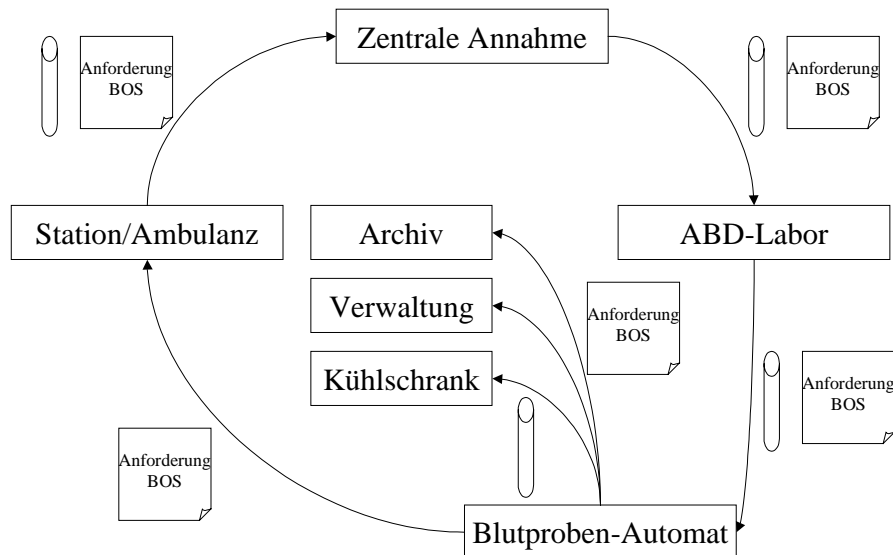


Abbildung 1: Kooperationsbild für die Anforderung und Bearbeitung einer Blutgruppenbestimmung

Um eine Bestimmung der Blutgruppe vorzubereiten, füllt die Schwester ein *Antragsformular* mit den Patientendaten aus und fertigt einen Aufkleber für das Blutprobenröhrchen an. Falls es sich um eine Bestätigung der Blutgruppe handelt, sucht sie zusätzlich den *Blutgruppenoriginalschein (BOS)* aus der Patientenakte heraus. Danach legt sie diese Dokumente in den *Arztkorb*. Außerdem beklebt sie noch ein Blutprobenröhrchen mit dem vorbereiteten Aufkleber. Nachdem der Arzt alle notwendigen Eintragungen in die Formulare vorgenommen hat, legt er sie zurück in den *Stationskorb*. Die Schwester kann nun eine Blutentnahme vorbereiten. Später führt eine dafür zuständige Person, z.B. ein Arzt, die Blutentnahme durch. Die entnommenen Proben werden unregelmäßig mehrmals täglich von einem *Boten* zusammen mit den Formularen abgeholt und in die zentrale Annahme der Labore gebracht.

Ist der Ausgangspunkt eine Ambulanz, so ist das Vorgehen analog der einer Station. Im weiteren wird keine Unterscheidung zwischen beiden mehr vorgenommen.

In der zentralen Annahme werden das Blutprobenröhrchen und die Formulare entgegengenommen und auf Vollständigkeit geprüft. Anschließend werden die Formulare mit einem Eingangsstempel versehen. Zum Schluß wird die Anforderungen, bestehend aus dem Blutprobenröhrchen und den Formularen, dem ABD-Labor zugeordnet und zugestellt.

Im ABD-Labor wird zunächst geprüft, ob die Anforderung bereits vorliegt. Das kann vorkommen, wenn der Stationsarzt eine Bestimmung anordnet, ohne zu wissen, daß diese vorher schon von der Ambulanz angefordert wurde. Liegt diese Anforderung noch nicht vor, wird eine freie Befundnummer ermittelt und zusammen mit den Patientendaten in einen Hauptbuchrechner eingetragen. Anschließend wird eine Blutgruppenkurzbestimmung (ABD) durchgeführt und das Ergebnis in das Anforderungsformular eingetragen. Später wird aus Sicherheitsgründen ein Abgleich mit der angeforderten Untersuchung im Labor vorgenommen. Zum Schluß wird der gesamte Vorgang in die *Ablage* des Blutgruppenautomaten abgelegt.

Die für den Blutgruppen-Automaten zuständige Person nimmt sich die Anforderung aus ihrer Ablage. Sie führt die darin aufgeführten Untersuchungen durch und vermerkt die Ergebnisse in den Formularen. Anschließend stellt sie die Blutprobe in den Kühlschrank, wo diese noch eine weitere Woche für evtl. Nachuntersuchungen aufbewahrt wird. Nun gibt sie die Ergebnisse mit der Befundnummer in den Hauptbuchrechner ein und sendet Formulare an den Ursprungsort zurück. Wurde die Anforderung von einer Station gestellt, so werden die Formulare in dem **Postfach** der Station abgelegt. War der Antragsteller dagegen die Ambulanz, so werden die Dokumente in einer Ablagemappe abgeheftet. Zu einem späteren Zeitpunkt können dann diese Unterlagen von einem Mitarbeiter der Ambulanz abgeholt werden. Sollte aber binnen 3 Tage niemand die Dokumente holen, so wird die Station ermittelt, auf die der Patient verlegt wurde. Anschließend werden dann diese Unterlagen wie oben beschrieben in das Postfach der Station abgelegt.

2.3 Unterschiede und Gemeinsamkeiten

In diesem Kapitel werden die Unterschiede und Gemeinsamkeiten ausgearbeitet. Dabei wird der Schwerpunkt der Betrachtung auf die Ausprägungen der Kooperation sowie deren Unterstützung durch bereits vorhandene Transportsysteme gelegt. Im konkreten sollen folgende Fragen geklärt werden:

- Was wird versendet?
- Was realisiert den Versand?
- Wie wird das Transportsystem verwendet?

Die mit der Beantwortung dieser Fragen gesammelten Erkenntnissen dienen der Begriffsbildung, die für die Entwicklung des Konzeptes des computergestützten Transportsystems die Grundlage ist.

Was wird versendet?

In den Szenarios wurde deutlich, daß die Kooperation auf dem Versand von **Materialien** basiert. In dem Szenario über die Bearbeitung eines Todesfallvorganges in einer Bank (siehe Kapitel 2.1) waren das Dokumente verschiedenster Art, sowie eine Checkliste. Bei der Blutgruppenbestimmung (siehe Kapitel 2.2) stellten das Antragsformular, der Blutgruppenoriginalschein (BOS) und das Blutprobenröhrchen die Materialien dar. In beiden Szenarios wurden die zu bearbeitenden Vorgänge durch diese Materialien charakterisiert. Desweiteren wird bei der Todesfallbearbeitung die Verwendung einer Mappe erwähnt. Sie repräsentiert einen Vorgang und beinhaltet die zugehörigen Materialien sowie eine Checkliste. Diese Mappe wird mittels Hauspost zu den verschiedenen involvierten Mitarbeitern versendet. Sie dient somit Repräsentations- sowie Transportzwecken und stellt im wesentlichen eine **Transportmappe** behaftet mit einer Checkliste dar. In weiterführender Literatur (vgl. [Wulf95]) wird diese Art von Mappe als Vorgangsmappe bezeichnet.

In dem Kapitel 2.2 über die Blutgruppenbestimmung ist keine Verwendung von Transportmappen ersichtlich, die alle zu sendenden Materialien beinhalten. Das ist aber darin begründet, daß es nicht ein Vielzahl verschiedener Materialien gibt, sondern höchstens zwei Formulare und das Blutprobenröhrchen. Aufgrund dieser geringen Vielfalt an Materialien repräsentieren die Formulare und das Blutprobenröhrchen selbst den Vorgang und erübrigen

einen zusätzlichen Transportbehälter. Es wird in diesem speziellen Fall also keine extra Mappe benötigt, die den Repräsentations- sowie Transportzweck erfüllt. Die Formulare sind sozusagen Transportmappe und Material zusammen.

Wie zu sehen ist, werden in beiden Szenarios Vorgänge durch Materialien in Form von Dokumenten und Formularen charakterisiert. Diese werden zum Zwecke der Kooperation in Transportmappen versendet. Dabei kann es vorkommen, daß ein Formular selbst die Transportmappe darstellt.

Was realisiert den Versand?

In beiden Szenarios wird der Versand meistens durch eine Hauspost realisiert. Diese Hauspost wird durch einen **Boten** repräsentiert, der die zu sendenden Dokumente abholt. Dabei ist jeweils festzustellen, daß der Zeitpunkt, wann er kommt, nicht genau definiert ist. Es können aber auch Situationen vorkommen, in denen die Hauspost nicht verwendet wird, sondern die bearbeitende Person selbst die Zustellung und somit die Rolle des Boten übernimmt. Ein Grund dafür ist die konkret definierte und kürzere Zustellzeit.

Wie wird das Transportsystem verwendet?

In beiden Szenarios dienen Behälter als Schnittstelle zwischen den Mitarbeitern und der Hauspost, in denen die Sendungen abgelegt werden. Diese Behälter wurden in den entsprechenden Kapiteln als Postein- und Postausgangskorb, Arztkorb, Stationskorb, Ablage oder als Postfach bezeichnet. Rein prinzipiell entsprechen sie generell Postein- und Postausgangskörben. Die zu sendenden Transportmappen kommen in die **Postausgangskörbe**, während die empfangenen in die **Posteingangskörbe** abgelegt werden.

Doch bevor die zu sendenden Transportmappen in den Postausgangskorb kommen, müssen sie adressiert werden. Das ist in den beiden Szenarios unterschiedlich gehandhabt worden. Bei der Bearbeitung eines Todesfallvorganges in einer Bank in Kapitel 2.1 werden die Sendungen in der Regel mit je einer **Adresse** für den Empfänger sowie für den Absender versehen. Wird dagegen ein gesamter Vorgang mit Checkliste versendet, so kann ein „intelligenter“ Bote anhand der Checkliste feststellen, wer der Empfänger ist. Außerdem ist auch der Absender in diesem Fall anhand der Signaturen in der Checkliste ersichtlich. Somit ist die explizite Angabe einer Adresse für Empfänger und Absender in diesem Sonderfall nicht notwendig, da die Checkliste der Adressierung dient. In Kapitel 2.2 wird generell keine Adresse für Empfänger sowie Absender angegeben, da immer die gleichen Dokumente an den gleichen Empfänger versendet werden. Somit ist die Adressierung implizit aus dem Bearbeitungszustand der Materialien gegeben.

Wurde eine Sendung adressiert, kann sie in dem Postausgangskorb abgelegt werden. Dort wird sie später von einem Boten entnommen und dem Empfänger zugestellt, indem sie in den entsprechenden Posteingangskorb hineingelegt wird.

Der Sonderfall des „intelligenten“ Boten deutet weitere mögliche Kooperationsformen an, bei denen z.B. eine explizite Adressierung der Sendungen entfallen kann. Im folgenden wird dieser Fall nicht weiter diskutiert, da er den Rahmen dieser Arbeit sprengen würde.

2.4 Zusammenfassung

In diesem Abschnitt wird noch einmal zusammenfassend festgehalten, welche Fakten für die Konzeptionierung des Transportsystems von Bedeutung sind. Außerdem wird eine einheitliche

Begriffsbildung gefunden, da in den einzelnen Szenarios unterschiedliche Wörter für die prinzipiell gleichen Sachverhalte verwendet wurden.

Die folgende Abbildung gibt einen Überblick aller Komponenten und findet eine Zuordnung zu den bisher verwendeten Begriffen.

Name der Komponente	Begriffe aus Kap. 2.1	Begriffe aus Kap. 2.2
Adresse	Adresse bzw. ergibt sich aus der Checkliste	ergibt sich aus dem Bearbeitungszustand und Ort der Materialien
Material	Dokument, Formular	Antragsformular, BOS, Blutprobenröhrchen
Transportmappe	Mappe	implizit durch die Materialien selbst gegeben, aufgrund der geringen Vielfalt
Posteingangskorb, Postausgangskorb	Posteingangskorb, Postausgangskorb	Arztkorb, Stationskorb, Ablage, Postfach
Bote	Postbote, evtl. Mitarbeiter	Bote, evtl. Bearbeiter

Tabelle 1 : Zuordnung verwendeter Begriffe zu den Komponenten eines Transportsystems

Mit diesem einfachen Begriffsgerüst läßt sich das Prinzip des Transportsystems folgendermaßen beschreiben.

Die mit je einer Adresse für Empfänger sowie Absender beschriftete Transportmappe wird mit den zu sendenden Materialien versehen und in einen Postausgangskorb abgelegt. Ein Bote entnimmt automatisch die Transportmappe aus dem Postausgangskorb und transportiert sie zum Empfänger. Dort legt er sie in den Posteingangskorb ab. Der Empfänger kann aus seinem Posteingangskorb die Transportmappe entnehmen und auf die darin befindlichen Materialien zugreifen.

Dieses allgemeine und grobe Prinzip dient als Ausgangspunkt für die Konzeptionierung eines computergestützten Transportsystems in dem folgenden Kapitel.

3 Konzeption des Transportsystems

In dem vorangegangenen Kapitel wurde in den Szenarios schon grob der Aufbau sowie die Verwendung eines Transportsystems deutlich. In diesem Kapitel wird nun das Konzept für die Unterstützung der Kooperation erstellt. Dabei sind nicht nur die Erkenntnisse aus den Szenarios berücksichtigt sondern zusätzlich auch die Erfahrungen, die der Verfasser in seiner Arbeit an dem Arbeitsbereich Softwaretechnik der Universität Hamburg sammeln konnte.

Das folgende Konzept hat als Leitbild den Arbeitsplatz für eigenverantwortliche qualifizierte Tätigkeit. Als Kontext dient die WAM-Metapher, welche auf diesem Leitbild basiert. Diese Metapher wird hier als bekannt vorausgesetzt. Dazu sei der Leser auf die entsprechende Literatur verwiesen, wie z.B. [RZ95], [GKZ94], [WAM98] sowie auf weitere an dem Arbeitsbereich Softwaretechnik der Universität Hamburg erstellte Studien- und Diplomarbeiten.

In dem Kapitel 3.1 wird grob im Überblick der Aufbau des Transportsystems dargestellt. Es wird erläutert, aus welchen Komponenten das Transportsystem besteht und in welcher Beziehung sie zueinander stehen.

In dem anschließenden Kapitel 3.2 werden die Komponenten im einzelnen genauer erörtert. Im Abschnitt 4 wird das Konzept des Verteilungsaspektes diskutiert, ohne den ein sinnvoller Einsatz des Transportsystems nicht möglich wäre.

Das Kapitel 3.3 beschreibt das Zusammenspiel der Komponenten, um die Abhängigkeiten und die Funktionsweisen zu verdeutlichen.

3.1 Aufbau

Wie aus den Szenarios ersichtlich wurde, stand im Mittelpunkt der Kooperation der Versand von Materialien. Diese Materialien werden in Transportmappen abgelegt, um sie mittels Versandautomat an den Empfänger zu überstellen. Dabei ist es möglich, mehr als nur ein Material in einer Transportmappe abzulegen. Dieser Sachverhalt wird schematisch in den folgenden Abbildungen dargestellt. Dabei wird eine Unterteilung in die Sicht des Anwenders und der des Versandautomaten vorgenommen. Doch nun zuerst zum Aufbau des Transportsystems aus der Sicht des Anwenders (siehe Abbildung 2).

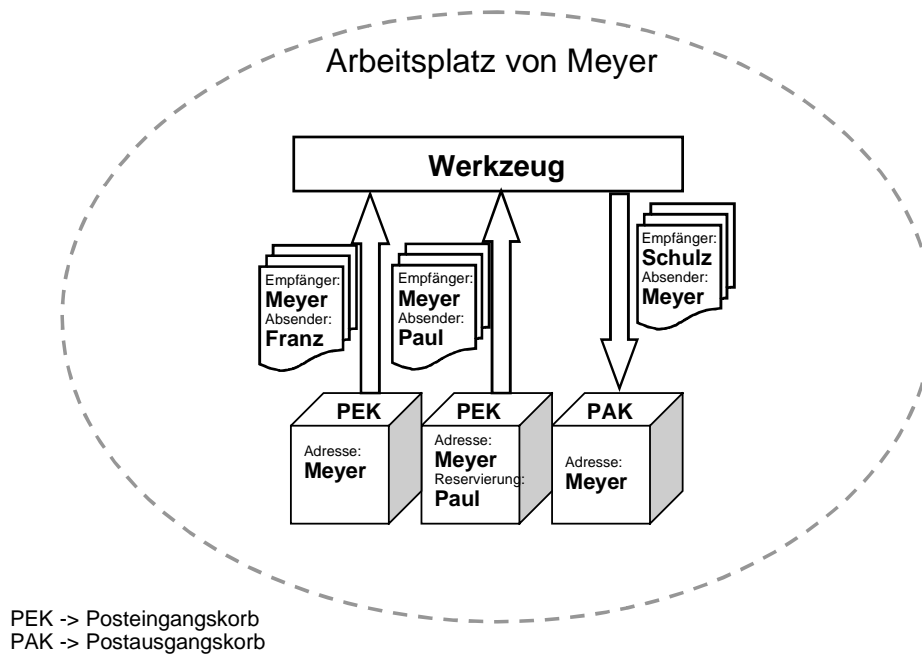


Abbildung 2: Aufbau des Transportsystems aus der Sicht der Anwenders

Jeder der Beteiligten im System besitzt seinen eigenen Arbeitsplatz. Auf diesem Arbeitsplatz befinden sich Postein- und Postausgangskörbe, welche zur Aufnahme der zu sendenden bzw. empfangenen Transportmappen dienen. Diese Postkörbe werden evtl. mit einer Reservierung versehen und dem Versandautomaten im Kontext des konkreten Arbeitsplatzes bekannt gegeben. Die Bedeutung der Reservierung ist beim Postein- und Postausgangskorb verschieden. Handelt es sich um einen Posteingangskorb, so dient die Reservierung der Selektion der empfangenen Transportmappen nach Absender. Ist der Postkorb dagegen ein Postausgangskorb, so wird die Reservierung zur Selektion nach Empfängern verwendet. Dabei können sich auf jedem Arbeitsplatz beliebig viele Postkörbe befinden. Das ist konkret bei mehreren Posteingangskörben mit Reservierung sinnvoll, um die bereits erwähnte Selektion vorzunehmen.

Die Sicht des Versandautomaten auf die Arbeitsplätze zeigt die folgende Abbildung.

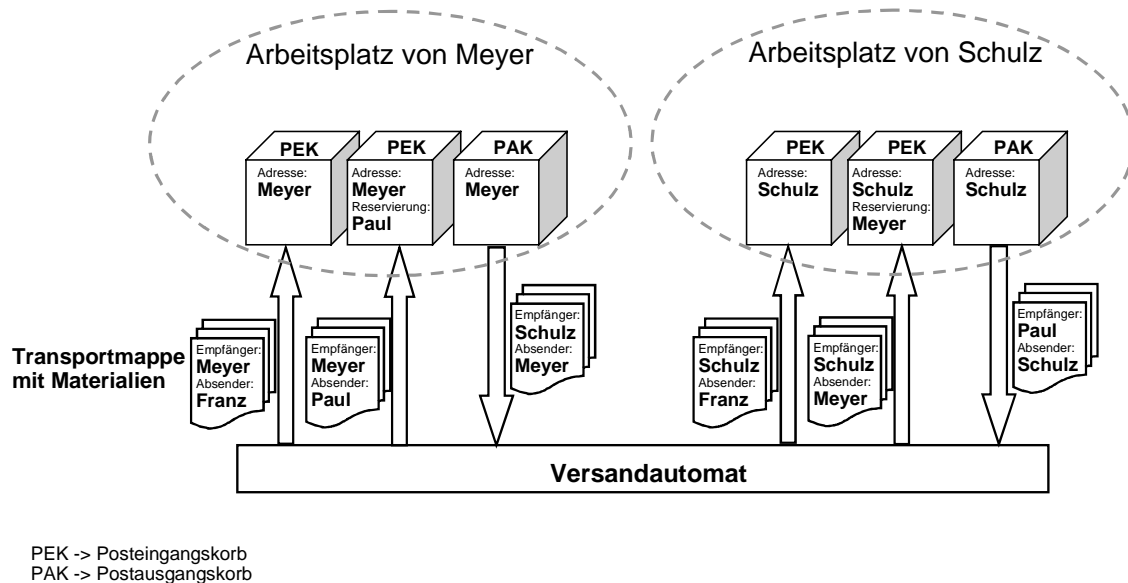


Abbildung 3: Sicht des Versandautomaten auf die Arbeitsplätze der Anwender

Der Versandautomat kennt alle im System verfügbaren Postkörbe und stellt automatisch die Transportmappen zu.

Ein Gruppenarbeitsplatz wird bei diesem Konzept nicht ausgeschlossen. In diesem Fall würde die Gruppe als eine Einheit auftreten, die im Kontext eines gemeinsamen Arbeitsplatzes ihre Arbeit verrichtet. Sie würde aus der Sicht des Transportsystems einen Empfänger darstellen und wäre somit konkret adressierbar. Als Beispiel dafür sei die Station eines Krankenhauses erwähnt.

Wie ersichtlich ist, sind bei diesem Transportsystem zusammenfassend folgende Komponenten enthalten:

- Adresse
- Material
- Transportmappe
- Postkorb
- Versandautomat

3.2 Fachliche Komponenten

Die im vorhergehenden Kapitel genannten Komponenten werden nun detailliert erörtert, um deren fachliche Bedeutung klar herauszuarbeiten.

3.2.1 Adresse

Die Adresse dient als Identifikator der Beteiligten im gesamten System und wird von mehreren Komponenten benutzt. So wird sie verwendet als:

- Empfängeradresse der Transportmappe
- Absenderadresse der Transportmappe
- Adresse des Postkorbes
 - Empfängeradresse bei Posteingangskorb
 - Absenderadresse bei Postausgangskorb

Da die Adresse zur Identifizierung dient, muß sie eindeutig für jeden Beteiligten im gesamten System sein. Um das sicherzustellen, muß ein weiterer Dienst angeboten werden, der Auskunft darüber gibt, ob eine Adresse bereits verwendet wird. Dieser Dienst kann separat als auch in dem Transportsystem integriert angeboten werden. Letzteres empfiehlt sich, wenn die Adressen nur vom Transportsystem verwendet werden. Die erste Variante ist dagegen besser geeignet, wenn die Adresse zur Identifikation von mehreren voneinander unabhängigen Systemen, z.B. Archivierungssystem, verwendet wird.

Von großer Bedeutung ist, daß die Adresse einen Fachlichen Wert (vgl. [JeSh]) darstellt. Dieser Sachverhalt ist sowohl fachlich als auch technisch motiviert. Aus fachlicher Sicht ist die Adresse ein elementarer Wert, welcher der Identifizierung im Transportsystem dient. Hierbei ist jedoch nicht das Exemplar selbst als viel mehr nur dessen Wert von Bedeutung. Somit ist die Adresse als Fachlicher Wert motiviert. Aus technischen Gründen empfiehlt es sich ebenfalls, eine Adresse als Wert zu behandeln, da sie von sehr einfacher Struktur ist und es den Umgang wesentlich vereinfacht. Weiterhin werden Seiteneffekte vermieden, wie sie aufgrund von Referenzsemantik auftreten können.

Adresse
erzeugen
gib Wert
ist gleich ?

Abbildung 4: Umgangsformen der Adresse

Zuerst muß ein Exemplar einer Adresse erzeugt werden. Dazu werden der erzeugenden Methode die Initialwerte übergeben. Danach ist es nicht mehr möglich, den Zustand dieses Exemplars zu ändern. Um eine Adresse auch visuell darstellen zu können, kann deren Wert erfragt werden. Ansonsten ist es nur möglich, das Exemplar einer Adresse auf Wertgleichheit mit einem anderen zu prüfen.

3.2.2 Material

Wie aus den Vorüberlegungen ersichtlich wurde, dient das Transportsystem dazu, Materialien zwischen den Arbeitsplätzen auszutauschen. Deshalb versteht man unter einem Material im engeren Kontext des Transportsystems alles, was versendet werden soll. Es handelt sich nach wie vor um das gleiche Leitbild, wie es von der WAM-Metapher her bekannt ist. Die Besonderheit besteht darin, daß nicht zustands- sondern ortsverändernd auf dem Material gearbeitet wird.

Damit ein Material auch versendet werden kann, muß es die Eigenschaft haben, in einer Transportmappe (vgl. Abschnitt 3.2.3) ablegbar zu sein. Dieser Aspekt ist sowohl fachlich als auch technisch motiviert.

Aus der fachlichen Sicht kann nur etwas versendet werden, das sich in einer Transportmappe befindet. Zwar sind in den beschriebenen Anwendungsfällen (siehe Kapitel 2) alle Materialien von sich aus schon prinzipiell in einer Transportmappe ablegbar gewesen bzw. benötigen keine explizit, doch gibt es andere Fälle, in denen das nicht gewünscht ist. Das ist z.B. insbesondere dann der Fall, wenn es sich um Materialien handelt, die an einen speziellen Ort gebunden sein sollen (z.B. verschiedene Originale in einem Archiv, die nur als Kopie zugreifbar sind).

Hauptsächlich ist jedoch der Aspekt, daß ein Material zum Versand in eine Transportmappe ablegbar sein muß, technisch motiviert. Es gibt dafür zwei Gründe. Zum einen müssen Objekte, die über Prozeßgrenzen hinweg ausgetauscht werden sollen, serialisiert werden und zum anderen wird eine allgemeine Schnittstelle für die Transportmappe benötigt, über die auf die verschiedenen Materialien zugegriffen werden kann.

Die Umgangsform der Serialisierung (vgl. [Zühlke98]) ist zwingend notwendig, um Objekte über ein Netzwerk versenden zu können. Unter Serialisierung versteht man das Umwandeln eines Objektes in einen Stream. Nur in Form eines Streams können Daten über ein Netzwerk transportiert werden.

Damit die Transportmappe die verschiedenen Ausprägungen von Materialien aufnehmen kann, muß eine einheitliche Schnittstelle geschaffen werden. Das kann in Form einer Oberklasse geschehen. Diese Oberklasse deklariert bzw. implementiert die bereits erwähnte Umgangsform der Serialisierung. Es gibt nun zwei Möglichkeiten, diese Oberklasse umzusetzen. Als erstes ist eine allgemeine Oberklasse für alle Materialien im System möglich (siehe Abbildung 5). Wird nun eine Klasse von dieser Oberklasse **Material** abgeleitet, so besitzt sie die Fähigkeit, vom Transportsystem versendet zu werden.

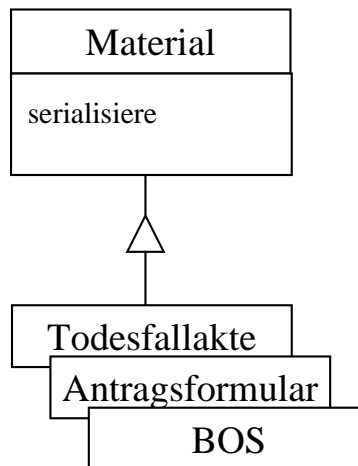


Abbildung 5: Umgangsform der Klasse Material

Als zweite Variante kommt in Frage, die Oberklasse als eine Aspektklasse `InTransportmappeAblegbar` (siehe Abbildung 6) zu realisieren. Das ist besonders dann von Vorteil, wenn solch ein Transportsystem nicht im „leeren Raum“ steht und in eine bereits existierende Softwareumgebung integriert werden soll. Außerdem gewinnt man damit die Flexibilität, nur ausgewählte Materialien mit diesem Aspekt zu versehen.

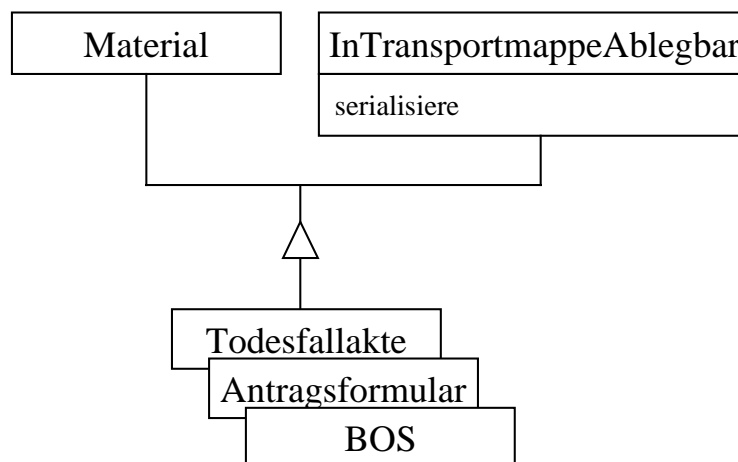


Abbildung 6: Umgangsform der Klasse Transportierbar

3.2.3 Transportmappe

Die Transportmappe ist ein fachlicher Behälter, der mittels Transportsystem versendet werden kann.

Sie dient dazu, alle zu sendenden Materialien aufzunehmen und zusätzlich für den Versandautomaten wichtige Informationen wie Empfängeradresse, Absender usw.

bereitzuhalten. Die sich daraus ergebenden Umgangsformen sind aus der folgenden Abbildung ersichtlich.

Transportmappe
Adresse setzen / erfragen Absender setzen / erfragen Beschriftung setzen /erfragen Inhaltsverzeichnis erfragen Priorität setzen / erfragen Material ablegen / entfernen / selektieren ist leer ?

Abbildung 7: Umgangsformen einer Transportmappe

Von grundlegender Bedeutung ist, daß sich auf der Transportmappe die Adresse des Empfängers erkennen läßt. Der Benutzer muß in der Lage sein, das entsprechende Adreßfeld auszufüllen. Zusätzlich dazu ist es sinnvoll, noch einen Absender anzugeben. Somit besteht im späteren die Möglichkeit, Rückfragen zu stellen bzw. Rückantworten zu senden. Damit der Empfänger nach Erhalt der Transportmappe erkennen kann, um was es sich hierbei handelt, ist es erforderlich, diese zu beschriften. Welchen Inhalt diese Beschriftung hat, kann der Sender selbst entscheiden. So kann z.B. der Vorgangsname darin vermerkt sein. Da die Beschriftung als einzige Inhaltsangabe oftmals nicht ausreichend ist, gibt es zusätzlich noch ein Inhaltsverzeichnis aller Materialien. Dieses vermittelt den Anwender einen Überblick über alle enthaltenen Materialien. Weiterhin kann die Transportmappe mit einer Priorität versehen werden, um die Dringlichkeit der Bearbeitung zum Ausdruck zu bringen. Dazu folgendes Beispiel. Im Abschnitt 2.2 wurde der Arztkorb (Posteingangskorb des Arztes) erwähnt, in dem alle Antragsformulare zur Blutgruppenbestimmung abgelegt werden. Nun sind das nicht alle Dokumente, die dem Arzt zugesendet werden. Somit können sich dort im Laufe eines Tages eine Menge verschiedenartiger Formulare ansammeln. Um gezielt die einzelnen Sendungen zu bearbeiten, bietet sich die Priorität als Selektionskriterium an. Da die Transportmappe dem Versand von Materialien dient, sind die Umgangsformen zum Hineinlegen, Entnehmen und Selektieren vorhanden. Um festzustellen, ob die Transportmappe Materialien beinhaltet, ist zusätzlich eine sondierende Funktion notwendig. Diese ist insbesondere dann von Bedeutung, wenn man nacheinander einzeln die Materialien entnehmen möchte bis die Transportmappe leer ist.

Abschließend werden in der folgenden Abbildung die Objektbeziehungen veranschaulicht.

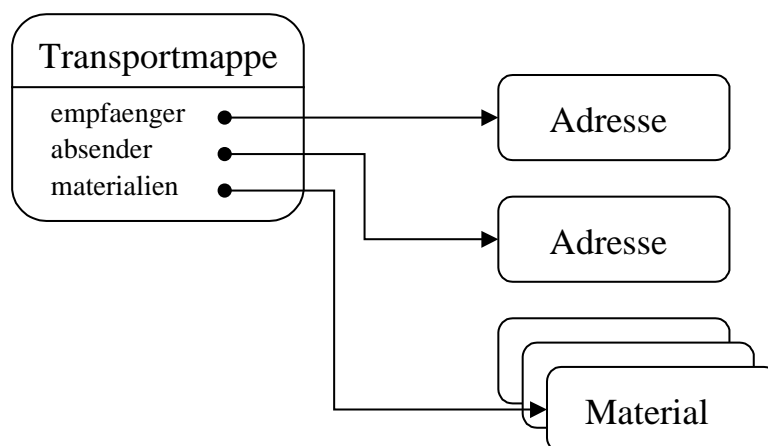


Abbildung 8: Die Objektbeziehungen der Transportmappe

Die Transportmappe bietet mit den genannten Eigenschaften genügend Funktionalität, um den meisten Anforderungen gerecht zu werden.

Hinweise zur Terminologie:

Oftmals wird die Transportmappe in anderen Literaturquellen auch als *Versandmappe* bezeichnet. Dabei ist dieser Begriff in seiner Semantik äquivalent. Eine weitere Bezeichnung für die Transportmappe ist *Vorgangsmappe*. Jedoch wird dieser Begriff mitunter verschieden interpretiert. Zum einen wird unter einer Vorgangsmappe, einen mit einem Prozeßmuster behafteten Transportbehälter verstanden (siehe [Wulf95], Kapitel 4.3.1). Zum anderen wird der Begriff Vorgangsmappe für eine reine Transportmappe, wie sie in dieser Arbeit dargestellt ist, verwendet (vgl. [WAM98]).

3.2.4 Postkorb

Der Postkorb ist die Schnittstelle zwischen Anwender und dem Versandautomat. Er dient zur Auf- bzw. Entnahme von Transportmappen. Ähnlich der Transportmappe, ist er als fachlicher Behälter realisiert.

Es gibt zwei Ausprägungen des Postkorbes, den Posteingangskorb und den Postausgangskorb. Aus der Sicht des Anwenders sind die Umgangsformen (siehe Abbildung 9) identisch. Der Anwender kann an seinen Postein- sowie Postausgangskörben die gleichen Tätigkeiten ausführen, z.B. Transportmappen entnehmen, hineinlegen, das Inhaltsverzeichnis erfragen usw. Dabei spielt es für ihn keine Rolle, als was der Postkorb fungiert. Er muß lediglich über das Kontextwissen verfügen, daß aus dem Postausgangskorb gelegentlich die zu sendenden Transportmappen vom Versandautomat entnommen bzw. in den Posteingangskorb hineingelegt werden.

Postkorb
Adresse setzen / erfragen Beschriftung setzen /erfragen Inhaltsverzeichnis erfragen Reservierung setzen / erfragen Transportmappe ablegen / entfernen / selektieren ist leer ?

Abbildung 9: Umgangsformen des Postkorbes

Der Postkorb befindet sich im Kontext eines konkreten und identifizierbaren Arbeitsplatzes und wird auch in diesem dem Versandautomat bekannt gegeben. Somit wird für die Identifizierung keine explizite Adresse benötigt. Da jedoch der Entwicklungsstand des JWAM zum Zeitpunkt der Erstellung dieser Arbeit das Leitbild eines identifizierbaren Arbeitsplatzes nicht unterstützte, war die Notwendigkeit einer Adresse für den Postkorb gegeben. Diese ist kontextsensitiv und hat zweierlei Bedeutung. Ist der Postkorb als Posteingangskorb beim Versandautomat registriert, so wird die Adresse als Empfängeradresse interpretiert. Ist der Postkorb dagegen ein Postausgangskorb, so wird die Adresse als Absender der darin befindlichen Transportmappen verwendet. Das scheint auf den ersten Blick für Verwirrung zu sorgen, ist aber in der praktischen Umsetzung völlig unproblematisch, da die Adressen der Posteingangskörbe sowie der Postausgangskörbe gleich sind. Das ist darin begründet, daß die Adresse den Arbeitsplatz des Anwenders identifiziert und somit für alle dort vorhandenen Postkörbe gleichermaßen relevant ist.

Weiterhin kann der Postkorb mit einer Beschriftung versehen werden. Damit ist es dem Benutzer möglich, textuell der Verwendung Ausdruck zu verleihen. Insbesondere evtl. Reservierungen sollten hier erwähnt werden.

Da der Postkorb ein fachlicher Behälter ist, ist ein automatisch geführtes Inhaltsverzeichnis vorhanden. Damit ist ein schneller Überblick über alle abgelegten Transportmappen möglich. Von besonderer Bedeutung ist das Reservieren eines Postkorbes für eine konkrete Adresse. Die Semantik der Reservierung ist vom Verwendungskontext abhängig. Handelt es sich um einen Posteingangskorb, so werden in diesem nur die Transportmappen vom Versandautomat zugestellt, deren Absender mit der Reservierung identisch sind. Ist es dagegen ein Postausgangskorb, so hat die Reservierung keine Bedeutung für den Versandautomat.

Da der Postkorb der Verwaltung der Transportmappen dient, sind die Umgangsformen zum Hineinlegen, Entnehmen und Selektieren vorhanden. Um festzustellen, ob der Postkorb Transportmappen beinhaltet, ist zusätzlich eine sondierende Funktion notwendig. Diese ist insbesondere dann von Bedeutung, wenn nacheinander einzeln die Transportmappen entnommen werden sollen, bis der Postkorb leer ist.

Aus der Perspektive des Versandautomaten ist der Umgang mit den zwei Ausprägungen des Postkorbes, dem Postein- und dem Postausgangskorb, grundverschieden. So darf er nur Transportmappen aus den Postausgangskörben entnehmen bzw. sie in die Posteingangskörbe hineinlegen. Um dieser Unterscheidung gerecht zu werden, müssen zwei unterschiedliche Schnittstellen geschaffen werden, die die Umgangsformen für den entsprechenden Zweck

einschränken. Es gibt zwei Möglichkeiten, um diese Schnittstellen zu realisieren. Zum einen mittels Aspektklassen und zum anderen mittels Wrapper (in [GHJ+95] als Decorator beschrieben). Da diese Problematik bereits in [Wulf95] ausführlich erörtert wurde, soll hier nur kurz darauf eingegangen werden.

Bei der Verwendung von Aspektklassen (siehe Abbildung 10) wird in der statischen Vererbungsbeziehung nicht deutlich, daß die Aspekte **entleerbar** (für den Postausgangskorb) bzw. **füllbar** (für den Posteingangskorb) alternativ vom Versandautomat verwendet werden sollen. Es besteht somit mittels Typumwandlung auch weiterhin die Möglichkeit, uneingeschränkt auf alle Umgangsformen des Postkorbes zuzugreifen.

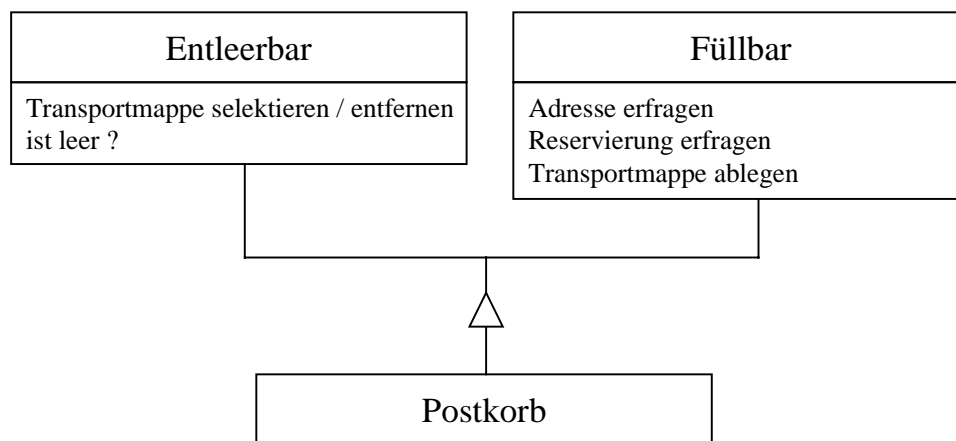


Abbildung 10: Verwendung von Aspektklassen zur Einschränkung der Umgangsformen des Postkorbes gegenüber dem Versandautomat

Anders dagegen bei der Einführung sogenannter Wrapper (siehe Abbildung 11). Hiermit können die bei der Verwendung von Aspektklassen genannten Nachteile vermieden werden. Diese dynamische Bindung bzw. Einschränkung von Umgangsformen des Postkorbes verhindert den zweckentfremdeten Zugriff auf die volle Funktionalität mittels Typumwandlung. Weiterhin ist auch keine Irreführung in der statischen Klassenbeziehung mehr vorhanden.

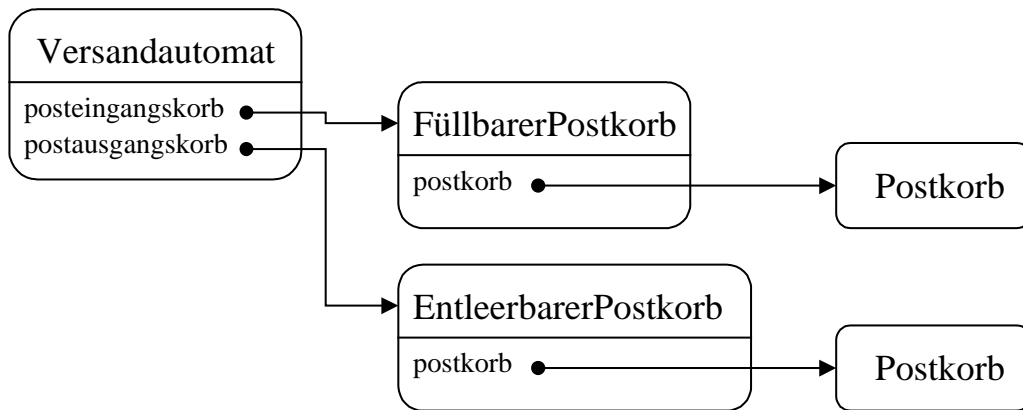


Abbildung 11: Verwendung eines Wrapper zur Einschränkung der Umgangsformen des Postkorbes gegenüber dem Versandautomaten

3.2.5 Versandautomat

Der Versandautomat ist ein Automat, wie er im Kontext der WAM-Metapher (vgl. [Gry95]) verstanden wird.

Der Versandautomat dient dazu, die in den Postausgangskörben abgelegten Transportmappen zu entnehmen, zu den entsprechenden Posteingangskörben zu transportieren und dort abzulegen. Dazu wird in den folgenden Schritten verfahren.

1. *Voraussetzung*
Um den Versandautomat verwenden zu können, müssen vorher die Postkörbe registriert worden sein. Aus diesen entnimmt später der Versandautomat die Transportmappen bzw. legt die empfangenen dort ab.
2. *Aktivierung des Versandautomaten*
Damit der Vorgang zum Versand aller in den Posteingangskörben befindlichen Transportmappen gestartet werden kann, muß der Versandautomat unter Angabe des eigenen Arbeitsplatzes in Form einer Adresse aktiviert werden.
3. *Entnahme der Transportmappen*
Der Versandautomat entnimmt alle Transportmappen aus den Postkörben, die als Postausgangskörbe unter der beim Start genannten Adresse registriert sind, und speichert sie in einem internen Puffer.
4. *Zuordnung der Transportmappen*
Der Versandautomat iteriert über alle gepufferten Transportmappen und versucht eine Zuordnung zu den registrierten Posteingangskörben zu finden. Dazu liest er die Empfängeradressen der Transportmappen und vergleicht sie mit den Adressen der Postkörbe.
5. *Zustellung der Transportmappen*
Ist zu einer gepufferten Transportmappe der entsprechende Posteingangskorb gefunden, so

wird diese aus dem internen Zwischenspeicher entfernt und dort hineingelegt.

Der Umstand, daß der hier beschriebene Vorgang erst gestartet werden kann, entspricht nicht den beschriebenen Anwendungsfällen aus Kapitel 2. Es gibt aber eine Reihe von Gründen für einen manuellen Start, die sowohl fachlich als auch technisch motiviert sind. Diese wären:

- *definierter Zeitpunkt der Zustellung (fachlicher Aspekt)*
Dieser Vorteil ist besonders dann offensichtlich, wenn eine sehr dringliche Transportmappe versendet werden soll. Der Anwender hätte sonst keine Möglichkeit, den sofortigen Versand zu veranlassen. Dieser Umstand kann zu nicht hinnehmbaren Verzögerungen führen.
- *Vermeidung von unnötiger Kapazitätsbelastungen (technischer Aspekt)*
Durch den manuellen Start und der Angabe des Arbeitsplatzes wird nur selektiv der Transport vorgenommen. Im anderen Fall müßte der Versandautomat in regelmäßigen Abständen alle Postausgangskörbe abfragen, ob etwas zu versenden ist. Dieses Polling beansprucht unnötig die vorhanden Ressourcen und kann besonders bei vielen Postausgangskörben zu vermeidbaren Kapazitätsengpässen führen.

Ein Argument zur automatischen Aktivierung des Versandautomaten sei hier noch erwähnt. Es kann vorkommen, daß die Anwender vergessen, die Versendung ihrer Post zu veranlassen, obwohl dies nach einer gewissen Zeit dingend notwendig wäre. Würde der Versandautomat automatisch in periodischen Zeitintervallen den Transport vornehmen, könnten solche kritischen Fälle vermieden werden.

Das Optimum zur Lösung des hier geschilderten Problems besteht aus einer Mischung aus manueller sowie automatischer Aktivierung. Generell wird der Versandautomat automatisch aktiv. Allerdings wird das Zeitintervall größer gewählt, als es bei einer rein automatischen Variante notwendig wäre, um die damit verbundenen zusätzlichen Kapazitätsbelastungen in Grenzen zu halten. Dafür hat aber der Anwender jederzeit die Möglichkeit, die Aktivierung manuell zu veranlassen.

Ein weiteres Problem ist, wie sich der Versandautomat verhalten soll, wenn kein entsprechender Posteingangskorb für eine Transportmappe im Moment vorhanden ist. Für diesen Fall wäre es sinnvoll, daß nicht nur momentan vorhandene Postein- und Postausgangskörbe beim Versandautomaten registriert werden, sondern auch alle potentiellen Arbeitsplätze. Wenn in dieser Ausbaustufe kein Posteingangskorb für eine Transportmappe vorhanden ist, so wird geprüft, ob die Mappe für einen potentiellen Arbeitsplatz vorgesehen ist. Kann bei der Prüfung eine Zuordnung gefunden werden, so wird die Transportmappe solange zwischengespeichert, bis ein entsprechender Posteingangskorb angemeldet wird und anschließend zugestellt. Kann jedoch kein registrierter Arbeitsplatz gefunden werden, so ist die Empfängeradresse offensichtlich nicht korrekt und die Mappe wird zurückgesendet.

Eine weniger komfortable Lösung wurde bei der Realisierung des Transportsystems gewählt, da wie zuvor bereits erwähnt, das Framework JWAM keine identifizierbaren Arbeitsplätze unterstützte. In diesem Fall wird die Mappe vorerst eine gewisse Zeit lang zwischengespeichert. Wird in diesem Zeitraum kein entsprechender Posteingangskorb bei dem Versandautomat registriert, so wird die Mappe an den Absender zurück gesendet.

Die Umgangsformen sind in Abbildung 12 ersichtlich.

Versandautomat
Versand durchführen als Posteingangskorb registrieren Posteingangskorb abmelden als Postausgangskorb registrieren Postausgangskorb abmelden für Ereignis Neuzugang registrieren von Ereignis Neuzugang abmelden

Abbildung 12: Umgangsformen des Versandautomaten

Wie bereits erwähnt, muß eine Möglichkeit bestehen, den Vorgangsautomat zu aktivieren. Damit dem Automat alle Postkörbe bekannt sind, die er zu bedienen hat, müssen diese erst registriert werden. Es gibt verschiedene Registrierungsarten getrennt nach Postein- und Postausgangskörben. Soll ein Postkorb nicht mehr bedient werden, so kann er wieder abgemeldet werden. Das sollte insbesondere dann erfolgen, bevor ein Postkorb entfernt bzw. vernichtet wird. Weiterhin ist es sinnvoll, daß man informiert wird, wenn neue Transportmappen in einen Posteingangskorb abgelegt worden sind. Dazu kann sich das Werkzeug für das Ereignis Neuzugang anmelden. Auch hier ist es wieder notwendig, das Abmelden zu ermöglichen.

3.3 Zusammenspiel der Komponenten

In diesem Abschnitt werden die Bedeutungen der einzelnen Komponenten und ihr Zusammenspiel verdeutlicht. Dabei sind auch die Abhängigkeiten und die Benutzt-Beziehungen von Interesse.

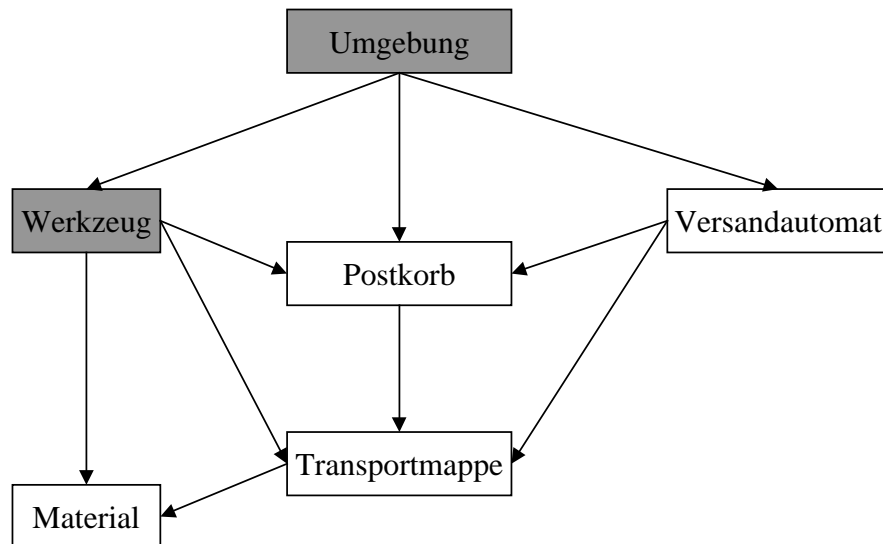


Abbildung 13: Benutzt-Beziehungen aller Komponenten des Transportsystems (Teil 1)

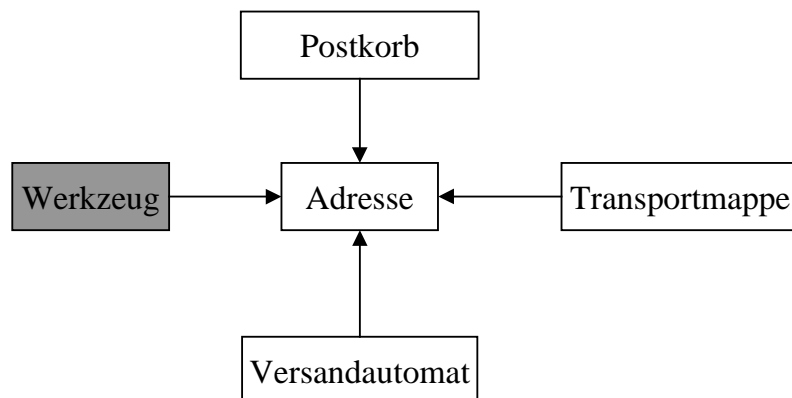


Abbildung 14: Benutzt-Beziehungen aller Komponenten des Transportsystems (Teil 2)

Die Abbildung 13 und Abbildung 14 zeigen die Abhängigkeiten der Komponenten und des einbettenden Kontextes (aus Gründen der Übersichtlichkeit wurden alle Abhängigkeiten in zwei Darstellungen abgebildet). Um diese Verflechtungen verständlicher erläutern zu können, wird nun der Transport einer Transportmappe unter Beteiligung aller Komponenten skizziert.

1. *Voraussetzungen für den Versand*

Bevor Transportmappen versendet werden können, muß das Transportsystem initialisiert worden sein. Darunter ist zu verstehen, daß der **Versandautomat** eingerichtet und funktionsbereit sein muß. Weiterhin müssen alle benötigten Postkörbe des lokalen Arbeitsplatzes von der **Umgebung** erzeugt und beim **Versandautomat** registriert worden sein.

2. *Der Versand*

Ist das Transportsystem initialisiert, können Transportmappen versendet werden. Das

geschieht in den folgenden Schritten.

- Ein **Werkzeug** legt das bearbeitete **Material** in einer **Transportmappe** ab. Diese muß bereits existieren oder wird gegebenenfalls erzeugt. Weiterhin muß diese **Transportmappe** mit einer **Adresse** des Empfängers sowie einen Absender beschriftet werden.
- Danach kann sie in den **Postkorb** für Postausgänge gelegt werden.
- Entschließt sich der Anwender, die Post abzusenden, so wird der lokale **Versandautomat** aktiviert. Dieser entnimmt die **Transportmappe** aus dem Postausgangskorb des Senders und legt sie in den entsprechenden Posteingangskorb des Empfängers ab. Anschließend löst er das Ereignis für den Neuzugang aus, damit die **Werkzeuge**, die die Posteingangskörbe beobachten, in der Lage sind, ihre Darstellungen zu aktualisieren.

4 Architektur der Verteilungskomponente

In diesem Kapitel wird der wesentlichste Aspekt erörtert, nämlich der der Verteilung. Das bisher beschriebene Konzept erfüllt noch nicht den beabsichtigten Zweck, wenn es nicht möglich ist, über Prozeßgrenzen hinweg die Transportmappen zu versenden.

Im Kapitel 4.1 wird zuerst ein Architekturprinzip vorgestellt, daß sich bei der Realisierung verteilter Systeme allgemein durchgesetzt hat. Im anschließenden Kapitel 4.2 wird darauf aufbauend eine Konzeption der Verteilungskomponente entworfen. In der Konzeption wird die Notwendigkeit zweier weiterer Konzepte für Realisierung deutlich. Diese sind die Remote Methode Invocation (RMI) und die Nachrichtenvermittlung. RMI stellt die Grundlage aller verteilten Anwendungen in Java dar und wird deshalb zuerst in Kapitel 4.3 beschrieben. Die Nachrichtenvermittlung ist von wesentlicher Bedeutung für das gesamte Konzept der Verteilungskomponente des Transportsystems. Sie basiert ebenfalls auf RMI und wird in Kapitel 4.4 erläutert. Mit dem Wissen über diese beiden Konzepte wird dann die Realisierung der Verteilungskomponente in Kapitel 4.5 beschrieben.

4.1 Dienstleistungsprinzip

Wie in den in Kapitel 2 beschriebenen Anwendungsfällen deutlich wird, sind eine Vielzahl von Personen in einen arbeitsteiligen Vorgang involviert. Um die dabei notwendige Kooperation mit Hilfe von Computern zu unterstützen, ist es erforderlich, Materialien zwischen den verschiedenen (Computer-)Arbeitsplätzen auszutauschen. Dazu ist es notwendig, eine Architektur zu entwickeln, die einen Informationsfluß über Rechnergrenzen hinweg ermöglicht.

Ein weit verbreiteter Ansatz in der objektorientierten Softwareentwicklung ist das Dienstleistungsprinzip (vgl. [GKZ94] Kapitel 3). Es unterteilt die Komponenten eines Systems in Leistungsanbieter und Leistungsabnehmer. Die Leistungsanbieter stellen an ihrer Schnittstelle verschiedene Dienstleistungen zur Verfügung. Dabei sind für die Dienstleistungen folgende Eigenschaften charakteristisch:

- sie sind *implementationsunabhängig*, d.h. die konkrete Realisierung bleibt dem Leistungsabnehmer an der Schnittstelle verborgen
- sie sind *ortsunabhängig*, d.h. die Dienstleistung ist nicht an einen Ort (Computer) gebunden

Die Leistungsabnehmer bedienen sich der angebotenen Dienstleistungen, um ihre Aufgaben zu erfüllen. Da dieses Konzept keine Annahmen über die Anzahl der Leistungsabnehmer trifft bzw. diese festlegt, ist die Skalierung derer im System möglich. Damit ist eine weitere grundlegende Eigenschaft gegeben, um Kooperation rechentechnisch zu unterstützen, da die Anzahl der Kooperationspartner weder festgelegt noch eingeschränkt werden sollte.

Mit diesem Prinzip ist es möglich, ein System zu entwickeln, das über eine Vielzahl von Computern verteilt arbeitet und als Basis für eine Kooperationsunterstützung dient.

4.2 Konzeption der Verteilungskomponente

Um die Verteilungskomponente zu modellieren, bietet sich das bereits vorgestellte Dienstleistungsprinzip (siehe Kapitel 4.1) an. Dazu wird der Versandautomat in zwei Komponenten zerlegt, in den Versandautomat-Client (Leistungsabnehmer) und in den Versandautomat-Server (Leistungsanbieter). Einen Überblick über den Aufbau zeigt Abbildung 15. Wie zu erkennen ist, gibt es genau einen Versandautomat-Server und mehrere Versandautomat-Clients. Dabei ist deren Anzahl nicht festgelegt oder beschränkt. Der Server hat die Aufgabe, Transportmappen zwischen den verschiedenen Prozeßräumen auszutauschen. Aufgrund seiner prozeßraumübergreifenden Funktion arbeitet der Server in einem separaten Server-Prozeß. Dieser Server-Prozeß ist so zentral gelegen, daß er von allen Clients aus zugreifbar ist.

Bei dieser Architektur läge es nahe, bei den Clients sogenannte Proxies (siehe [GHJ+95]) des Servers zu vermuten. Bei dem hier skizzierten Konzept sind die Clients selbständige Automaten, die sich lediglich der Dienste des Servers bedienen, um Transportmappen über Prozeßgrenzen hinweg zu den anderen kooperierenden Versandautomat-Clients versenden zu können. Die Aufgaben der Versandautomat-Clients sind folgende:

- alle Postkörbe des lokalen Arbeitsplatzes zu verwalten
- die zu sendenden Transportmappen aus den Postausgangskörben zu entnehmen und mittels Versandautomat-Server zu versenden
- auf die Nachricht des Servers über den Neuzugang von Transportmappen zu reagieren, indem die Mappen vom Server abgeholt werden
- die abgeholt Mappen den entsprechenden Posteingangskörben (gemäß evtl. Reservierungen) zuzuteilen

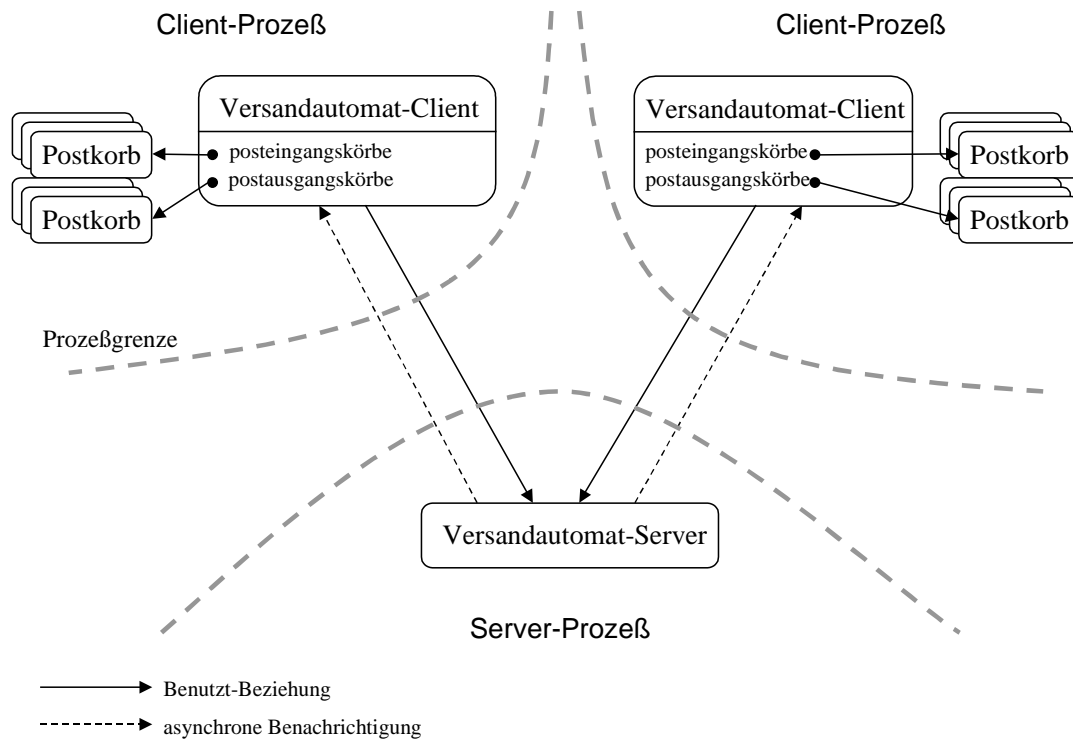


Abbildung 15: Konzept der Verteilungskomponente des Transportsystems

Die Funktionsweise dieser Architektur sieht wie folgt aus. Der Versandautomat-Client benutzt den Versandautomat-Server, indem er die angebotenen Dienstleistungen verwendet. Damit ist es unabdingbar, daß er die volle Schnittstelle des Servers kennt. Da sich der Versandautomat-Server in einem anderen Prozeß (Ort) als der Versandautomat-Client befindet, muß er sich eines weiteren Konzeptes bedienen, daß einen Aufruf entfernter Methoden ermöglicht. Java bietet dafür die Remote Methode Invocation (RMI) an, daß im Kapitel 4.3 näher beschrieben wird.

Umgekehrt dagegen kennt der Server das Interface des Client nicht. Damit der Client über den Erhalt neuer Transportmappen vom Server informiert wird, muß jedoch auch in dieser Richtung ein Informationsfluß realisiert werden. Dazu wird der Reaktionsmechanismus der Nachrichtenvermittlung (siehe Kapitel 4.4) verwendet. Diese funktioniert in der Art, daß der Server dem Client eine entsprechende Nachricht sendet, wenn ein bestimmtes Ereignis wie der Neuzugang einer Transportmappe eingetreten ist. Diese Benachrichtigung geschieht asynchron, um eine verzögerte Ausführung des nachrichtenauslösenden Prozesses zu vermeiden. Der benachrichtigte Client greift auf den Versandautomat-Server über die ihm bekannte Schnittstelle zu und erfragt die gesendete Transportmappe.

Der Transport einer Transportmappe geschieht folgendermaßen (siehe Abbildung 16). Der Versandautomat-Client A ruft die dem Versanddienst entsprechenden Methode des Versandautomat-Server auf und übergibt als Parameter die Transportmappe. Daraufhin benachrichtigt der Versandautomat-Server den Versandautomat-Client B, indem er eine entsprechende Nachricht sendet. Der so benachrichtigte Versandautomat-Client B fragt nun seinerseits über eine sondierende Funktion des Versandautomat-Server diese gesendete Transportmappe ab und bekommt sie als Rückgabewert übergeben. Nun kann der

Versandautomat-Client B die erhaltene Transportmappe in den dafür vorgesehenen Posteingangskorb ablegen.

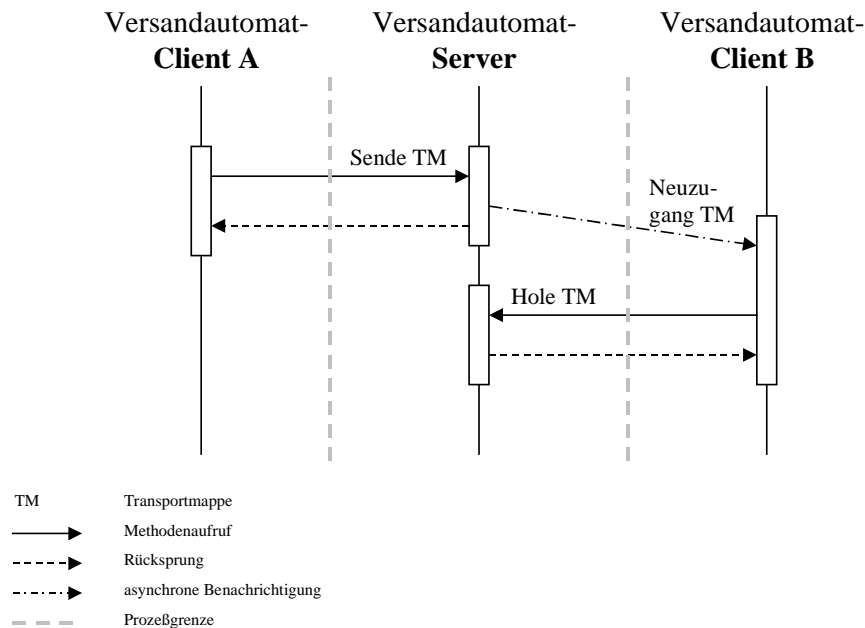


Abbildung 16: Versenden einer Transportmappe

Mit diesem Konzept ist eine Übertragung der Transportmappen über Prozeßgrenzen hinweg realisierbar. Dadurch sind die Kooperationspartner in der Lage, von verschiedenen Orten aus miteinander zu kommunizieren und Materialien auszutauschen.

4.3 Remote Methode Invocation

Seit der Version 1.1 bietet Java das Konzept der Remote Methode Invocation (RMI), das eine einfache Realisierung verteilter Anwendungen erlaubt (vgl. [JS97]).

Das Prinzip dieses Konzeptes funktioniert in der Art, daß Objekte, auf die entfernt zugegriffen werden soll, bei einem Naming-Service registriert und somit global bekannt gemacht werden. Interessiert sich ein Client für ein solches Objekt, so stellt er eine Verbindung über den RMI-Mechanismus her und kann fortan auf dieses zugreifen, als wäre es lokal. Markant hierbei ist, daß alle Aufrufe von Methoden synchron durchgeführt werden, d.h. daß der Client-Prozeß die vollständige Ausführung des entfernten Methodenaufrufes abwarten muß. In der Umgangsweise mit entfernten Objekten gibt es nur geringe Unterschiede zu den lokal vorhandenen, das wären:

1. die Erzeugung

Damit der Client auf ein entferntes Objekt zugreifen kann, muß dieses bereits auf dem Server erzeugt und beim Naming-Service angemeldet worden sein. Die Abbildung 17 zeigt eine Implementationsmöglichkeit. Die Voraussetzungen dafür sind, daß das Server-Objekt ein sogenanntes *Remote Interface* implementiert und von *UnicastRemoteObject*

abgeleitet ist.

```
public class VersandautomatServerImpl
    extends UnicastRemoteObject      // Basisklasse für Server-Objekte
    implements VersandautomatServer // das Remote Interface
{
    ...
}
...
try
{
    // Erzeugung des Server-Objektes
    VersandautomatServerImpl va=new VersandautomatServerImpl();
    // Registrierung des Server-Objektes beim Naming-Service
    Naming.rebind ("WAM-Versandautomat-Server", va);
}
catch (RemoteException e)
{
    ...
}
```

Abbildung 17: Implementation der Erzeugung und Anmeldung eines Objektes auf einem Server

Ist das Server-Objekt erzeugt und angemeldet, so kann der Client eine Verbindung über den RMI-Mechanismus zu diesem aufbauen. Dazu gibt er dem Naming-Service eine URL-Adresse (Uniform Resource Locator) des Servers und des entfernten Objektes an und erhält als Rückgabeparameter einen lokalen Repräsentanten des Server-Objektes (siehe Abbildung 18). Über das darin implementierte *Remote Interface* kann der Client entfernt auf das Server-Objekt zugreifen.

```
try
{
    String serverURL = "rmi://Server/WAM-Versandautomat-Server";
    // Versuchen, den RMI-Server zu finden
    VersandautomatServer va = (VersandautomatServer)Naming.lookup(serverURL);
}
catch (RemoteException e)
{
    ...
}
```

Abbildung 18: Implementation der Anbindung an das Server-Objekt

2. *die Parameterübergabe*

Normalerweise gilt in Java bei den Parametern und den Rückgabewerten Referenzsemantik. Wird im Gegensatz dazu ein entferntes Objekt (*Remote Object*) aufgerufen, so wird Wertsemantik verwendet. Das ist damit begründet, daß Java alle Parameter serialisiert und so über das Netzwerk versendet. Voraussetzung für die Serialisierung ist, daß die Parameter das Interface **Serializable** geerbt haben. Auf der Seite des Servers werden die Daten wieder in Objekte umgewandelt und dem gerufenen Server-Objekt als lokale Kopie übergeben. Bei dem Rückgabewert verhält es sich analog dazu genauso, nur in umgekehrter Richtung. Die Besonderheit der Verwendung der Wertsemantik ist bei der Realisierung verteilter Objekte zu berücksichtigen. Allerdings gibt es auch hier wieder eine Ausnahme. Sind die Parameter bzw. der Rückgabewert selbst wieder Remote Objects, so gilt die Referenzsemantik.

3. *die Fehlermöglichkeiten*

Da die Aufrufe von Methoden entfernter Objekte über das Netzwerk gesendet werden, bestehen hier zusätzliche Fehlermöglichkeiten. Ist das Server-Objekt nicht mehr zugreifbar, so muß dieser Fehler im Programmablauf berücksichtigt werden. Der RMI-Mechanismus bedient sich dazu der Ausnahmebehandlung und löst eine **RemoteException** im Fehlerfall aus. Um auf diese Ausnahmefälle reagieren zu können, müssen alle entfernten Aufrufe in Try-Catch-Blöcke eingefaßt (siehe auch Abbildung 17 und Abbildung 18) werden. Es liegt dann in der Verantwortung des Clients, diese Fehlerfälle angemessen zu bearbeiten.

4. *die Ausführungsgeschwindigkeit.*

Da die Kommunikation über das Netzwerk abgewickelt wird, müssen alle Daten, die für entfernte Methodenaufrufe notwendig sind, in einen Stream umgewandelt und versendet werden. Es ist offensichtlich, daß zur Umwandlung sowie für den Datentransport Ressourcen beansprucht werden. Dadurch kommt es zu merklichen Verzögerung in der Programmausführung, insbesondere auch deshalb, weil keine asynchronen Aufrufe möglich sind. Somit führt jeder entfernte Methodenaufruf beim Client zu einer Unterbrechung der Ausführung, bis der Kontrollfluß zurückgekehrt ist.

Zusammenfassend kann festgehalten werden, daß bei der Verwendung von RMI zur Realisierung verteilter Anwendungen folgende Komponenten benötigt werden:

- ein von **Remote** abgeleitetes Interface, über das auf das Server-Objekt zugegriffen werden kann
- ein Server-Objekt, das von **UnicastRemoteObject** abgeleitet ist und das Interface implementiert

4.4 Nachrichtenvermittlung

Die Nachrichtenvermittlung wurde am Arbeitsbereich Softwaretechnik der Universität Hamburg von Martin Lippert entwickelt und ist Bestandteil des JWAM-Frameworks (**J**ava **W**AM-Framework). Sie ist eine Ausprägung eines Reaktionsmechanismus (vgl. [RoWo]).

In diesem Kapitel wird die Nachrichtenvermittlung näher beschrieben, da sie von grundlegender Bedeutung für die Realisierung der Verteilungskomponente des Transportsystems ist. Dabei ist es nicht das Anliegen, jedes Detail zu beleuchten. Vielmehr

wird ein Verständnis geschaffen, wie sie prinzipiell funktioniert. Dazu wird in Kapitel 4.4.1 kurz der Aufbau der Nachrichtenvermittlung dargestellt, damit die Komponenten deutlich werden, aus denen sie besteht. Anschließend werden in Kapitel 4.4.2 alle beteiligten Komponenten genauer erörtert.

4.4.1 Aufbau

Die Nachrichtenvermittlung operiert im wesentlichen in zwei verschiedenartigen Prozessen, dem Client- und dem Server-Prozeß. Im Client-Prozeß arbeitet der **MessageBroker**, der den Nachrichtenaustausch im lokalen Prozeßraum vornimmt. Ist ihm der **MessageBrokerServer**, welcher seine Dienste im Server-Prozeß anbietet, bekannt, so nutzt er ihn, um Nachrichten auch über Prozeßgrenzen hinweg zu versenden.

Die Abbildung 19 verschafft einen ersten Überblick über den Aufbau sowie weiteren beteiligten Komponenten.

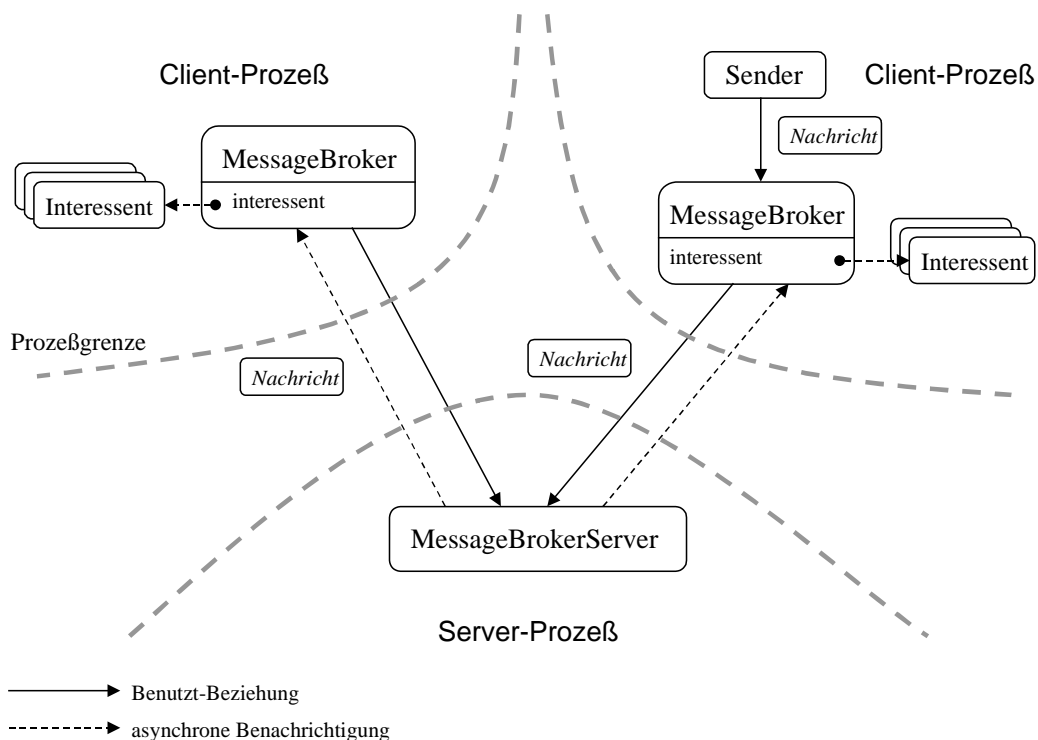


Abbildung 19: Aufbau der Nachrichtenvermittlung

Im folgenden wird die Funktionsweise kurz geschildert.

1. Voraussetzung:

In jedem Prozeßraum wird genau ein Exemplar des **MessageBroker** erzeugt. Dieses prüft, ob auf dem Server der **MessageBrokerServer** gestartet wurde. Ist das der Fall, so registriert der lokale **MessageBroker** sich beim **MessageBrokerServer**, um auch Nachrichten über Prozeßgrenzen hinweg versenden bzw. empfangen zu können. Ansonsten kann keine Registrierung bei dem Server vorgenommen werden und der

Nachrichtenaustausch geschieht nur im lokalen Prozeßraum. Eine sehr bedeutende Eigenschaft der Nachrichtenvermittlung ist, daß sämtliche Nachrichten asynchron versendet werden. Das hat den entscheidenden Vorteil, daß der Prozeß, der eine Nachricht sendet, nicht abwarten muß, bis alle Interessenten die Nachricht empfangen und darauf reagiert haben.

2. *Empfang von Nachrichten:*

Der Interessent meldet sich bei seinem lokalem **MessageBroker** für eine konkrete Nachricht an, indem er ein Klassenobjekt der gewünschten Nachricht sowie ein Objekt, das benachrichtigt werden soll, und einen Methodennamen angibt. Zusätzlich kann eine Klausel angegeben werden, die ein weiteres Filtern der empfangenen Nachrichten ermöglicht. Erhält der lokale **MessageBroker** eine Nachricht, die eine evtl. Klausel erfüllt, so wird bei dem registrierten Objekt die Methode mit dem zuvor übergebenen Methodennamen aufgerufen.

3. *Senden von Nachrichten:*

Der Sender erzeugt ein Exemplar einer Nachricht, das er versenden möchte und übergibt es dem **MessageBroker**.

4. *Transport von Nachrichten:*

Nach Übergabe der zu sendenden Nachricht überprüft der **MessageBroker**, ob es im lokalen Prozeßraum Interessenten gibt und benachrichtigt diese. Ist keine Verbindung zu einem **MessageBrokerServer** vorhanden, so ist der Vorgang beendet, ansonsten wird die Nachricht an diesen weitergeleitet. Der **MessageBrokerServer** verteilt die Nachricht an alle bei ihm dafür registrierten **MessageBroker**, welche die Benachrichtigung der Interessenten im jeweiligen lokalen Prozeßraum vornehmen.

4.4.2 Komponenten

Im vorangegangenen Kapitel wurden die zur Nachrichtenvermittlung gehörenden Komponenten kurz genannt und werden nun im folgenden detailliert erörtert. Die Komponenten sind die Nachricht (Kapitel 4.4.2.1), die Klausel (Kapitel 4.4.2.2), der **MessageBroker** (Kapitel 4.4.2.3) und der **MessageBrokerServer** (Kapitel 4.4.2.4).

4.4.2.1 Nachricht

Eine Nachricht ist ein Objekt, das über den **MessageBroker** versendet werden kann, um Interessenten über Ereignisse zu informieren.

Die allgemeine Oberklasse für alle Nachrichtenklassen ist **msgObjekt**. Um eigene Nachrichtenklassen zu definieren, müssen diese von **msgObjekt** abgeleitet werden. Außerdem ist es auf diesem Wege möglich, eine Nachrichtenklassen mit weiteren Attributen zu versehen, um den Interessenten zusätzliche Informationen zu übermitteln. Es ist jedoch sicherzustellen, daß diese Attribute auch serialisierbar sind, da die Exemplare dieser Klasse serialisiert versendet werden.

4.4.2.2 Klausel

Die Klausel wird vom `MessageBroker` verwendet und dient dem Filtern von typgleichen Nachrichten.

Die allgemeine Oberklasse für alle Klauseln ist `clauseObject`. Um eine spezielle Klausel zu definieren, muß von dieser Oberklasse abgeleitet werden. Weiterhin ist die Methode `isFullfilled()` zu redefinieren, welche die Gültigkeitsbedingung für einen konkreten Nachrichtentyp implementiert.

```
public abstract class clauseObject implements Serializable
{
    public void configure (msgObject obj)
    protected msgObject configureObject ()
    public abstract boolean isFullfilled ();
}
```

Abbildung 20: Schnittstelle clauseObject

Hat sich ein Interessent für eine konkrete Nachrichtenklasse beim `MessageBroker` angemeldet und zusätzlich eine Klausel angegeben, so stellt der `MessageBroker` mit Hilfe der Methode `isFullfilled()` fest, ob eine empfangene Nachricht diesem Selektionskriterium genügt. Wenn ja, dann wird der Interessent benachrichtigt, ansonsten nicht.

4.4.2.3 MessageBroker

Der `MessageBroker` ist der lokale Repräsentant des Nachrichtenvermittlungsdienstes. Er ist als Singleton (vgl. [GHJ+95]) implementiert und existiert somit nur einmal pro Prozeßraum.

```
public class MessageBroker
{
    public static MessageBroker instance()
    public void register (Object recipient, String methodName, Class messageType)
    public void register (Object recipient, String methodName, Class messageType, clauseObject clause)
    public void unregister (Object recipient, String methodName, Class messageType)
    public void unregister (Object recipient, String methodName, Class messageType, clauseObject clause)
    public void sendMessage (msgObject message)
}
```

Abbildung 21: Schnittstelle MessageBroker

Um ein bzw. das Exemplar des `MessageBroker` zu erhalten, wird die Methode `instance()` aufgerufen. An seiner Schnittstelle wird die Funktionalität zur Verfügung gestellt, die notwendig ist, um einen Interessenten für einen konkreten Nachrichtentyp an- bzw.

abzumelden. Dazu werden das Exemplar des Interessenten, der Name der aufzurufenden Methode sowie das Klassenobjekt der Nachrichten angegeben. Die Methode, die beim Interessent aufgerufen werden soll, muß einem bestimmten Format genügen. Konkret bedeutet das, daß die Methode nur einen Parameter haben darf, der vom Typ `msgObject` ist. Wird der Interessent benachrichtigt, indem die angegebene Methode vom `MessageBroker` aufgerufen wird, so wird das Exemplar der erhaltenen Nachricht als Parameter mit übergeben. Es ist zu beachten, daß der Interessent nicht nur bei Nachrichten der angegebenen Nachrichtenklasse informiert wird, sondern auch bei allen Unterklassen von dieser. Meldet sich z.B. ein Interessent für Nachrichten vom Typ `msgObject` an, so wird er bei jeder beliebigen empfangenen Nachricht informiert. Wie in der Abbildung 21 zu erkennen ist, kann wahlweise eine Klausel (siehe Abschnitt 4.4.2.2) zusätzlich angegeben werden. Weiterhin können mit dem `MessageBroker` auch Nachrichten versendet werden. Als Parameter wird ein Exemplar einer Nachricht übergeben, das zu allen Interessenten transportiert wird.

4.4.2.4 *MessageBrokerServer*

Der `MessageBrokerServer` dient dem Nachrichtenaustausch über Prozeßgrenzen hinweg. Er wird auf einem zentralen Server gestartet und fungiert als Dienstleistungserbringer für alle im System vorhandenen `MessageBroker`. Die gesamte Nachrichtenvermittlung ist auch ohne diese Komponente möglich, jedoch können dann nur im lokalen Prozeßraum Nachrichten übermittelt werden.

Der `MessageBrokerServer` wird wie folgt vom `MessageBroker` verwendet. Meldet sich ein Interessent beim `MessageBroker` für eine bestimmte Nachricht an, so meldet sich wiederum der `MessageBroker` selbst als Interessent beim `MessageBrokerServer` an. Wird eine Nachricht mittels `MessageBroker` versendet, so informiert er zuerst alle registrierten Interessenten im lokalen Prozeßraum und leitet sie anschließend an den `MessageBrokerServer` weiter. Dieser sendet die Nachricht an alle `MessageBroker`, die für diesen Nachrichtentyp bei ihm registriert sind.

4.5 Realisierung der Verteilungskomponente

In den vorangegangenen Kapiteln 4.3 und 4.4 wurden zwei Konzepte vorgestellt, welche grundlegend für die Realisierung der Verteilungskomponente des Transportsystems sind. In diesem Kapitel wird der Aufbau bzw. die Realisierung der Verteilungskomponente geschildert. Dabei liegt der Schwerpunkt auf der technischen Realisierung der in Kapitel 4.2 vorgestellten Konzeption des Verteilungsaspektes. Alle weitere im Transportsystem vorhandene Komponenten werden in den folgenden Betrachtungen vernachlässigt, da sich deren technische Umsetzung recht einfach aus den Konzepten der Komponenten in Kapitel 3.2 ableiten läßt.

Prinzipiell gibt es zwei Prozeßräume, die zu unterscheiden sind. Das sind der Client- und der Server-Prozeßraum.

Im Client-Prozeßraum, welcher den lokalen Arbeitsplatz repräsentiert, sind der `MessageBroker` (siehe Abschnitt 4.4.2.3) und der `Versandautomat-Client` vorhanden. Der `MessageBroker` dient dazu, den `Versandautomat-Client` zu benachrichtigen, wenn beim `Versandautomat-Server` neue Transportmappen vorliegen. Der `Versandautomat-Client` verwaltet alle lokalen Postkörbe. Er entnimmt die zu sendenden Transportmappen aus den

Postausgangskörben und versendet sie über den Versandautomat-Server. Weiterhin ist er auch dafür zuständig, neue Transportmappen beim Versandautomat-Server abzuholen und entsprechend evtl. Reservierungen in den Posteingangskörben abzulegen.

Im Server-Prozessraum arbeiten der MessageBroker und der Versandautomat-Server. Der MessageBroker hat hier die Aufgabe, die vom Versandautomat-Server erzeugten Nachrichten an die entsprechenden Versandautomat-Clients zu versenden. Der Versandautomat-Server dient dazu, die zu sendenden Transportmappen entgegenzunehmen und den Empfänger zu benachrichtigen. Weiterhin können bei ihm die neuen Transportmappen abgeholt werden.

Es gibt noch einen weiteren separaten Prozeß. Es ist der des MessageBrokerServer. Dieser wird aber in den weiteren Ausführungen ohne Bedeutung bleiben, da er für die technische Realisierung der Verteilungskomponente unmittelbar keine Relevanz hat. Lediglich aus Gründen der Vollständigkeit ist er hier erwähnt.

Der vollständige Aufbau ist in Abbildung 22 dargestellt.

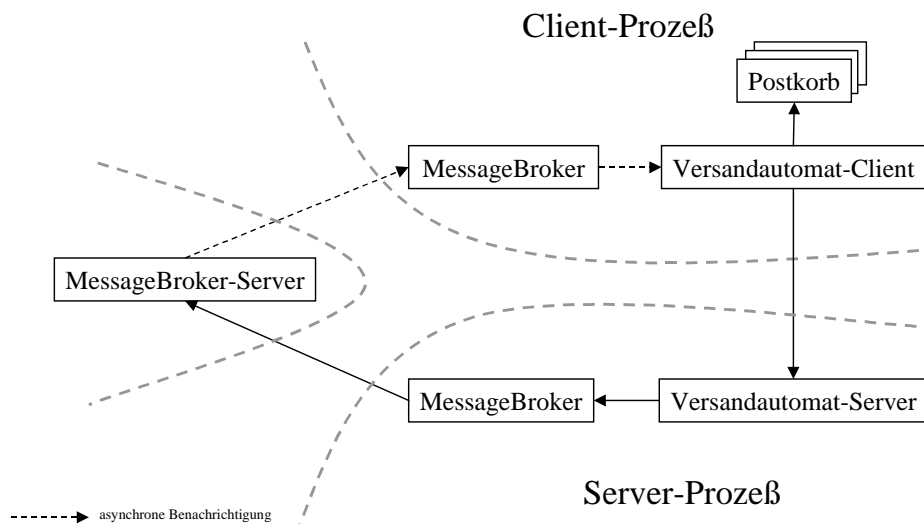


Abbildung 22: technische Realisierung der Verteilungskomponente

Die technische Realisierung wird anhand der folgenden Fragen erläutert:

- Wie wird sich für eine Adresse angemeldet?
- Wie wird eine Transportmappe an den Versandautomat-Server übergeben?
- Wie wird der Versandautomat-Client über den Erhalt einer neuen Transportmappe informiert?
- Wie erhält der Versandautomat-Client die neue Transportmappe vom Versandautomat-Server?

Wie wird sich für eine Adresse angemeldet?

Die Anmeldung für eine konkrete Adresse erfolgt bei dem lokalen MessageBroker. Dazu

wird eine Klausel vom Typ `clauseVersandautomat` (siehe Kapitel 4.4.2.2) erzeugt und zusammen mit dem Klassenobjekt der Klasse `msgVersandautomat` dem `MessageBroker` übergeben. Weiterhin müssen noch das Objekt, das sich für diese Nachricht interessiert, sowie der Methodennamen angegeben werden. Der `MessageBroker` verwendet die ihm übergebene Klausel, um festzustellen, ob die in einer Nachricht enthaltene Empfängeradresse für die lokalen Posteingangskörbe zutreffend ist.

Wie wird eine Transportmappe an den Versandautomat-Server übergeben?

Um eine Transportmappe versenden zu können, muß der Zugriff auf den `Versandautomat-Server` möglich sein. Das geschieht, wie in Abschnitt 4.3 bereits beschrieben ist, indem beim `Naming-Service` das `Server-Objekt` erfragt wird. War die Anfrage erfolgreich, so wird ein lokaler Repräsentant (`Proxy`) zurückgeliefert. Über dieses `Proxy` kann auf den Server zugegriffen werden, als wäre er im lokalen Prozeßraum vorhanden. Somit kann die Transportmappe als Parameter einer Methode dem `Versandautomat-Server` übergeben werden. Die Abbildung 23 zeigt eine beispielhafte Implementierung.

```
try
{ // Adresse des Server-Objektes
  String serverURL = "rmi://Server/WAM-Versandautomat-Server";
  // Versuchen, den RMI-Server zu finden
  VersandautomatServer va = (VersandautomatServer)Naming.lookup(serverURL);

  // übergeben der Transportmappe an den Versandautomat-Server
  va.sendeTransportmappe(t);
}
catch (RemoteException e)
{ ... }
```

Abbildung 23: Aufruf des Versandautomat-Server

Wie wird der Versandautomat-Client über den Erhalt einer neuen Transportmappe informiert?

Hat der `Versandautomat-Client A` dem `Versandautomat-Server` eine Transportmappe übergeben, so versendet der `Versandautomat-Server` eine Nachricht vom Typ `msgVersandautomat` über den `MessageBroker`. Diese Nachricht enthält die Adresse des Empfängers. Der `MessageBroker` des `Client B` ermittelt anhand der bei ihm registrierten Klauseln, ob der `Versandautomat-Client B` der Empfänger ist. Ist das der Fall, so wird an ihm die Methode gerufen, dessen Namen bei der Registrierung angegeben wurde. Der so informierte `Versandautomat-Client B` hat nun auf die Benachrichtigung angemessen zu reagieren.

Wie erhält der Versandautomat-Client die neue Transportmappe vom Versandautomat-Server?

Nachdem der `Versandautomat-Client` darüber benachrichtigt wurde, daß bei dem `Versandautomat-Server` eine neue Transportmappe vorliegt, kann er diese abholen. Dazu ruft er eine sondierende Methode am `Versandautomat-Server` auf. Wie bereits dargestellt

wurde, verhält sich der Methodenaufruf eines entfernten Objektes wie bei einem lokalen. Als Parameter übergibt der **Versandautomat-Client** die Angaben, die mit der Nachricht mitgesendet wurden (die Empfängeradresse sowie eine evtl. Reservierung). Mit diesen kann der **Versandautomat-Server** die entsprechende Transportmappe aus seinem Zwischenspeicher selektieren. Die gefundene Transportmappe wird dem Aufrufer als Rückgabewert übermittelt. Der RMI-Mechanismus von Java stellt sicher, daß die Datenübertragung vom Server zum Client transparent geschieht. Hat der **Versandautomat-Client** auf diesem Wege die Transportmappe erhalten, so legt er sie in einen entsprechend registrierten Posteingangskorb ab und löst das Ereignis für Neuzugang aus.

5 Zusammenfassung und Ausblick

Das in dieser Arbeit vorgestellte Transportsystem stellt ein Kooperationsmedium dar, mit dem kooperative Arbeit unterstützt werden kann. Mit diesem ist es möglich, computergestützt über Prozeßgrenzen hinweg Materialien, wie z.B. Dokumente und Formulare, auszutauschen. Ein Vorteil in der Verwendung des Transportsystems besteht u.a. darin, daß zum Zwecke des Versandes ein Medienbruch vermieden werden kann. Das trifft insbesondere dann zu, wenn bereits eine Softwareumgebung besteht und eingesetzt wird, in der die computergestützte Bearbeitung von Materialien realisiert wurde.

Wie bereits erwähnt, wurde mit diesem Transportsystem nur ein Kooperationsmedium umgesetzt. Weitere Möglichkeiten wären u.a. das Schwarze Brett, Postfächer sowie Rohrpost, um nur solche zu nennen, die die explizite Kooperation durch Materialaustausch (vgl. [WAM98]) unterstützen. Es ist aber auch eine Erweiterung in der Richtung denkbar, die eine noch komplexere Kooperation ermöglicht.

Ein anderer noch nicht berücksichtigter Aspekt ist die Kopplung an einen Vorgangsmoitor, mit dem der Versand von Materialien verfolgt werden kann.

In einigen Stellen dieser Arbeit wurde deutlich gemacht, daß eine optimale Umsetzung des Konzeptes nicht möglich war, da das zugrundeliegende Framework JWAM nicht allen Anforderungen entsprach. So ist bis jetzt kein identifizierbarer Arbeitsplatz durch das JWAM gegeben. Das hat zur Folge, daß die Postkörbe selbst die Eigenschaft besitzen müssen, in dem gesamten System identifiziert werden zu können. Besser ist die bereits erörterte Variante, bei der die Postkörbe durch den Kontext des Arbeitsplatzes, auf dem sie sich befinden, identifiziert werden können. Weiterhin mußte in diesem Zusammenhang ein Kompromiß für den Fall gefunden werden, daß kein entsprechender Posteingangskorb für eine gesendete Transportmappe vorhanden ist. So wird dieser Umstand bis jetzt so behandelt, daß die Transportmappe für einen gewissen Zeitraum zwischengespeichert wird. Wird in dieser Zeit kein entsprechender Posteingangskorb beim Versandautomat registriert, so wird die Transportmappe zurückgesendet. Eine bessere Lösung wäre, wenn alle potentiell möglichen Arbeitsplätze angemeldet werden würden anstatt der aktuell vorhandenen Postkörbe. Bei dieser Variante könnte der Versandautomat anhand der Registrierungen ermitteln, ob der Empfänger im System bekannt ist. Wenn ja, so wird die Transportmappe solange vorgehalten, bis der entsprechende Arbeitsplatz eingerichtet und die zugehörigen Posteingangskörbe angemeldet sind.

Das mit diesem Transportsystem verfolgte Leitbild eines Postversandsystems (vgl. [WAM98]) integriert sich in die WAM-Methode, so daß sich eine Implementation problemlos in eine bereits bestehende nach dem WAM-Ansatz entwickelte Softwareumgebung einbinden läßt.

6 Literaturverzeichnis

[Bohl98]

Holger Bohlmann: „Behälterbibliothek in Java“ (vorläufiger Titel), Studienarbeit Universität Hamburg, 1998

[GHJ+95]

E. Gamma, R. Helm, R. Johnson, J. Vlissides: „Design Patterns – Elements of Reusable Object-Oriented Software“, Addison-Wesley, 1995

[GKZ94]

Guido Gryczan, Klaus Kilberth, Heinz Züllighoven: „Objektorientierte Anwendungsentwicklung“, 2. Auflage, Vieweg, Braunschweig/Wiesbaden, 1994

[Gry95]

G. Grycan: „Situierete Koordination computergestützter qualifizierter Tätigkeit über Prozeßmuster“, Dissertation, Universität Hamburg

[JeSh]

Martti Jeenicke, Nol Shala: „Entwurf und Realisierung eines Fachlichen Modells für ein Pflegeplanungs- und dokumentationssystem“, Studienarbeit der Universität Hamburg, 1998

[JS97]

JavaSoft: RMI Specification; Sun Microsystems, 1997

[PLoP95]

John M. Vlissides, N. Korth, James O. Coplien (Eds.): „Pattern Languages of Program Design“, Addison-Wesley, 1996

[RoWo]

Stefan Roock, Henning Wolf: „Die Raummetapher zur Entwicklung kooperationsunterstützender Softwaresysteme für Organisationen“, Diplomarbeit der Universität Hamburg 1998

[RZ95]

Dirk Riehle, Heinz Züllighoven: „A Pattern Language for Tool Construction and Integration Based on the Tool & Material Metaphor“ in [PLoP95]

[WAM98]

Heinz Züllighoven: „Das objektorientierte Konstruktionshandbuch nach dem Werkzeug- & Material-Ansatz“, dpunkt verlag, 1998

[Wulf95]

Martina Wulf: „Konzeption und Realisierung einer Umgebung zur Koordination rechnergestützter Tätigkeiten in kooperativen Arbeitsprozessen“, Diplomarbeit Universität Hamburg, 1995

[Zühlke98]

Marco Zühlke: „Serialisierer-Pattern“ (vorläufiger Titel), Studienarbeit Universität Hamburg, voraussichtlich 1998