

# Frameworkbasierte Anwendungsentwicklung (Teil 5): Unterstützung von Kooperation mit persistenten fachlichen Behältern

Guido Gryczan, Andreas Havenstein, Stefan Roock, Ingrid Wetzel, Heinz Züllighoven

## Zusammenfassung

In diesem Teil unserer Serie über frameworkbasierte Anwendungsentwicklung beschäftigen wir uns mit den Themen Persistenz und Kooperation. Wir stellen mit der Registratur ein fachlich motiviertes Konzept vor, das Persistenz und Kooperation flexibel unterstützt. Zunächst veranschaulichen wir das fachliche Konzept und beschreiben dann am Beispiel des JWAM-Frameworks, wie dieses Konzept für die Anwendungsentwicklung bereitgestellt werden kann.

## Einführung und Motivation

Unsere Vorstellung von dezentralen Anwendungssystemen hat sich in den letzten Jahren herausgebildet: der Arbeitsplatz eines Anwenders, der mit Arbeitsmitteln (Werkzeugen) und Arbeitsgegenständen (Materialien) ausgestattet ist. Dieser Arbeitsplatz ist meist als Schreibtisch gestaltet. Der Anwender kann seine Aufgaben mit den verfügbaren Werkzeugen und Materialien erledigen. Zur typischen Ausstattung eines Schreibtischs gehören Werkzeuge zur Text- und Tabellenverarbeitung, Dokumente und Behälter, in denen Arbeitsgegenstände gesammelt werden können. Je nach Anwendungsbereich und Aufgabenspektrum des Anwenders wird die Ausstattung des Arbeitsplatzes mit spezialisierten Werkzeugen und Materialien unterschiedlich ausfallen. So wird der Arbeitsplatz für die Kundenberatung in einer Bank über die Standardausstattung hinaus Werkzeuge für die Depotanalyse eines Kunden oder für Modellrechnungen enthalten.

Für die Anwender resultiert aus dieser Grundidee ein einfaches und nachvollziehbares *Benutzungsmodell*: Dinge, die auf dem persönlichen Schreibtisch liegen, werden je nach Anwendungssituation verwendet, um eine bestimmte Aufgabe zu erledigen. Ein Benutzungsmodell ist für uns ein Modell darüber, wie Anwendungssoftware bei der Erledigung der anstehenden Aufgaben im jeweiligen Einsatzkontext benutzt werden kann (vgl. [Zül98]).

Zunächst haben wir nur die Aufgaben eines einzelnen Anwenders an seinem Arbeitsplatz betrachtet. Meist werden Aufgaben in Organisationen aber arbeitsteilig erledigt. Eine einfache Form der Arbeitsteilung ist die Bearbeitung von Dokumenten, bei der eine gemeinsame, zentrale Ablage verwendet wird: Nachdem der zuständige Mitarbeiter seine Aufgabe an einem Dokument erledigt hat, stellt er es in die Ablage zurück. Aus dieser Ablage kann ein anderer Mitarbeiter das Dokument entnehmen und zur weiteren Bearbeitung auf seinen Schreibtisch legen. So läßt sich z.B. das Vier-Augen-Prinzip zur Kontrolle bei der Kreditvergabe realisieren.

Dieses einfache Szenario führt zu den Fragen, die wir in diesem Artikel ansprechen wollen. Wir erläutern zunächst, wie das einfache Benutzungsmodell vom elektronischen Schreibtisch so erweitert werden kann, daß Arbeitsteilung in der gerade skizzierten Form leicht integriert werden kann. Dazu werfen wir einen Blick auf die Verwaltungswissenschaften: Dort existieren ausgearbeitete Modelle, wie Ablagen, oder genauer, Registraturen und Archive verwendet werden. Daraus leiten wir ein geeignetes Kooperationsmodell ab. Dann diskutieren wir die technische Architektur eines zentralen Ablagesystems im WAM-Ansatz. Wir gehen der Frage nach, welche Werkzeuge, Automaten und Materialien notwendig sind, um eine fachliche motivierte zentrale Ablage zu implementieren. Zuletzt diskutieren wir, wie ein Ablagesystem in das JWAM-Framework integriert wird. Wir motivieren, daß ein Ablagesystem mit denselben Konstruktionsprinzipien entworfen und implementiert werden kann, wie die anderen Komponenten des Frameworks.

# Ablagen zur Verwaltung gemeinsam genutzter Materialien<sup>1</sup>

Das Motiv für den Blick in die Verwaltungswissenschaften ist aus dem WAM-Ansatz bekannt: Wir betrachten bewährte Materialien des Anwendungsbereichs und deren Handhabung, um ein Verständnis für die Problematik des Anwendungsfalles zu entwickeln.

Auch wenn Verwaltungswissenschaften auf eher staubige Konzepte hindeuten, nutzen wir die Erfahrungen dieses Bereichs um ein fachlich motiviertes technisches Modell zu entwickeln. Dieses Modell soll einerseits den vertrauten Umgang mit bewährten Gegenständen erlauben und andererseits die Schwächen vermeiden, die sich aus dem Umgang mit materiellen Gegenständen ergeben und die Potentiale nutzen, die uns Software bietet. Mit anderen Worten: wir motivieren die Verwendung einer neuen Technologie auf der Basis bewährter Konzepte.

Ein zentraler Begriff für die Verwaltung gemeinsamer Dokumente ist die Zwischenregistratur (im folgenden: *Registratur*). Die Registratur

- ist die zentrale Instanz für die Schriftgutverwaltung in Unternehmen und Behörden,
- ist der „Kern des Informationswegenetzes“ (vgl. [Jäg96]), und
- dient als Schnittstelle für die verschiedenen unabhängigen Abteilungen.

Jedes Dokument ist in der Registratur erfaßt, nachdem es angelegt und in der Registratur abgelegt wurde. Zu jedem Zeitpunkt ist bekannt, wo sich ein Dokument befindet, und wer es aus der Registratur entnommen hat. In der Registratur lagern alle Dokumente, die nicht aktuell bearbeitet aber zur Vorgangsbearbeitung prinzipiell benötigt werden.

Für unser Beispiel müssen wir zudem verstehen, welche Aufgaben die beteiligten Personen als Rollen übernehmen. Für die Verwaltung der Registratur ist ein *Registrator* verantwortlich. Er

- koordiniert die konkurrierenden Zugriffe auf die Registratur, und
- gewährleistet ihre Konsistenz.

Zur Erfüllung seiner Aufgaben verwendet der Registrator verschiedene *Registriermittel*:

- Im *Inventarverzeichnis* ist jedes Dokument, auch Registriergut genannt, verzeichnet. Jedes Registriergut ist durch Informationen charakterisiert; etwa: Aussteller, Registrierdatum oder die Aufbewahrungsfrist, nach der das Dokument vernichtet werden kann.
- Als *Aktenausgabeverbleibsnachweis* (!) können Fehlkarten verwendet werden. Sie stehen als Platzhalter in der Registratur an der Stelle des Registrierguts. Auf ihnen ist die Ausgabedauer und der aktuelle Verwahrort verzeichnet.
- Im *Terminbuch* trägt der Registrator nach Vorgabe eines Sachbearbeiters den Wiedervorlagetermin für ein Dokument ein.
- In der *Vormerkliste* verwaltet der Registrator Anfragen von Sachbearbeitern, die auf ein entnommenes Registriergut zugreifen wollen.

Neben der hier beschriebenen Registratur kennt die Verwaltungswissenschaft (siehe z.B. [Jäg96]) noch weitere Formen von Ablagen: dazu gehören z.B. Altablagen oder Archive. In unserem Zusammenhang ergeben sich jedoch keine signifikanten Unterschiede zur Registratur.

Wir betrachten im folgenden, wie die Registratur als Kooperationsmittel und der Registrator als Koordinator in ein fachlich und technisch stimmiges Kooperationsmodell für Arbeitsplatzsysteme integriert werden kann.

## Kooperationsmodell: Arbeiten mit einer Registratur<sup>2</sup>

Um aus dem oben skizzierten Umgang mit einer traditionellen Registratur gewinnbringende Rückschlüsse für die softwaregestützte Modellierung zu ziehen, müssen wir uns zunächst scheinbar Selbstverständliches verdeutlichen:

---

<sup>1</sup> Dieser Abschnitt basiert auf [Hav99].

<sup>2</sup> Dieser Abschnitt ist eine im Hinblick auf das JWAM-Framework vorgenommene Konkretisierung des entsprechenden Kapitels aus [Zül98].

- Eine Registratur dient der *impliziten Kooperation*.

Bei der impliziten Kooperation sind *die anderen* nicht direkt im Benutzungsmodell sichtbar. Sie treten für einen Anwender der Registratur nur indirekt auf. Verwendet ein anderer die Registratur, indem er z.B. ein Dokument entnimmt, hinterläßt er eine *Spur* in Form einer Fehlkarte. Explizite Formen der Kooperation, z.B. durch Postversand, werden durch die Registratur nicht unterstützt (siehe dazu Kasten 1 Explizite Kooperationsmodelle)

- Die Anwender koordinieren sich im wesentlichen durch Konventionen außerhalb des Systems.

Unter Koordination verstehen wir hier den Mechanismus zur Abstimmung bei kooperativer Arbeit: Wenn wir kooperieren, müssen wir uns koordinieren. Die Registratur stellt über die Registriermittel, z.B. die Fehlkarten, die Grundlage einer Koordination der Anwender bereit.

- Ein Material kann zu einem Zeitpunkt nur an genau einem Ort sein.

Entweder befindet sich ein Dokument am Arbeitsplatz eines Anwenders oder es befindet sich in der Registratur. Es kann nicht sein, daß sich ein Dokument zu einem Zeitpunkt an zwei Orten befindet. Zudem soll der Eindruck vermieden werden, daß ein Dokument in der Registratur bearbeitbar ist. Ein Dokument muß zunächst auf einen Arbeitsplatz transportiert werden, ehe es dort bearbeitet werden kann.

- Dokumente können kopiert werden.

Zwischen einer Arbeitskopie und dem Original dieser Kopie besteht keine technische Verbindung. Eine Änderung an einer Arbeitskopie hat daher keine Auswirkungen auf das Original.

Diese Merkmale übertragen wir auf unser Modell einer *Softwareregistratur*. Eine Softwareregistratur ist demnach die Verwaltung des Behälters „Registratur“ durch den Registrator. Der Registrator ist, softwaretechnisch gesehen, ein Automat (vgl. [Zül98]): Eine Entwurfs- und Konstruktionseinheit, die nach fest vorgegebenen Regeln eine bestimmte Dienstleistung erbringt. Hier besteht die Dienstleistung in der Wahrung der Konsistenz der Registratur und der Behandlung von Konkurrenzsituationen beim Zugriff.

Wir können jetzt darüber nachdenken, wie wir die Möglichkeiten von Software in den Entwurf der Softwareregistratur einfließen lassen. Dazu gehört insbesondere die Verwaltung von Originalen und Kopien durch die Softwareregistratur.

Wir sehen im wesentlichen zwei übliche Verwendungszusammenhänge von Originalen und Kopien, die eng verknüpft sind.

Im Zusammenhang (1) verwenden wir den Begriff Original im Sinne von rechtlich verbindlichen Dokumenten, evtl. auf speziellem Papier, mit Unterschriften, Stempel oder Siegeln. Es kann mehrere Originale (z.B. bei Vertragsabschlüssen) oder besondere Kopien (Zweitexemplare) geben. Normalerweise können von einem Original beliebig viele Kopien gemacht werden, die fallweise beglaubigt werden. Dies geschieht bei der Ausstellung; spätere Änderungen sind nur unter besonderer Kennzeichnung (z.B. mit Abzeichnen) möglich. Zweitexemplare werden wie Originale gehandhabt. Änderungen an Kopien sind normalerweise bedeutungslos.

Der andere Zusammenhang (2) betont die Verwendung von Kopien, im Zusammenhang mit der (parallelen) Nutzung oder Distribution von Information. Kopien können hierbei als Arbeitsgrundlage dienen und mit Anmerkungen versehen werden.

Beim Übergang zu elektronischen Registraturen und Dokumenten unterscheiden wir nachfolgend die Fälle a, b und c. In Anlehnung an (1) kann es Anwendungsbereiche geben, die strikt zwischen

#### **Kasten 1:** Explizite Kooperationsmodelle

Die folgend benannten expliziten Kooperationsmodelle dienen jeweils dazu, eine bestimmte Form der Kooperation zu unterstützen. Wir unterscheiden:

- Gruppenpostfächer: Sie ermöglichen unter Verwendung einer Postfachwand eine flexible Art der Post- und Arbeitsverteilung für kleine Gruppen von Anwendern
- Rohrpost: Die 1 zu 1 Verbindung von Arbeitsplätzen unterstützt den Dokumentenaustausch zwischen genau zwei Arbeitsplätzen
- Postversand: Unter Verwendung von (unternehmensspezifischen) Adressierungsarten und Adressen wird eine allgemeine Verbindung von Arbeitsplätzen erreicht.
- Laufzettel: Sie unterstützen die flexible Abarbeitung von routinemäßigen Arbeitsvorgängen.

Näher werden Kooperationsmodelle in [Zül99] erläutert.

Originalen und Kopien mit entsprechend unterschiedlichen Handhabungen trennen (siehe a, b). In anderen Bereichen in Anlehnung an (2) möchten wir die parallele Nutzung – Einsichtnahme und gegebenenfalls parallele Bearbeitung – von Dokumenten am Rechner unterstützen. Um dies zu erzielen, bedienen wir uns der Kopiermetapher. Wir modifizieren in diesen Fällen den Begriff Original (siehe c).

Die drei Fälle sind:

(a) *Exklusiver Zugriff auf das Material für einen Anwender; es gibt keine Kopien.*

Das entnommene Material ist für andere Benutzer als entliehen gekennzeichnet und damit nicht ausleihbar.

*Grund für die Modellierung:* Ein Material soll nur an einer Stelle vorhanden und bearbeitbar sein.

*Anwendungsbeispiel:* Es gibt Dinge, bei denen Kopien nicht mit vertretbarem Aufwand erstellt werden können oder wo die Arbeit auf möglicherweise modifizierten oder veralteten Kopien problematisch ist. Ein Jahresabschluß oder eine Bonitätsprüfung sollte beispielsweise nur auf den aktuellen Informationen des Originals gemacht werden.

*Konsequenz:* Die Benutzer müssen darüber informiert werden, wer das Material aus der Registratur entnommen hat (und sich gegebenenfalls an diesen wenden), da es sonst zu Unterbrechungen und Störungen im Arbeitsprozeß kommen kann. Um Unterbrechungen zu vermeiden, muß die Registratur für weitere Ausleiher Informationen über den Verbleib des Materials bereitstellen. Mitarbeiter müssen sich per e-mail oder außerhalb des Systems koordinieren, wie mit den konkurrierenden Ausleih- und Bearbeitungswünschen umgegangen wird.

(b) *Exklusiver Zugriff auf ein Original, Kopien sind möglich.*

Nur ein Anwender kann das Original des Materials aus der Registratur entnehmen. Weitere Anwender erkennen, daß das Original ausgeliehen ist, können sich aber Arbeitskopien ziehen. In diesem Fall behält der Registrator immer eine Kopie des Originals. So kann er weitere Kopien erstellen und herausgeben. Wird das Original zurückgestellt, ersetzt es die Kopie des Registrators.

*Grund für die Modellierung:* Auch wenn das Original eines Materials nicht mehr verfügbar ist, sollen Arbeitskopien zur Verfügung stehen.

*Konsequenz:* Die Koordination konkurrierender Änderungen liegt bei den Anwendern. Änderungen am Original ziehen keine automatischen Änderungen an den Kopien nach sich. Benutzer von Kopien können über die Rückgabe des Originals informiert werden.

*Anwendungsbeispiel:* In vielen Anwendungsfällen reicht es, „einen Blick“ auf ein Dokument zu werfen. Dies zu verhindern wäre unsinnig, da fachlich deutlich gemacht werden kann, daß ein Anwender nur eine Kopie eines Materials erhält.

(c) *Zugriff immer nur auf Kopien möglich.*

Jeder Anwender erhält nur eine Arbeitskopie mit internem Zeitstempel. Eine Arbeitskopie kann das Original ersetzen. Dabei werden die Zeitstempel verglichen. Wenn ein Original durch eine Arbeitskopie mit einem älteren Zeitstempel ersetzt werden soll, wird der Anwender über den Konflikt und seine Ursache informiert.

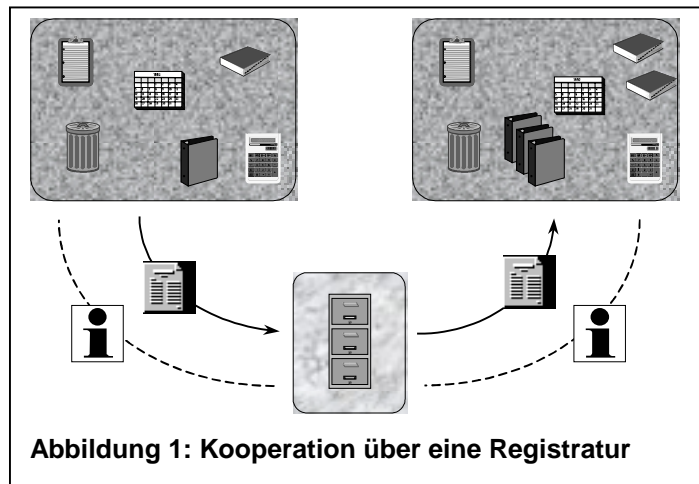
*Grund für die Modellierung:* Oft kann das Original nicht dauerhaft exklusiv für einen Anwender verfügbar gemacht werden (z.B. ein Konto). Bei vorrangig lesendem Zugriff können viele Anwender konkurrierend zugreifen. Die wenigen konkurrierenden Bearbeitungsvorgänge können nicht automatisch geklärt werden.

*Konsequenz:* Der Benutzer wird über Bearbeitungskonflikte informiert und kann sich anhand der mitgelieferten Informationen mit anderen Benutzern über die Regelung des Konflikts verständigen.

*Anwendungsbeispiel:* Im gesamten Formularwesen ist es unsinnig, Originale und Kopien zu unterscheiden. Formulare sind eigentlich immer Kopien einer Vorlage. Eine Organisation wird aber Wert darauf legen, daß immer aktuelle Kopien eines Formulars verwendet werden.

Der Originalbegriff gewinnt hier eine neue Bedeutung. Formulare werden nur als Kopien verteilt und bearbeitet. Die parallele Bearbeitung ist weitgehend unproblematisch, soweit sie sich auf unterschiedliche Vorgänge bezieht. Im Formulararchiv wird ein eindeutige Formularvorlage vorgehalten, die ihrerseits veränderbar ist. So wird sichergestellt, daß nach Änderung nur aktuelle Formularexemplare entnommen werden.

Welche Strategie von einer Softwareregistratur implementiert wird, ist vom jeweiligen Anwendungsfall abhängig. Wir werden im nächsten Abschnitt die in JWAM gewählte Entwurfsentscheidung begründen und herausarbeiten, wie das Kopierverhalten in der JWAM-Softwareregistratur eingestellt werden kann. Abbildung 1 zeigt das aus diesen Überlegungen resultierende Kooperationsmodell: Von einem persönlichen Arbeitsplatz werden Dokumente in die Registratur gestellt und aus ihr herausgeholt. Die Registratur gibt Auskunft über den Entnehmer von



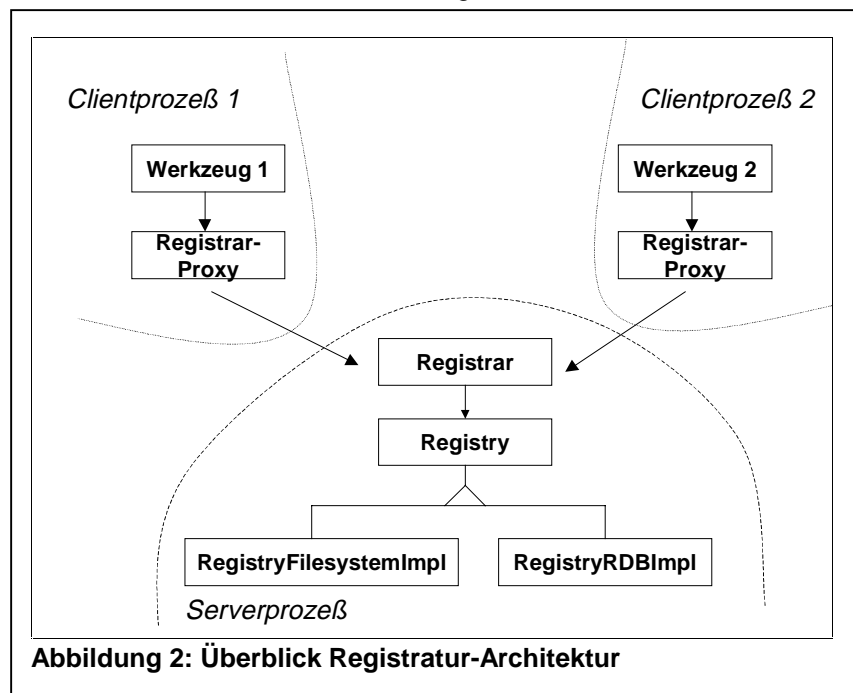
Dokumenten. Die möglicherweise auftretende Konkurrenzsituation ist transparent, d.h. ein Anwender kann erkennen, daß ein anderer Anwender ein Dokument in Arbeit hat.

Eine weitere Selbstverständlichkeit im Umgang mit traditionellen Registraturen haben wir bewußt noch nicht erwähnt: Wir erwarten, daß die in einer Registratur verwalteten Dokumente nicht flüchtig sind. Technisch ausgedrückt: Die Registratur ist ein Persistenzmedium für ihre Dokumente. Diese Selbstverständlichkeit soll auch für die Softwareregistratur gelten. Wir werden deswegen im folgenden Abschnitt nicht nur den technischen Entwurf der Benutzung der Softwareregistratur erläutern, sondern auch, wie die Softwareregistratur Dokumente persistent macht.

## Die Architektur der Software-Registratur im JWAM-Framework

Das Registraturkonzept läßt sich einfach in einem Framework für die Anwendungsentwicklung bereitstellen. Einen Überblick über die Architektur der Software-Registratur für den Einsatz in einer verteilten Umgebung gibt Abbildung 2.

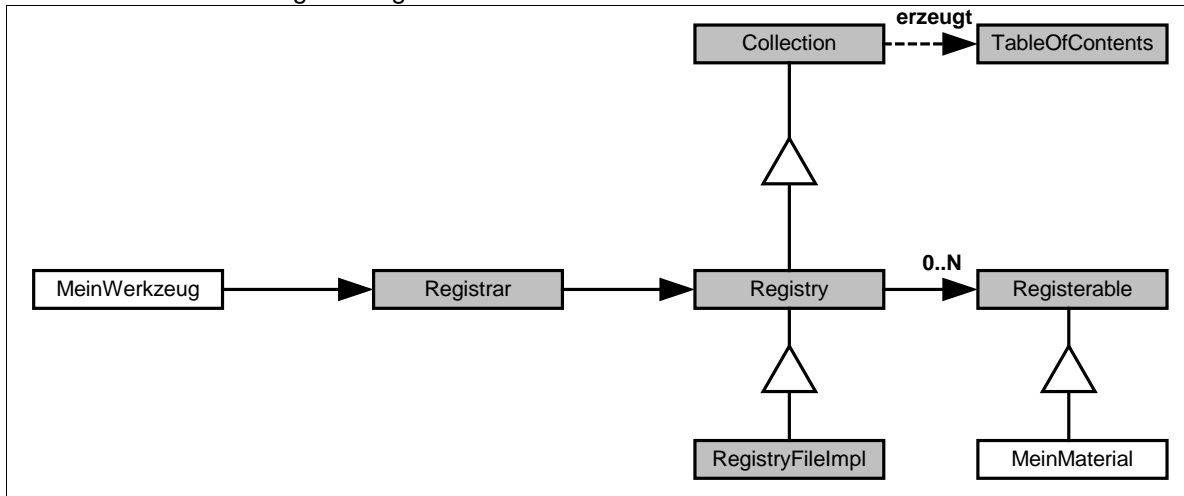
Abbildung 2 zeigt die Komponenten der Registratur (engl. Registry) mit einer möglichen Aufteilung auf Prozesse. Aus der Abbildung wird deutlich, daß ein vom Benutzer verwendetes Werkzeug nicht direkt auf der Registratur arbeitet, sondern über einen Registrar-Proxy (engl.: Registrar) den Registrar auf dem Server anfragt. Implementiert wird der Zugriff auf den Registrar über das Proxy-Muster (vgl. [Gam98]). Näheres dazu später.



Wir haben dieses Konzept im JWAM-Framework (siehe [JWAM99]) in der Handhabungs- und

Präsentationsschicht<sup>3</sup> implementiert. Auch wenn das Registraturkonzept originär keine Oberflächenanteile enthält, beeinflusst es doch ganz wesentlich die Handhabung des Systems.

Abbildung 3 zeigt die prinzipiellen Klassen des Registraturkonzepts im JWAM-Framework. Die Framework-Klassen sind grau dargestellt.



**Abbildung 3: Klassendiagramm: Unterstützung im JWAM-Framework**

Wie oben im fachlichen Kooperationsmodell beschrieben, greifen Werkzeuge nicht direkt auf die in der Registratur verwalteten Materialien zu, sondern fordern Materialien beim Registrar an. Der Registrar ist als Automat nach dem Proxy-Pattern realisiert. Der lokale Proxy leitet die Anfragen an den Registrar auf dem Server weiter, wo der konkurrierende Zugriff auf Materialien koordiniert wird. Der Registrar-Proxy kapselt die Verteilungsmechanismen, so daß Werkzeuge, die in einer verteilten Umgebung in verschiedenen Client-Prozessen laufen, keine Kenntnis über die verwendete Middleware besitzen müssen<sup>4</sup>. Der Umgang des Werkzeugs mit dem Registrar ist fachlich motiviert. Abbildung 4 zeigt einen Ausschnitt der Schnittstelle des Registrators.

|   |   |
|---|---|
| <p><b>Registrar</b></p> <p>Thing original(RegistrationNo) //liefert ein Ding<br/>         Registerable copy(RegistrationNo) //liefert eine fachliche Kopie<br/>         Registerable register(Registerable) //meldet ein Ding bei der Registratur an<br/>         void restore(Registerable) // stellt ein Ding in die Registratur zurück<br/>         void reserve(RegistrationNo) // reserviert ein entliehenes Ding<br/>         void notifyMeOf(RegistrationNo) // merke für Änderungen vor<br/>         TableOfContents tableOfContents() // liefert das Inhaltsverzeichnis<br/>         void sniff(Sniffer) // suche Dinge in der Registratur<br/>         Sniffer[] waitingSniffers () // die wartenden Schnüffler mit Suchergebnissen</p> <p><b>Abbildung 4: Schnittstelle des Registrators</b></p> | <p>Der Registrar ist die koordinierende Instanz im Serverprozeß. Die Anfragen der Werkzeuge aus verschiedenen Prozeßräumen fließen hier zusammen und werden vom Registrar koordiniert. Der Registrar gibt die</p> |
|---|---|

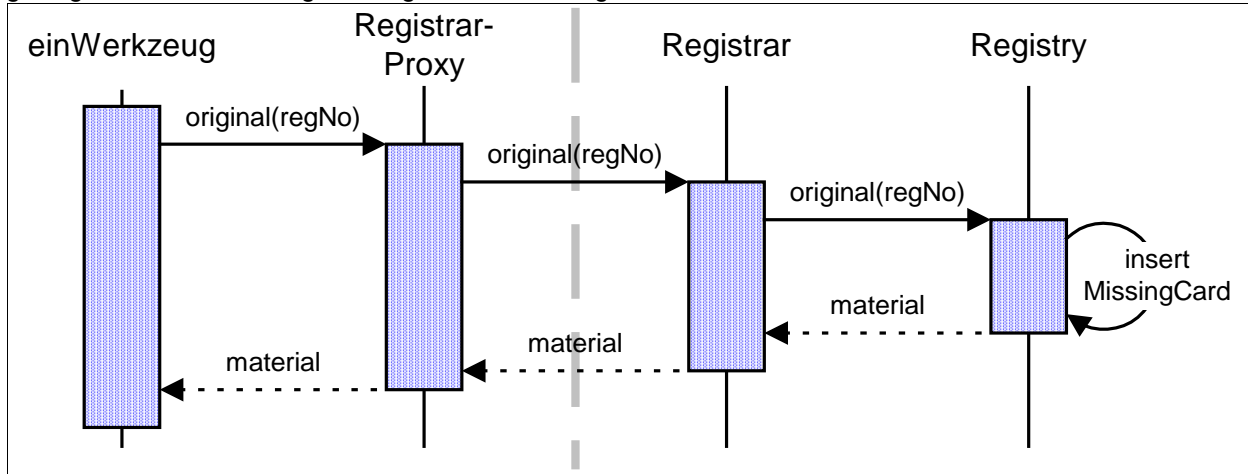
Materialien aus, registriert neue Materialien und stellt entnommene Materialien zurück. Zudem verwaltet er die Vormerkliste und benachrichtigt Werkzeuge über Ereignisse in der Registratur. Der Registrar verwendet eine Registratur zur Verwaltung der Materialien.

Die Registratur ist ein fachlicher, persistenter Behälter. In diesem Behälter können Materialien jeden Typs und jeder Größe gelagert werden, solange sie die im Interface „Registrierbar“ (engl. Registerable) festgelegte Schnittstelle erfüllen. Über diese Schnittstelle können zur Registrierung notwendige

<sup>3</sup> Zum Schichtenmodell siehe den ersten Teil dieser Serie ( [Ble99a]).

<sup>4</sup> Im JWAM-Framework ist das Registraturkonzept mit RMI (Remote-Method-Invocation) realisiert. Eine Umstellung auf eine andere Technologie (z.B. CORBA) ist sehr einfach, da lediglich Registrar-Proxy und Registrar ausgetauscht werden müssen, die jeweils sehr wenig Programmcode enthalten.

Informationen an das Material angeheftet werden. Über jedes Material sind damit Metainformationen wie eine Materialbezeichnung, der Ersteller und das Erstellungsdatum, sowie der Bearbeiter und Zeitpunkt der letzten Bearbeitung verfügbar. Zur eindeutigen Identifizierung von Materialien verwendet die Registratur Registriernummern, die im JWAM als Fachwerte<sup>5</sup> implementiert sind. In JWAM wird das Registrierbar-Interface von den fachlichen Behältern implementiert. In Abbildung 5 ist beispielhaft gezeigt, wie ein Werkzeug ein Original aus der Registratur entnimmt.



**Abbildung 5: Entnehmen eines Originals aus der Registratur**

## Kooperationsunterstützung

Die Registratur unterstützt in der aktuellen Implementation im JWAM-Framework Kooperation durch das Konzept von Original und Kopie. Je nach fachlich geforderter Kopierstrategie stellt die Registratur dem Benutzer einen der drei oben beschriebenen Kopiermechanismen bereit. Kopien sind dabei jeweils eigene, vom Original technisch unabhängige Materialien, die über ihre Metainformationen als Kopie des Ursprungsmaterials erkennbar sind. Wird von einem Werkzeug ein Original angefordert, das an einen anderen Benutzer ausgegeben ist, so gibt die Registratur ein Fehlkarten-Objekt zurück, an dem der Aufenthaltsort des Originals ablesbar ist. Es mag nicht sofort einleuchten, warum diese Operation so unterschiedliche Objekte liefert. Dies liegt in der Aufteilung in mehrere Prozesse begründet. Würde man vor dem Zugriff auf ein Material in der Registratur prüfen, ob das Original verfügbar ist, so wäre die erhaltene Information möglicherweise irreführend. Denn zwischen der Prüfung auf Vorhandensein und der Anfrage nach dem Original kann ein anderer Prozeß den Zustand der Registratur bereits verändert haben. Insbesondere kann er das Original in der Zwischenzeit entnommen haben.

Um verschiedene Persistenzmedien performant nutzen zu können, gibt es auf das jeweilige Persistenzmedium spezialisierte Subklassen der Registratur. In Abbildung 2 sind zwei spezialisierte Komponenten angedeutet, die zur Speicherung auf die Nutzung des Dateisystems und einer relationalen Datenbank optimiert sind.

## Inhaltsverzeichnisse

Wie Abbildung 3 zeigt, ist die Registratur als fachlicher Behälter in die JWAM-Behälterhierarchie eingebunden. Damit läßt sie sich wie andere Behälter auch unabhängig vom Registrator als lokale Arbeitsplatzregistratur einsetzen.

Wie jeder fachliche JWAM-Behälter (es gibt z.B. noch Ordner) kann die Registratur ein Inhaltsverzeichnis (TableOfContents) ihrer verwalteten Materialien erstellen und nach außen geben. Das von der Registratur erzeugte Inhaltsverzeichnis enthält eine Zusammenstellung der Metainformationen aller in der Registratur verwalteten Materialien. Die Metainformationen eines

<sup>5</sup> Zum Konzept der Fachwerte siehe den vorherigen Teil dieser Serie ([Ble99b]).

Materials werden in Beschreibungsobjekten (ThingDescription) im Inhaltsverzeichnis geliefert. Zu jeder Materialklasse kann es eine Beschreibungsklasse geben. Existiert keine Beschreibungsklasse, so wird die Beschreibungsklasse der Oberklasse verwendet. Somit ist es möglich, die Inhaltsverzeichnisse sehr flexibel anzupassen.

## Behälter als fachliche Interpretation von Transaktionen

Müssen mehrere Dinge immer zusammen als quasi-atomare Einheit gespeichert werden, so kämen in einem datenbankbasierten Ansatz Transaktionen zum Einsatz. Transaktionen sind jedoch kein fachlich motiviertes Konzept. Wir setzen dazu den fachlichen Behälter ein. Ein Ordner oder eine Mappe sind solche fachlichen Behälter. In der Regel werden z.B. ausgefüllte Formulare nicht direkt in einer Registratur ablegt, sondern in einer Mappe zu einem Vorgang zusammengefasst. In diesen fachlichen Behältern finden sich also die Dinge, die man in einem datenbankbasierten Ansatz über eine Transaktion klammern muß. Die Klammerung lässt sich hier fachlich interpretieren; sie wird vor dem Anwendungsentwickler verborgen. Der Anwendungsentwickler übergibt die Mappe einfach dem Registrator. Dieser bzw. die zugrundeliegende Registratur-Implementierung speichern die Inhalte fachlicher Behälter und schützen dies ggf. durch eine Transaktion.

## „Schnüffler“

Bei großen Datenbeständen in verteilten Anwendungen ist es nicht praktikabel, zum Suchen eines Materials das gesamte Inhaltsverzeichnis der Registratur auszulesen und über das Netz zu transportieren. Um Materialien aufzuspüren, lassen sich deshalb spezielle Automaten, die sogenannten Schnüffler (Sniffer), verwenden. Schnüffler können vom Werkzeug mit Suchkriterien konfiguriert und dann auf einen Behälter angesetzt werden, um dort lokal den Datenbestand zu untersuchen. Wird die Registratur in einer verteilten Umgebung eingesetzt, wird im Klientenprozeß von einem Werkzeug der passende Schnüffler konfiguriert und über den Registrator an die Registratur im Serverprozeß geleitet. Dort arbeitet der Schnüffler auf dem lokalen Datenbestand und erstellt ein Verzeichnis der zu den Suchkriterien gefundenen Materialien. Hat der Schnüffler seine Arbeit erledigt, kann das lokale Werkzeug ihn beim Registrator abholen und sich das Verzeichnis der gefundenen Dinge vom Schnüffler geben lassen. Jetzt kann das Werkzeug z.B. das Verzeichnis anzeigen und der Anwender kann einen Eintrag zur Bearbeitung auswählen. Das Werkzeug versucht daraufhin, beim Registrator das Original zu bekommen. Ist das Original nicht mehr verfügbar, so wird die zurückgelieferte Fehlkarte angezeigt. Auf dieser kann der Anwender nachsehen, wer das Original zur Zeit bearbeitet. Dadurch, daß das Inhaltsverzeichnis explizites Bestandteil des Benutzungsmodells ist, kann der Benutzer dieses Verhalten gut nachvollziehen: es ist einsichtig, daß ein Inhaltsverzeichnis immer nur den Stand zu einem bestimmten Zeitpunkt repräsentiert und inzwischen veraltet sein kann.

## Benachrichtigungen

Für einige Werkzeuge kann es interessant sein, über Aktionen der Registratur informiert zu werden. Eine entsprechende Benachrichtigung kann durch den Nachrichtenvermittler im JWAM-Framework leicht realisiert werden. Werkzeuge können sich beim Nachrichtenvermittler für bestimmte Nachrichten anmelden und werden informiert, wenn die zu diesen Nachrichten passenden Ereignisse auftreten (siehe Abbildung 6).

So kann sich beispielsweise ein Werkzeug für das Zurückgeben von Materialien an die Registratur interessieren. In diesem Fall meldet sich das Werkzeug für die Nachricht „ThingRestored“ beim Nachrichtenvermittler an. Mitunter sind für das Werkzeug

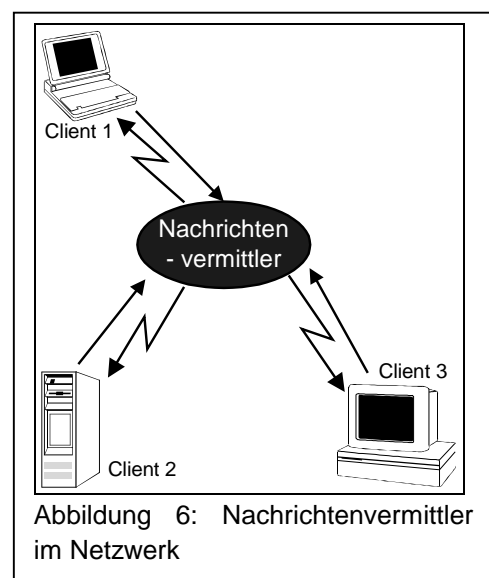
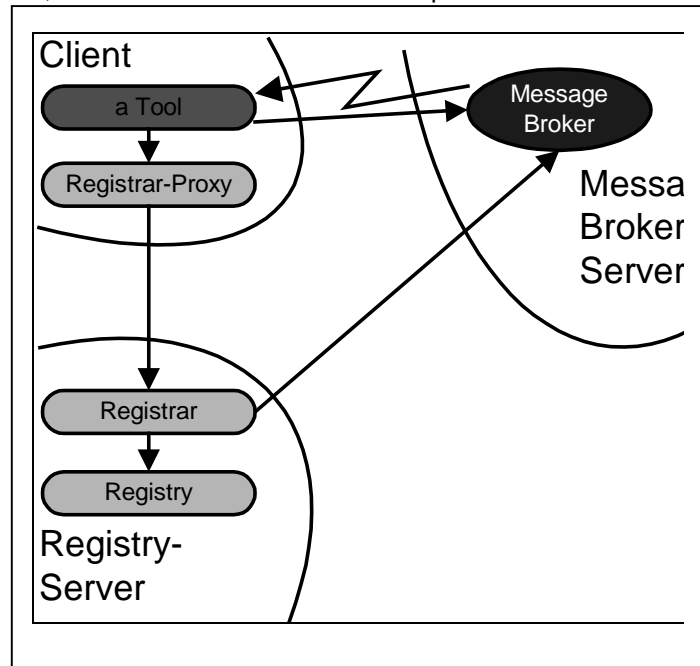


Abbildung 6: Nachrichtenvermittler im Netzwerk



nicht alle Nachrichten eines Typ interessant, sondern nur bestimmte. Beispielsweise könnte ein Werkzeug, das eine Kopie anzeigt, sich das Original reservieren wollen. Es möchte dann informiert werden, wenn das Original wieder in der Registratur verfügbar ist. Dazu können die empfangenen Nachrichten über Klauseln eingeschränkt werden. Für den skizzierten Fall kann die Klausel ein Material mit einer bestimmten Registraturnummer festlegen. Da die Klauseln auf dem Server gehalten und geprüft werden, ist dieser Mechanismus ein effektives Verfahren zur Minimierung des Netzlast. Abbildung 7 zeigt die resultierende Laufzeitarchitektur.



## Zusammenfassung und Ausblick

Wir haben mit der Konzept von Registrator und Registratur eine Modell für die implizite Kooperation implementiert. Damit haben wir gezeigt, daß die klassische WAM-Herangehensweise auch für die Unterstützung kooperativer Arbeit tragfähig ist: Wir haben die Konzepte aus den Verwaltungswissenschaften adaptiert und sind mit dem JWAM-Framework jetzt in der Lage einen Ausschnitt kooperativer Arbeit mit einem nachvollziehbaren Benutzungs- und Kooperationsmodell zu unterstützen. Erweiterungen dieses Kooperationsmodells sind zur Zeit in Arbeit: Wir entwickeln Automaten, die Dokumente "verfolgen", d.h. die einem Anwender bei Bedarf darüber informieren wann sich ein Dokument wo und wie lange aufgehalten hat. Hier eröffnet sich ein weites Feld für Geschäftsprozeßanalysen.

Weiterhin arbeiten wir an der Realisierung weiterer Kooperationsmodelle. Die nächsten JWAM-Versionen werden neben Mitteln zur Versendung von Dokumenten per Post auch die Bearbeitung komplexer arbeitsteiliger Vorgänge unterstützen.

## Literatur

- [Bäu98] D. Bäumer, Softwarearchitekturen für die rahmenwerkbasierete Konstruktion großer Anwendungssysteme, Dissertationsschrift am FB Informatik der Uni Hamburg, 1/98
- [Ble99a] W.-G. Bleek, G. Gryczan, C. Lilienthal, M. Lippert, S. Roock, H. Wolf, H. Züllighoven, Frameworkbasierte Anwendungsentwicklung (Teil 1), in: OBJEKTspektrum 1/99
- [Ble99b] W.-G. Bleek, C. Lilienthal, H. Züllighoven, Frameworkbasierte Anwendungsentwicklung (Teil 4), in: OBJEKTspektrum 4/99
- [Jäg96] Wieland Jäger, Ulrike Scharfenberger, Bernhard Scharfenberger: Verwaltungsreform durch Neue Kommunikationstechnik? Soziologische Untersuchungen am Beispiel Schriftgutverwaltung. Westdeutscher Verlag 1996.
- [Gam98] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Entwurfsmuster : Elemente wiederverwendbarer objektorientierter Software, 2. Auflage, Addison-Wesley 1998
- [Gry99b] G. Gryczan, C. Lilienthal, M. Lippert, S. Roock, W. Strunk, H. Wolf, H. Züllighoven, Frameworkbasierte Anwendungsentwicklung (Teil 2): die Konstruktion interaktiver Anwendungen, in: OBJEKTspektrum 2/99
- [Hav99] A. Havenstein. Unterstützung Kooperativer Arbeit durch eine Software-Registratur. Studienarbeit. Arbeitsbereich Softwaretechnik. Universität Hamburg, Fachbereich Informatik, 1999.
- [JWAM99] JWAM: Java Framework for the Tools and Materials Approach, Uni Hamburg, FB

Informatik, Arbeitsbereich Softwaretechnik, 1999 (siehe <http://www.jwam.de> )

- [Zül98] H. Züllighoven, Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz, dpunkt-Verlag, 1998
- [Zül99] Heinz Züllighoven, Guido Gryczan, Anita Krabbel, Ingrid Wetzel. Application-Oriented Software Development for Supporting Cooperative Work. In: Hans-Jörg Bullinger, Jürgen Ziegler (Hrsg.): Human-Computer Interaction. Ergonomics and User Interfaces. Volume 1. S. 1213-1217. 1999.