

Reihe: Telekommunikation @ Mediendienste · Band 11

Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, und Prof. Dr. Claudia Löbbecke, Köln

PD Dr.-Ing. habil. Martin Engelien
Dipl.-Inf. Jens Homann (Hrsg.)

Virtuelle Organisation und Neue Medien 2001

Workshop GeNeMe2001
Gemeinschaften in Neuen Medien

TU Dresden, 27. und 28. September 2001



JOSEF EUL VERLAG
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Virtuelle Organisation und Neue Medien 2001 / Workshop GeNeMe 2001 – Gemeinschaften in Neuen Medien – TU Dresden, 27. und 28. September 2001. Hrsg.: Martin Engeli; Jens Homann. – Lohmar; Köln: Eul, 2001

(Reihe: Telekommunikation und Mediendienste; Bd. 11)

ISBN 3-89012-891-2

© 2001

Josef Eul Verlag GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 90 10 6-6

Fax: 0 22 05 / 90 10 6-88

<http://www.eul-verlag.de>

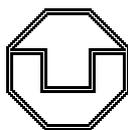
info@eul-verlag.de

Alle Rechte vorbehalten

Printed in Germany

Druck: RSP Köln

Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.



Technische Universität Dresden
Fakultät Informatik • Institut für Angewandte Informatik
Privat-Dozentur „Angewandte Informatik“

PD Dr.-Ing. habil. Martin Engelen,
Dipl.-Inf. Jens Homann
(Hrsg.)

Dresden, 27./28.09.2001

GENEME 2001

Gemeinschaften in Neuen Medien

*Workshop zu Organisation, Kooperation und
Kommunikation auf der Basis innovativer Technologien*

Forum für den Dialog zwischen Wissenschaft und Praxis



an der
Fakultät Informatik der Technischen Universität Dresden

gefördert von der Klaus Tschira Stiftung
gemeinnützige Gesellschaft mit beschränkter Haftung



am 27. und 28. September 2001
in Dresden

<http://pdai.inf.tu-dresden.de/geneme>
Kontakt: Thomas Müller (tm@pdai.inf.tu-dresden.de)

C.5. ProjectWeb – Eine web-basierte Projektumgebung

Bert F. Koch

Siemens AG

Gerald Eichler

T-Nova, Technologiezentrum

Dipl.-Inf. Mike Fischer, Dipl.-Inf. Falk Fünfstück, Prof. Dr. Heinrich Hußmann,

Ansgar W. Konermann, Dipl.-SWT. Anne Thomas

Fakultät Informatik, Technische Universität Dresden

1. Einleitung

Dieser Artikel beschreibt Ziele und Lösungsansätze des Projekts *ProjectWeb*, einer web-basierten Arbeitsumgebung für räumlich verteilte Projekte, das im Rahmen des diesjährigen Komplexpraktikums durch Studenten am Institut für Software- und Multimediatechnik der Fakultät Informatik (TU Dresden) bearbeitet wird [7].

Das Komplexpraktikum findet zum vierten Mal am Institut statt. Im Gegensatz zu den bisherigen Praktika wurde die Aufgabe für dieses Projekt aus realen Anforderungen an eine gemeinsame Arbeits-, Kommunikations- und Projektumgebung abgeleitet. Die Anforderungen stammen aus langjährigen Erfahrungen von Teilnehmern internationaler Forschungsprojekte und wurden von Teilnehmern des *AQUILA*-Projekts³⁰ [6] aufbereitet.

Das *AQUILA*-Projekt übernimmt daher quasi die Rolle des Auftraggebers, während Mitarbeiter des Instituts die fachliche Betreuung der Studenten übernehmen.

Neun Studenten arbeiten in einem Team weitestgehend selbstständig an der Entwicklung des *ProjectWebs*. Vorausgesetzt wird allerdings die Verwendung von modernen Methoden der Softwaretechnologie (z.B. UML [4], Entwurfsmuster [2, 8]) und ein geeignetes Vorgehensmodell.

Im Folgenden wird im Kapitel 2 zunächst auf den derzeitigen Status bei der Projektarbeit und denen sich daraus ergebenden grundlegenden Anforderungen an eine solche Umgebung eingegangen. Kapitel 3 analysiert diese Anforderungen im Detail mit dem Ziel, sie als Basis für eine flexible, erweiterbare und konfigurierbare Architekturlösung zu nutzen, auf die im Kapitel 4 in einigen Aspekten näher eingegangen wird. Der Artikel schließt mit einer Zusammenfassung der Resultate, dem aktuellen Stand der Realisierung und einem Ausblick auf weitere Vorhaben.

³⁰ Adaptive Resource Control for QoS Using an IP-based Layered Architecture, teilgefördert durch die EU unter der Vertragsnummer IST-1999-10077

2. Motivation und Problemstellung

Die Arbeit in einem internationalen, räumlich verteilten Projektteam – in AQUILA beispielsweise in sechs Ländern Europas – ist geprägt von einem starken Kommunikations-, Dokumentations- und Organisationsbedarf und entsprechendem Aufwand. Es besteht ein kontinuierlicher Diskussionsbedarf auch außerhalb der regelmäßig stattfindenden Projekttreffen, es werden sehr viele fachliche Dokumente erstellt, die zu bestimmten Arbeitsgruppen bzw. -paketen gehören und anderen Projektmitgliedern zugänglich gemacht werden müssen. Der Prozess der Dokumentenerstellung und dessen Verfügbarmachung ist dabei üblicherweise an die Einhaltung bestimmter organisatorischer Regeln gebunden.

Moderne Internetwerkzeuge sind heute in der Lage, Unterstützung bei einigen dieser Aufgaben zu liefern. Die Diskussion zwischen den Workshops findet beispielweise fast ausschließlich per E-Mail statt; zur Reduzierung des Verteilungsaufwandes werden mehrere themen- und gruppenorientierte Mail-Verteilerlisten eingerichtet und gepflegt.

Des Weiteren wird eine zentrale Dokumentenverwaltung für den Zugriff auf und die Archivierung von Projektdokumenten benötigt; im Falle AQUILA ist dies ein zentraler FTP-Server. Um ein schnelles Wiederauffinden der zahlreichen Dokumente und deren Kontextinformation zu gewährleisten, müssen zudem Regeln im Hinblick auf Zuordnung zu Arbeitspaketen, Dokumentennummer und -version eingehalten werden. Auch hierfür werden unterstützende Werkzeuge benötigt, z.B. zur automatischen Nummerierung und Versionierung von Dokumenten.

Das folgende, in AQUILA anzutreffende Szenario verdeutlicht das aufwendige Zusammenspiel der heterogenen Werkzeuge: Nach dem Anfertigen eines Dokuments muss der Projektmitarbeiter (Autor) zunächst mittels einer E-Mail einen Text in einer vorgegeben Syntax an einen zentralen Nummernserver schicken. Dieser Text enthält Informationen wie Titel, Autor, Erstellungsdatum, Dokumententyp, Verteilerkreis etc. Der Nummernserver schickt eine Antwort mit der endgültigen und im Projekt eindeutigen Dokumentennummer. Der Autor trägt diese Nummer in das neue Dokument ein und lädt es unter einem ebenfalls festgelegten Dateinamen in ein separates (incoming) Verzeichnis des Projektserver. Anschließend kündigt er die Verfügbarkeit des neuen Dokuments per E-Mail über die zugehörige Verteilerliste an. Schließlich ist der Projektmanager dafür verantwortlich, die Korrektheit der Dokumenteneigenschaften (Nummer, Version, Name und andere Formalien) zu überprüfen und die Datei in dem endgültigen Verzeichnis einer freigewählten Struktur des Projektserver abzulegen, für das nur er Schreibrechte hat. Zusätzlich werden die Kontextinformationen des

Dokuments in eine allen Projektteilnehmern zugängliche Indexdatei im HTML-Format aufgenommen, um Suche und Zugriff zu erleichtern.

Insgesamt werden also für die alltägliche Aufgabe der Registrierung und der Verfügbarmachung sehr viele verschiedene Werkzeuge eingesetzt, die unterschiedliche Anforderungen an deren Benutzung stellen. Zudem wird von den Projektmitarbeitern die genaue Kenntnis der vereinbarten Regeln verlangt sowie ein gewisses Maß an Disziplin bei deren Einhaltung, während der Projektmanager die Einhaltung der Regeln überprüfen und gegebenenfalls korrigierend eingreifen muss.

Es existieren bereits interessante Arbeitsumgebungen für verteilt arbeitende Projekte und ihre Teilnehmer, die diesen Prozess zumindest teilweise vereinfachen können. Ein Beispiel hierfür ist der web-basierte BSCW-Server der GMD im Hinblick auf zentrale Dokumentenverwaltung und -speicherung [1]. So ermöglicht der BSCW-Server, die Strukturierung eines Projektbereichs in verschiedene Unterbereiche, erlaubt die Versionierung und Kommentierung von Dokumenten und beinhaltet eine eingeschränkte Rechteverwaltung.

Die Anforderungen, die ein verteilt arbeitendes Projektteam an eine gemeinsame Arbeitsumgebung stellt, sind aber weitaus höher bzw. beschränken sich nicht auf die Dokumentenverwaltung:

- Der oben detailliert beschriebene Prozess der Dokumentenverfügbarmachung kann durch eine geeignete Dokumentenverwaltung teilweise automatisiert werden. Dies betrifft z.B. das Erstellen des Registrierungsauftrags, die Zuordnung zu Arbeitspaketen inklusive Rechtevergabe sowie die Indexierung der Dokumentendaten.
- Es wird eine anspruchsvolle Benutzerverwaltung benötigt, die es gestattet, den Projektteilnehmern Rollen (z.B. „Projektmanager“, „Mitarbeiter“, etc.) zuzuordnen. Des weiteren ist es nützlich, Kontaktinformationen über die Projektmitglieder in einer Art Adressbuch zu verwalten.
- Eine Rechteverwaltung regelt die Zuordnung von Rechten zu Rollen einerseits und überwacht andererseits Zugriffe, z.B. auf Dokumente, für die bestimmte Rechte benötigt werden. So würde die Rolle „Projektmanager“ beispielweise weitreichende Rechte auf allen Dokumenten, dem Adressbuch usw. mit einschließen, während ein einzelner „Mitarbeiter“ tätigkeitsorientiert angepasste Rechte auf den Diensten der Arbeitsumgebung besitzt. Die Rechteverwaltung setzt das Vorhandensein einer Anmeldeprozedur voraus.
- Die Arbeit in Forschungsprojekten hat häufig vertraulichen Charakter. Neben der Rechteverwaltung werden also auch Sicherheitsmechanismen benötigt. Dies beinhaltet die Benutzung von verschlüsselten Verbindungen, z.B. via SSL.

- Neben dem projektinternen Bereich soll die Arbeitsumgebung auch einen öffentlichen Bereich anbieten, der frei zugängliche, das Projekt repräsentierende Informationen, z.B. in Form einer strukturierbaren Homepage, bereitstellt. Es sollte auch möglich sein, Dokumente und andere Informationen aus dem projektinternen in den öffentlichen Bereich zu „exportieren“.
- Es sollte möglich sein, externe Werkzeuge, z.B. den Mailverteiler, in die Arbeitsumgebung einzubinden, indem z.B. die registrierten Nutzer bestimmten Verteilerlisten zugeordnet werden können.
- Neben der reinen Dokumentenverwaltung werden oftmals zusätzliche Dienste, z.B. kommentierbare Literaturlisten, E-Mailarchive, Zeitpläne, etc. benötigt. Beispielweise sind sehr viele wichtige Detailinformationen nicht in Dokumenten sondern in E-Mails enthalten. Eine Archivierungsfunktion wäre wünschenswert.
- Die Arbeitsumgebung sollte eine flexible, konfigurierbare Suchfunktion anbieten.

Es ließen sich noch eine Reihe weiterer möglicher Funktionalitäten oder Dienste auflisten. Wesentlich ist, dass die Mitarbeiter räumlich verteilt arbeitender Projekte und Forschergruppen eine integrierte Arbeitsumgebung benötigen, die die verschiedenen projektunterstützenden Dienste bündelt und unter einer projektspezifischen, einheitlichen, web-basierten und somit für einen Standardbrowser geeigneten Benutzungsoberfläche anbietet. Organisatorischer Aufwand ließe sich so reduzieren, die Einarbeitung für neue Projektmitglieder vereinfachen und die Einhaltung notwendiger Regeln automatisieren.

Darüber hinaus ist zu beachten, dass unterschiedliche Projekte recht unterschiedliche Anforderungen an das Vorhandensein und die Arbeitsweise bestimmter Dienste, die Strukturierung des Projekts, das Vorhandensein bestimmter Rollen, aber auch im Hinblick auf projektspezifische Layouts, auf Internationalisierung bzw. Mehrsprachigkeit usw. haben.

Angestrebt wird daher eine generische Lösung – ProjectWeb genannt – aus der nach projektabhängiger Konfiguration verschiedene Instanzen für konkrete Projekte erzeugt werden können. Eine Beispielinstantz für eine solche Arbeitsumgebung wird für das AQUILA-Projekt erstellt.

Benötigt wird also eine hinreichend flexible und modulare Architektur, die aufgrund genormter Schnittstellen eine weitreichende Konfiguration (teilweise auch zur Laufzeit des Projekts) sowie Erweiterbarkeit um neue Dienste bzw. Werkzeuge sicherstellt.

3. Anforderungen an das ProjectWeb

Die in Kapitel 2 nur grob beschriebenen funktionalen Anforderungen an eine Projektumgebung z.B. für AQUILA werden im folgenden für das generische ProjectWeb detailliert und aus der Sicht der Auftraggeber beschrieben mit dem Ziel, diese Anforderungen in eine geeignete Architektur einfließen zu lassen. Die Anforderungen beziehen sich dabei auf zwei wesentliche Teilaspekte: die konkrete Laufzeitinstanz (Abschnitt 3.1) einer solchen web-basierten Arbeitsumgebung sowie deren Konfiguration (Abschnitt 3.2). Letzterer Teilaspekt betont grundlegende globale Ziele:

- Die zu entwickelnde Software soll wiederverwendbar sein. Arbeitsumgebungen werden in verschiedensten Projekten benötigt und sollen möglichst einfach an die Erfordernisse des jeweiligen Projekts angepasst werden können.
- Die Wiederverwendung soll dabei weitestgehend über eine Konfiguration möglich sein. Eine Änderung von Programmcode soll bei der Anpassung der unterstützten Dienste (z.B. Dokumentenverwaltung, Literaturverzeichnis, Adressbuch usw.) an ein bestimmtes Projekt nicht notwendig sein.

Die funktionalen Anforderungen an das System spiegeln sich in den zu unterstützenden Anwendungsfällen wider, die hier mittels der UML-Notation [4] für Anwendungsfall-Diagramme beschrieben sind. Diese Notation ist einfach gehalten und eignet sich gut für die Auftraggeber-Kunden-Kommunikation.

3.1 Laufzeitinstanz

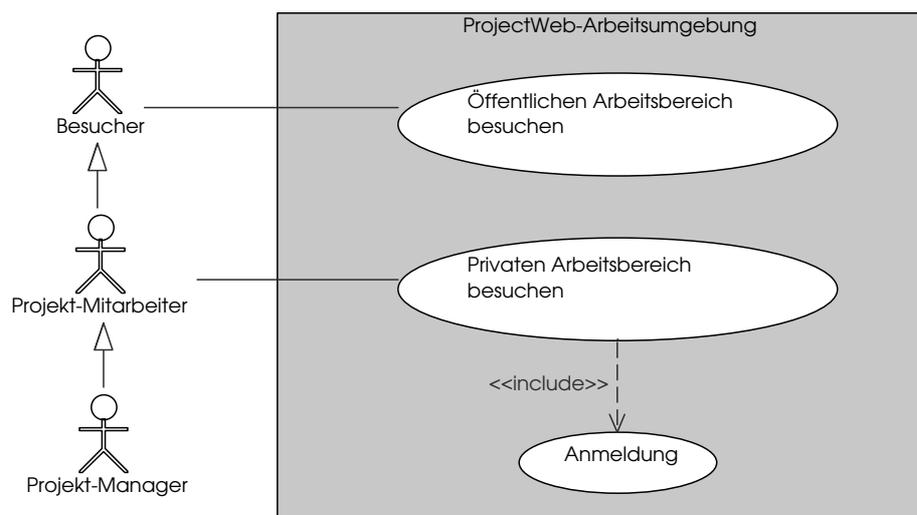


Abbildung 2: Anwendungsfall-Diagramm für eine ProjectWeb-Arbeitsumgebung

Die Arbeitsumgebung soll aus zwei wesentlichen Bereichen bestehen, einem öffentlich zugänglichen (projektexternen) und einem privaten (projektinternen) Arbeitsbereich. Die elementaren Anwendungsfälle sowie daran beteiligte Akteure werden in Abbildung 2 dargestellt.

3.1.1 Anwendungsfall „Öffentlichen Arbeitsbereich besuchen“

Beliebige Interessenten am Projekt können den öffentlichen Arbeitsbereich besuchen, hier symbolisiert durch einen Akteur „Besucher“. Dabei umfasst der Anwendungsfall „Öffentlichen Arbeitsbereich besuchen“ folgende Möglichkeiten:

- ◆ Aufruf öffentlicher Web-Seiten

Dies beinhaltet den Aufruf statischer Web-Seiten, sowie von Web-Seiten mit dynamisch generierten öffentlichen Inhalten. Diese Inhalte können z.B. Informationen über Ziele und Struktur des Projekts, Informationen über die beteiligten Projektpartner und Projekt-Mitarbeiter sowie weitere öffentliche Informationen umfassen.

- ◆ Suche über der Menge der öffentlichen Projektinformationen

Ein wesentlicher Bestandteil einer Arbeitsumgebung ist ein globaler Suchdienst, der basierend auf angegebenen Suchbegriffen die Gesamtheit der Arbeitsumgebung auf zutreffende Informationen durchsucht. Die von einem Besucher ausgelöste Suche ist selbstverständlich auf die öffentlichen Projektinformationen/-dokumente beschränkt.

- ◆ Herunterladen öffentlicher Projektdokumente

Ein Besucher kann auch Dienste der Arbeitsumgebung nutzen. Als Beispiel hierfür sei die Dokumentenverwaltung genannt. Die Nutzung dieses Dienstes ist für den Besucher jedoch auf die Suche und das Herunterladen von öffentlichen Projektdokumenten beschränkt. Ein Zugriff auf private Dokumente bzw. das Ablegen neuer Dokumente ist einem Besucher nicht möglich. Diese Funktionalität wird vor dem Besucher versteckt bzw. als nicht ausführbar gekennzeichnet.

3.1.2 Anwendungsfall „Privaten Arbeitsbereich besuchen“

Der Zugriff auf den privaten Arbeitsbereich bleibt Mitarbeitern des Projekts vorbehalten. Bei einer Anmeldung (Anwendungsfall „Anmelden“) authentifiziert sich der jeweilige Projektmitarbeiter mit Nutzernamen und Passwort gegenüber der Arbeitsumgebung. Nach erfolgreicher Anmeldung stehen einem Projektmitarbeiter neben den Informationen des öffentlichen Bereichs zusätzlich alle Informationen und Dienste des privaten Arbeitsbereichs (verfeinert in Abbildung 3 und Abbildung 4) zur Verfügung.

Innerhalb des privaten Arbeitsbereichs unterscheiden sich die Projektmitarbeiter durch die ihnen in Abhängigkeit von der Projektstruktur zugeordneten Rollen und Rechte. In den Anwendungsfall-Diagrammen ist dies beispielhaft durch die Unterscheidung von

Projektmitarbeitern und Projektmanager illustriert. Eine ProjectWeb-Arbeitsumgebung kann jedoch noch viele weitere Rollen kennen, z.B. sind dies Rollen, die Mitarbeiter als Verantwortliche für einzelne Arbeitspakete oder als Verantwortliche für Teile der Arbeitsumgebung auszeichnen.

Nachfolgend wird die erste von zwei Gruppen verfeinerter Anwendungsfälle, dargestellt in Abbildung 3, beschrieben. An diesen sind Projektmitarbeiter, inkl. Projektmanager als spezieller Projektmitarbeiter, beteiligt. Die Anwendungsfälle beschreiben projektbezogene Aufgaben bei der Nutzung der Arbeitsumgebung.

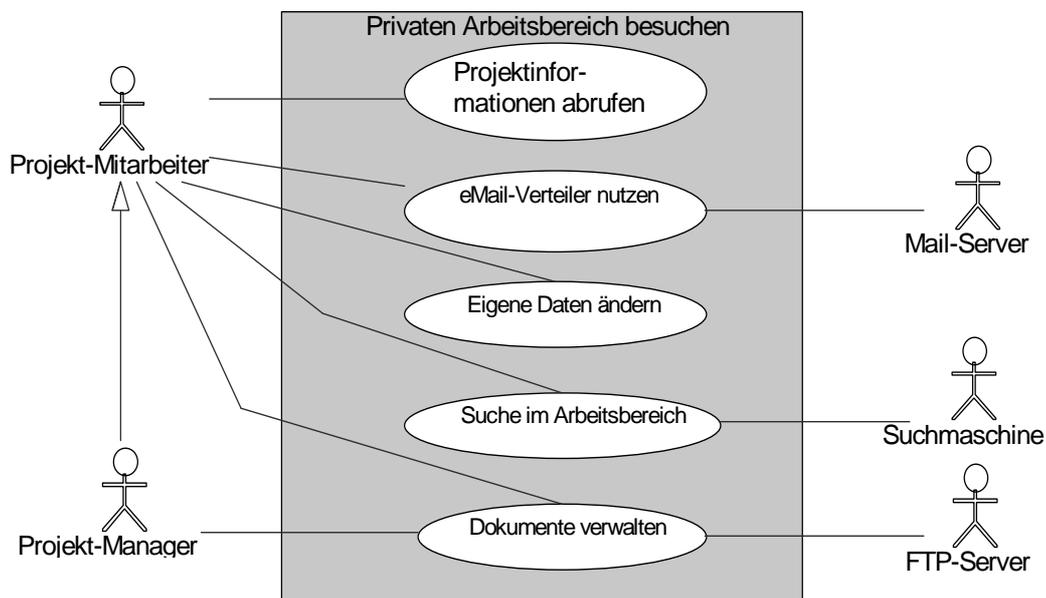


Abbildung 3: Anwendungsfall-Diagramm für den privaten Arbeitsbereich (1)

Anwendungsfall „Dokumente verwalten“

Die Verwaltung der im Rahmen des Projekts erstellten Dokumente bildet eine Kernfunktionalität der ProjectWeb-Arbeitsumgebung. Der Anwendungsfall „Dokumente verwalten“ beinhaltet die folgenden, weiteren Anwendungsfälle:

- ◆ Dokumente ablegen

Projektmitarbeiter können Dokumente beliebiger Formate in der Arbeitsumgebung ablegen, die dabei vom System den Status „ungeprüft“ erhalten. Zu deren physikalischen Speicherung kann die Arbeitsumgebung auf einen Dokumenten-Server zurückgreifen. Neben der reinen Ablage ist die Angabe von Beschreibungsinformationen zu den Dokumenten notwendig. Derartige Informationen umfassen Autor, Titel, Prüfstatus, Kurzfassung, Datum der Dokumentenerstellung, Dokumentennummer usw. Zum Teil kann das System dem Benutzer eine Voreinstellung dieser Informationen anbieten, z.B. basierend

auf der aktuellen Zeit (Datum, Uhrzeit) und den Anmeldeinformationen des Benutzers, der das Dokument ablegen möchte.

◆ Dokumente bestätigen

Der Prüfstatus von Dokumenten kann nur durch einen berechtigten Projektmitarbeiter, z.B. dem Projektmanager, verändert werden. Erst nachdem der Status eines als öffentlich deklarierten Dokuments auf „geprüft“ geändert wurde, ist das Dokument in den entsprechenden Bereichen der Arbeitsumgebung „sichtbar“.

◆ Dokumentenbeschreibungsinformationen ändern

Eine berechnigte Person (z.B. der Akteur „Projektmanager“) darf Beschreibungsinformationen für Dokumente ändern. Dies betrifft u.a. den Dokumententitel, die Kurzfassung oder die Sichtbarkeit (öffentlich, privat) des Dokuments.

◆ Dokumente herunterladen

Projektmitarbeiter können Dokumente herunterladen und auf ihrem lokalen Rechner abspeichern. Dabei haben die Projektmitarbeiter das Dokument zuvor über eine geeignete Suchfunktion in der Dokumentenverwaltung gefunden. Die Dokumentenverwaltung unterstützt den Benutzer, indem eine Sortierung der Projektdokumente nach unterschiedlichen Kriterien, wie z.B. Datum der Dokumentenerstellung, Titel, Autor, Dokumentennummer, Prüfstatus usw., angeboten wird.

◆ Snapshot erstellen

Vielfach (z.B. bei Projekttreffen) benötigen Projektmitarbeiter lesenden Zugriff auf die Dokumente des Projekts; so auch in Situationen, in denen kein Netzzugang vorhanden ist. Dazu kann ein Mitarbeiter vorab den aktuellen Zustand der Dokumentenverwaltung kopieren (*Snapshot*). Die Dokumente werden dabei in einer Verzeichnisstruktur auf seinem lokalen Rechner (z.B. einem Notebook) abgelegt. Zusätzlich wird ein Übersichtsdokument in Form einer HTML-Seite erstellt, das das Auffinden von Dokumenten erleichtert.

Anwendungsfall „Projektinformationen abrufen“

Dieser Anwendungsfall umfasst eine Menge weiterer Anwendungsfälle, die eine Abfrage von verschiedensten Projektinformationen betreffen. Exemplarisch seien die nachfolgenden Möglichkeiten aufgeführt:

◆ Adressbuch abrufen

Ein Projektmitarbeiter informiert sich im projektweiten Adressbuch über Kontaktinformationen der beteiligten Mitarbeiter bzw. Institutionen.

- ◆ Meilensteinplan abrufen
Projektmitarbeiter können den Zeitplan des Projekts einsehen.
- ◆ Meetingkalender abrufen
Ein Meetingkalender enthält Informationen (z.B. Ort und Zeit) der Projekttreffen. Ein Projektmitarbeiter kann Detailinformationen wie z.B. Hinweise für die Anreise, Informationen über die Unterkunft und weiteres abrufen.

Anwendungsfall „E-Mail-Verteiler nutzen“

In Projekten ist es vielfach notwendig, E-Mail-Verteilerlisten zu definieren. E-Mail-Verteiler erlauben es, Gruppen von Projektmitarbeitern via E-Mail zu erreichen. Dieser Anwendungsfall umfasst daher die Möglichkeit, bestehende Verteilerlisten zu recherchieren, die eingetragenen Projektmitarbeiter abzufragen sowie eine E-Mail an eine oder mehrere Verteilerlisten zu senden. Für die zu erstellende Software ist es nicht notwendig, einen E-Mail-Verteiler bzw. -Server selbst zu realisieren. Vielmehr ist es hier sinnvoll, auf bestehende Software, z.B. Majordomo, zurückzugreifen (dargestellt durch den Akteur „Mail-Server“) und diese in die einheitliche Oberfläche einer ProjectWeb-Arbeitsumgebung zu integrieren.

Anwendungsfall „Eigene Daten ändern“

Ein Projektmitarbeiter kann bei Bedarf eigene Kontaktinformationen, wie z.B. Anschrift seiner Institution, Telefon- und Faxnummer oder sein Passwort für den Zugang zur Arbeitsumgebung, selbstständig abändern.

Anwendungsfall „Suche im Arbeitsbereich“

Für einen an der Arbeitsumgebung angemeldeten Projektmitarbeiter steht eine globale Suchfunktion zur Verfügung. Im Gegensatz zu einem projektexternen Besucher, der über die Suchfunktion nur öffentliche Informationen auffinden kann, erlangt ein Projektmitarbeiter Zugriff auf alle Dokumente und Informationen im Arbeitsbereich. Nicht notwendig erscheint es dabei, die Leserechte zwischen Projektmitarbeitern mit unterschiedlichen Projektrollen zu unterscheiden, da üblicherweise alle Projektteilnehmer unabhängig von ihrer Rolle Zugang mit Leseberechtigung zu allen Projektinformationen und -dokumenten haben. Die Berechtigungen, um Dokumente bzw. Informationen zu modifizieren, erhält ein Projektmitarbeiter jedoch nur in Abhängigkeit von den von ihm im Projekt zu erledigenden Arbeitsaufgaben.

Im folgenden wird die zweite Gruppe verfeinerter Anwendungsfälle, dargestellt in Abbildung 4, beschrieben. An diesen sind nur besonders berechtigte Projektmitarbeiter (hier dargestellt als Akteur „Projektmanager“) beteiligt, die administrative Aufgaben bei der Konfiguration und Verwaltung der Arbeitsumgebung übernehmen.

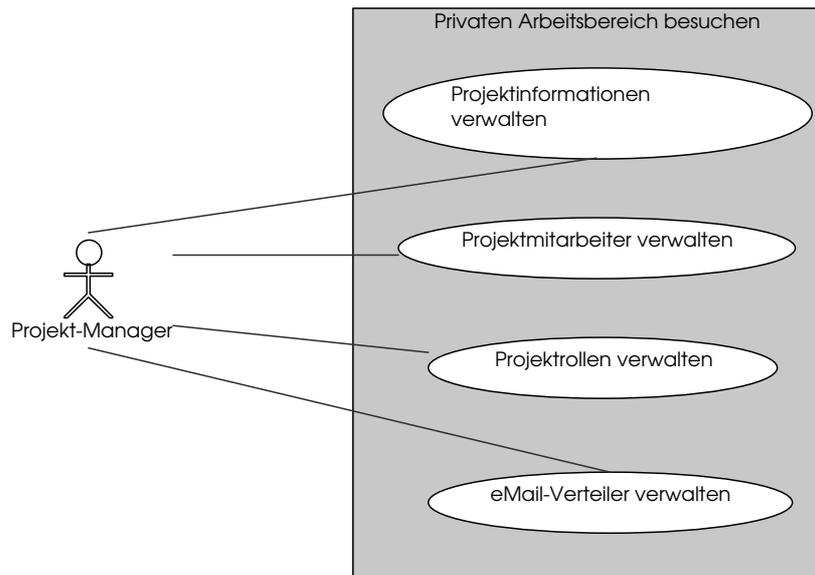


Abbildung 4: Anwendungsfall-Diagramm für den privaten Arbeitsbereich (2)

Anwendungsfall „Projektinformationen verwalten“

Der Anwendungsfall umfasst das Erstellen, Ändern und Löschen von statischen Web-Seiten im privaten und öffentlichen Bereich der Arbeitsumgebung durch einen entsprechend privilegierten Projektmitarbeiter. Dieser Anwendungsfall umfasst ebenfalls die Verwaltung spezieller Dienste wie Adressbuch, Meilensteinplan und Meetingkalender.

Anwendungsfall „Projektmitarbeiter verwalten“

Dem Projektmanager obliegt es, mit Hilfe der in der Arbeitsumgebung integrierten Nutzerverwaltung, die Anmeldeinformationen der Projektmitarbeiter zu verwalten. Die verwalteten Informationen bilden die Grundlage für die Anmeldung der Projektmitarbeiter am System sowie für den Adressbuch-Dienst. Der Anwendungsfall beinhaltet folgende Möglichkeiten:

- ◆ Projektmitarbeiter hinzufügen
- ◆ Projektmitarbeiter löschen
- ◆ Projektmitarbeiterinformationen ändern

Anwendungsfall „Projektrollen verwalten“

Jedem Projektmitarbeiter sind eine oder mehrere Rollen zugeordnet, die letztlich eine Menge von Rechten bündeln und für den einzelnen Projektmitarbeiter definieren, welche Berechtigungen er in der Arbeitsumgebung besitzt. Es ist die Aufgabe des Projektmanagers, solche Rollen zu verwalten. Folgende Funktionen zur Verwaltung der Rollen müssen vom System unterstützt werden:

- ◆ Projektrolle hinzufügen
- ◆ Projektrolle löschen
- ◆ Projektrolle ändern
- ◆ Zuordnung von Projektrollen zu Projektmitarbeitern

Anwendungsfall „E-Mail-Verteiler verwalten“

Der Projektmanager verwaltet den E-Mail-Verteiler bzw. dessen Verteilerlisten. Eine Verteilerliste besitzt jeweils eine E-Mail-Adresse und eine Liste von Projektmitarbeitern, die dafür registriert sind. Wird an die Verteilerliste eine Nachricht gesendet, dann werden die aktuellen E-Mail-Adressen der registrierten Projektmitarbeiter ermittelt, und die Nachricht wird entsprechend weitergeleitet. Die Arbeitsumgebung unterstützt dabei Funktionen wie:

- ◆ E-Mail-Verteilerliste hinzufügen
- ◆ E-Mail-Verteilerliste löschen
- ◆ E-Mail-Verteilerliste modifizieren

3.2 Konfiguration

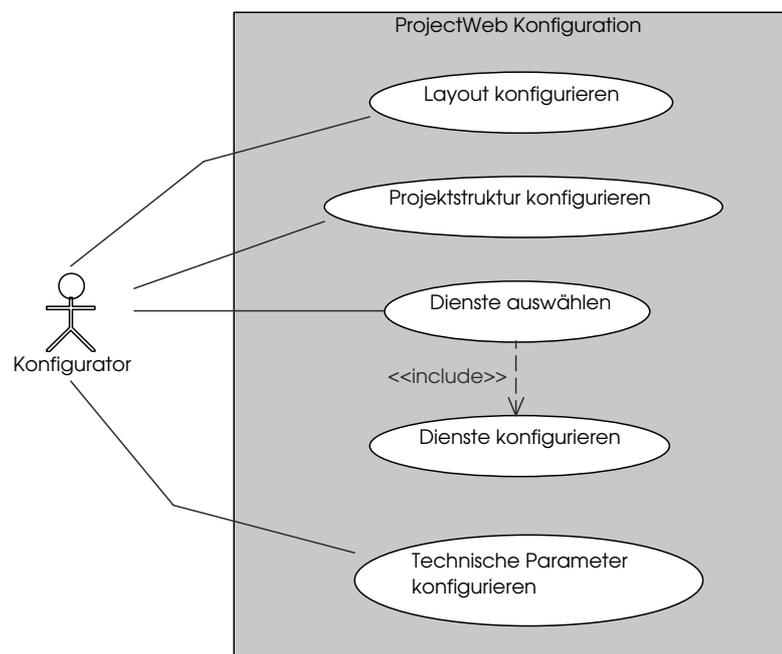


Abbildung 5: Anwendungsfall-Diagramm für die ProjectWeb-Konfiguration

Die Konfiguration des ProjectWebs hat eine Anpassung der Arbeitsumgebung an die Erfordernisse und Rahmenbedingungen konkreter Projekte zum Ziel. Die dafür wesentlichen Anwendungsfälle sind in der Abbildung 5 dargestellt.

Anwendungsfall „Layout konfigurieren“

Ziel der Konfiguration des Layouts ist es, die Arbeitsumgebung an die Erfordernisse des konkreten Projekts anzupassen. Beispiele für Konfigurationsparameter sind z.B.:

- URLs von StyleSheet-Dateien, in denen Farben, Schriftarten und -größen und Web-Seiten-Hintergründe definiert werden können,
- Einstellungen für die Sprache der Projektumgebung,
- Definition von Grafiken für Navigations-Links,
- Angabe von Projektlogos und ähnliches.

Die Konfiguration des Layouts ermöglicht es somit, die Arbeitsumgebung an ein vorgegebenes „corporate design“ anpassen zu können.

Anwendungsfall „Projektstruktur konfigurieren“

Jedes Projekt besitzt eine vorgegebene Projektstruktur. Beispielsweise ist ein Projekt in Arbeitspaketgruppen und einzelne Arbeitspakete unterteilt. Diese Struktur ändert sich erfahrungsgemäß im laufenden Projekt nur sehr selten. Daher ist es Teil der Konfiguration einer ProjectWeb-Arbeitsumgebung, diese Struktur festzulegen. Bedeutung erlangt diese Strukturierung bei der Rechte- bzw. Rollenvergabe an die einzelnen Projektmitarbeiter, die in einem oder mehreren Arbeitspaketen mitarbeiten. So muss für ein Arbeitspaket z.B. die Rolle „Leiter des Arbeitspakets“ an einen der Mitarbeiter vergeben werden.

Anwendungsfälle „Dienste auswählen / Dienste konfigurieren“

Die Konfiguration der Arbeitsumgebung beinhaltet die Auswahl der notwendigen Dienste. Einige Dienste, insbesondere Infrastrukturdienste wie z.B. die Rollen- und Nutzerverwaltung, sind zwingend notwendig und müssen deshalb immer ausgewählt werden. Realisierungen derartiger Dienste werden im folgenden als obligatorische Funktionsmodule bezeichnet. Andere Dienste, im folgenden auch optionale Funktionsmodule genannt, können je nach Bedarf ausgewählt werden. Dies sind z.B. die Dokumentenverwaltung, der E-Mail-Verteiler, das Adressbuch, der Meilensteinplan, der Meetingkalender und weitere.

Ausgewählte Funktionsmodule bedürfen selbst einer weiteren Konfiguration. Dies wird durch den Anwendungsfall „Dienste konfigurieren“ verdeutlicht. Dies betrifft z.B. die Belegung von Freitextfeldern in der Nutzerverwaltung, die Konfiguration von Grafiken für Links der Funktionsmodule usw.

Anwendungsfall „Technische Parameter konfigurieren“

Die Angabe technischer Parameter dient dazu, bestehende Dienste, wie z.B. Suchmaschinen, Datenbank- oder FTP-Server und Mail-Server, in eine ProjectWeb-Arbeitsumgebung einzubinden, um sie dort unter einer einheitlichen Oberfläche – im „corporate design“ – den Projektmitarbeitern anbieten zu können.

Die oben beschriebenen funktionalen Anforderungen sind keineswegs vollständig und endgültig. Sie werden jedoch aus der Sicht des AQUILA-Projekts als diejenigen Anforderungen betrachtet, die bei einer Umsetzung in der ProjectWeb-Software für einen praktischen Einsatz die minimal notwendige Funktionalität darstellen. Ausgehend von dem praktischen Einsatz der Software ist die Definition weiterer funktionaler Anforderungen zu erwarten.

4. Architektur des ProjectWebs

Eine der zentralen Forderungen an das zu erstellende System ist die Erweiterbarkeit um zusätzliche Funktionen. Um dies zu gewährleisten, wurde bei der Architektur von ProjectWeb ein weitgehend modularer Ansatz gewählt.

4.1 Grobarchitektur

Abbildung 6 zeigt die Architektur des Systems als Blockdiagramm. Der sogenannte *Instance Manager* (Abschnitt 4.4) bildet als systemweite Informations- und Verwaltungszentrale das Kernsystem der Anwendung und verwaltet die Instanzen der einzelnen Funktionsmodule. Diese beinhalten die eigentlichen Systemfunktionen und machen sie über eine definierte Schnittstelle im System verfügbar. Die Mehrzahl der während des Praktikums implementierten Funktionsmodule sind obligatorisch, d.h. sie stellen notwendige Infrastrukturdienste innerhalb der Anwendung zur Verfügung, die nicht im Zuge der Konfiguration deaktiviert werden dürfen (vgl. Abschnitt 3.2).

Die unterste Schicht realisiert die persistente Datenhaltung für das System. Oberhalb vom Instance Manager und den Funktionsmodulen ist die grafische Schnittstelle zum Nutzer angesiedelt. Der *Config Manager* schließlich stellt die Schnittstelle zwischen den in einer Textdatei abgelegten Konfigurationsinformationen und den übrigen Systemteilen dar.

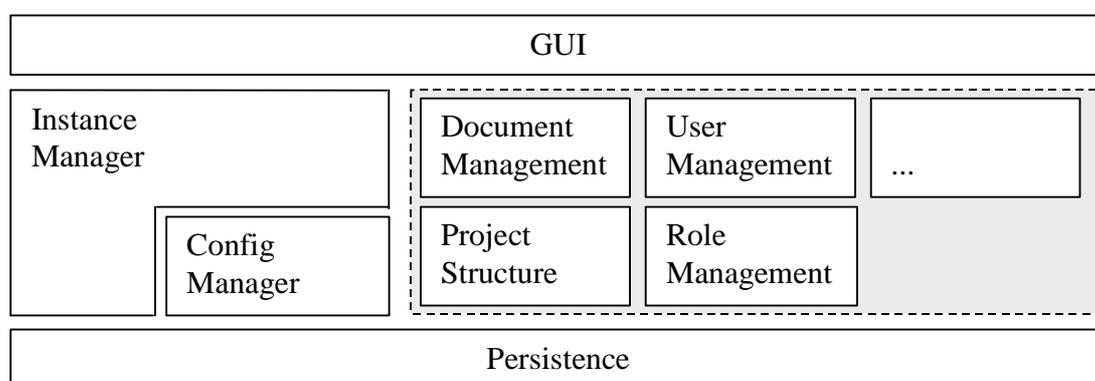


Abbildung 6: Grobarchitektur ProjectWeb – Blockdiagramm

4.2 Verwendete Technologien

Aufgrund der Komplexität des geforderten Systems wurde auf vorhandene, effiziente Technologien zur Realisierung serverseitiger Anwendungen zurückgegriffen.

Als Implementationssprache wurde Java [9] gewählt. Dadurch wird die Verwendung zahlreicher weiterer auf Java basierender Technologien möglich:

Enterprise Java Beans (EJBs) [3, 10, 11] sind serverseitige Softwarekomponenten zur Realisierung von Datenpersistenz (sog. Entity Beans) und Anwendungslogik (sog. Session Beans). Servlets [12] und die darauf basierenden Java Server Pages (JSPs) [5, 13] sind Softwarekomponenten für Web-Server, die dynamische Inhalte generieren. Im Rahmen des Praktikumsprojekts besteht dieser dynamische Inhalt aus HTML-Seiten [14], deren Erscheinungsbild teilweise mit Hilfe von Cascading Style Sheets (CSS) [15] festgelegt wird. Für die verschlüsselte Datenübertragung wird auf die in praktisch allen Web-Servern vorhandene Implementierung von Secure Sockets Layer (SSL) [16, 17] zurückgegriffen.

4.3 Funktionsmodule

Alle Funktionsmodule dienen der modularen Erweiterung des Kernsystems um neue Funktionen. Um diese Erweiterbarkeit und eine problemlose Interaktion mit dem Kernsystem sicherzustellen, wurde vom Praktikumsteam eine Implementierungsrichtlinie für Funktionsmodule erarbeitet und als verbindlicher Standard vereinbart.

Abbildung 7 zeigt in einer an die UML angelehnten Notation den schematischen Modulaufbau. Kreise repräsentieren Schnittstellen, Rechtecke repräsentieren Klassen [4]. Funktionsmodule müssen insbesondere ihre Benutzungsschnittstelle und die Persistenzschicht selber bereitstellen, sofern diese Elemente notwendig sind. Zwischen diesen zwei Schichten wird die eigentliche Funktionalität von einer oder mehreren Implementierungsklassen realisiert, die im vorliegenden Fall häufig als Session Beans ausgeführt sind. Die GUI-Elemente (JSPs) rufen über die ausgezeichnete öffentliche Schnittstelle des Moduls (*Module Interface*) Funktionen innerhalb des Moduls auf.

Darüber hinaus sind einige Infrastruktur-Klassen notwendig, um ein Modul an das Kernsystem anzukoppeln. Der *Module Descriptor* bildet den initialen Einsprungspunkt des Moduls, über den der Instance Manager mit dem Modul kommunizieren kann. Insbesondere bietet der Module Descriptor Funktionen für die Erzeugung einer neuen Proxy-Instanz des Moduls. Zusätzlich lässt sich die Menge der Rechte ermitteln, die von diesem Modul definiert werden.

Der *Proxy Manager* (Abbildung 7) dient schließlich dem vereinfachten Zugriff auf das Funktionsmodul von anderen Modulen oder der Benutzungsschnittstelle aus.

4.4 Instance Manager

Der Instance Manager und seine Hilfsklassen bilden das Kernsystem, an das sich Funktionsmodule über eine definierte Schnittstelle ankoppeln können. Er lädt und initialisiert alle Funktionsmodule, die in der Dienstekonfiguration aktiviert sind (vgl. Abschnitt 3.2). Außerdem verwaltet er alle HTTP-Sitzungen und die Instanzen der Modul-Proxys. Hierbei wird für jedes Modul in jeder Sitzung eine eigene Instanz des entsprechenden Proxys erzeugt und im Sitzungskontext abgelegt. Dadurch können die Module mit den Proxys als typischerer Abstraktion einer HTTP-Sitzung arbeiten.

Daneben bietet der Instance Manager Zugang zu systemweit notwendigen Informationen. Besonders wichtig ist hier die Menge aller Rechte, mit denen Funktionen des Gesamtsystems geschützt werden, sowie korrekt in die konfigurierte Sprache übersetzte Textbausteine für die Benutzungsschnittstelle.

Im Rahmen des Praktikumsprojekts wurde vereinfachend festgelegt, dass je ausgeführter Anwendung genau ein Projekt verwaltet wird. Um den Aufwand für die Softwareinstallation bei einer großen Anzahl zu verwaltender Projekte gering zu halten, ist bei zukünftigen Weiterentwicklungen eine Aufhebung dieser Begrenzung wünschenswert. Im Instance Manager werden in diesem Fall vermutlich die größten Erweiterungen notwendig sein.

4.5 Benutzungsschnittstelle (Web-GUI)

Die Interaktion zwischen den Funktionsmodulen und dem Nutzer erfolgt über eine web-basierte Schnittstelle.

Optisch gliedert sich diese Schnittstelle im Wesentlichen in zwei Bereiche (Abbildung 9). Auf der linken Seite befindet sich der Navigationsbereich. Dort werden abhängig von den aktivierten Funktionsmodulen und den Rechten des aktuellen Nutzers Links zu den Einstiegsseiten der Module in Form eines Menüs angezeigt. Bei Betätigung eines Links wird die Einstiegsseite des entsprechenden Moduls im Hauptbereich rechts angezeigt.

Auch logisch ist die Benutzungsoberfläche zweigeteilt. Die im öffentlichen Bereich angebotenen Funktionen dürfen von jedem Nutzer des Systems verwendet werden. Alle Funktionen im privaten Bereich dagegen sind ausschließlich den im System registrierten Nutzern nach erfolgter Authentifizierung vorbehalten. Diese erfolgt durch die Eingabe des Nutzernamens und des dazu gehörenden Passworts innerhalb des Navigationsbereichs.

Die geeignete Darstellung der Benutzungsschnittstelle fällt in den Verantwortungsbereich des Modulentwicklers. Vom technischen Standpunkt aus gesehen kann ein beliebiger Aufbau dieser Schnittstelle realisiert werden. Innerhalb der Praktikumsgruppe wurde jedoch zur Vereinheitlichung des Erscheinungsbildes eine Vereinbarung über die Gestaltung der Programmoberfläche getroffen. Diese besagt, dass im oberen Teil jeweils eine Zeile mit modulspezifischen Schaltflächen angezeigt wird und die eigentlichen Ein- und Ausgabeelemente grundsätzlich im unteren Teil angeordnet werden.

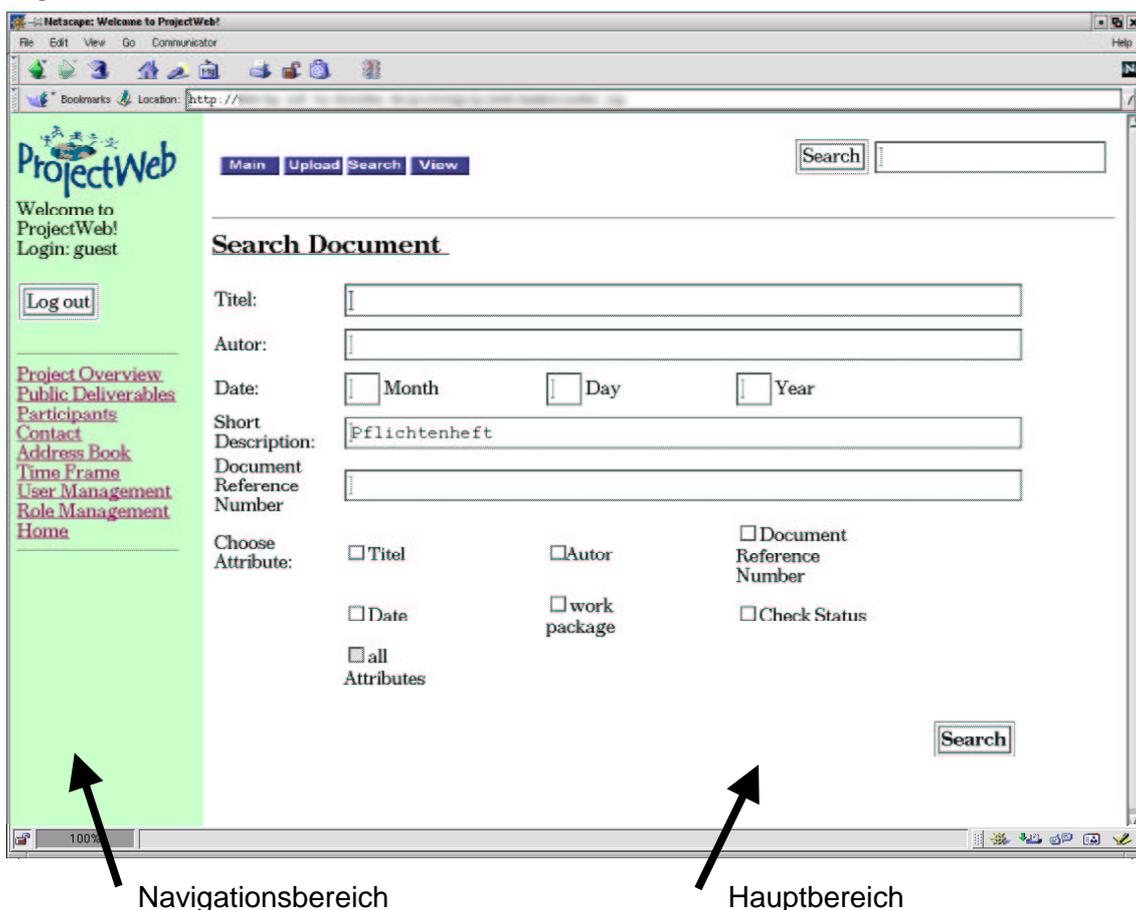


Abbildung 9: Benutzungsoberfläche am Beispiel der globalen Suchfunktion nach bereits erfolgter Anmeldung am System

Für alle während des Praktikumsprojekts realisierten Funktionsmodule wird die Benutzungsschnittstelle durch eine Menge von JSPs gebildet. Das Erscheinungsbild wird dabei teilweise über eine StyleSheet-Datei eingestellt (vgl. Abschnitt 3.2). Weitere Angaben, beispielsweise über die URLs der zu verwendenden Logogramme und Grafiken für Navigations-Links, werden aus der Konfigurationsdatei entnommen (Abschnitt 4.7).

Um die Anwendungslogik von der Darstellung zu trennen, wurde auf das bekannte Model-View-Controller-Muster (MVC) für Web-Komponenten zurückgegriffen [20, 21]. Dies ermöglicht beispielsweise die Anpassung der ausschließlich zur Darstellung verwendeten JSPs auch durch bei der Programmierung weniger erfahrene Web-Designer.

Besondere Beachtung muss dem Schutz von statischen Dokumenten im privaten Bereich gewidmet werden. Da diese in kein Funktionsmodul eingebunden sind, können sie nicht selbstständig eine Rechteprüfung durchführen. Daher wird diese Prüfung mit Hilfe eines Servlets realisiert, das Zugriffe auf einen Teil des URL-Namensraumes der Anwendung verhindert und vor dem Ausliefern des angeforderten Dokuments die notwendige Prüfung durchführt. Die korrekte Konfiguration dieses Servlets ist von entscheidender Bedeutung für den Schutz der privaten Dokumente. Der Sicherheitsmechanismus des Web-Containers [10] kann hier nicht verwendet werden, da dieser aufgrund fehlender Schnittstellen nicht mit der systemeigenen Rechteverwaltung zusammen arbeiten kann.

4.6 Persistenzschicht

Um die von einem Funktionsmodul verwalteten Daten dauerhaft zu speichern, muss jedes dieser Module einen eigenen geeigneten Persistenzmechanismus bereitstellen.

Die im Rahmen des Praktikums implementierten Funktionsmodule stützen sich bei der dauerhaften Speicherung ihrer Daten auf Entity Beans [10, 11]. Bei deren Implementierung wurden zur Performance-Steigerung sogenannte Value Objects eingesetzt [18]. Durch deren konsequente Verwendung lässt sich die Anzahl der entfernten Funktionsaufrufe für den Zugriff auf EJBs deutlich reduzieren.

4.7 Systemkonfiguration

Obwohl die Konfigurationsinformation (vgl. Abschnitt 3.2) für das System bedeutend ist, wird die Schnittstelle zur verwendeten Informationsquelle für diese Daten von einer einfachen Singleton-Klasse realisiert [2]. Zur Zeit stützt sich die Implementierung auf ein wiederverwendetes Modul aus dem WebApp Framework [19]. Es erlaubt das einfache Einlesen von menschenlesbaren Konfigurationsdateien. Leider ist dieses

Modul nicht uneingeschränkt frei verwendbar, so dass ein Austausch durch einen XML-Parser wünschenswert wäre. Durch die klar definierte Schnittstelle zum System dürfte diese Änderung mit geringem Aufwand durchführbar sein.

Ein kurzer Ausschnitt aus der Konfigurationsdatei soll beispielhaft verdeutlichen, wie Navigations-Links als ein konkreter Aspekt der Konfiguration festgelegt werden:

```

navigation = {
  links = (
    {
      caption      = "Public Deliverables"
      href         = "/document/publicdocmain.jsp"
      target       = "main"
      requiredModules = ( "mod_document_management" )
      isImage      = false
      isPublic     = true
    } ...
  )
}

```

Die Konfigurationsdatei enthält einen Abschnitt „navigation“, in dem alle notwendigen Angaben zum Aufbau des Navigationsbereichs spezifiziert werden. Im Unterabschnitt „links“ wird eine Liste von Link-Beschreibungen angegeben. Deren Reihenfolge in der Liste bestimmt gleichzeitig die Reihenfolge im Navigationsmenü. Tabelle 1 erläutert die Bedeutung der einzelnen Angaben.

Schlüsselwort	Bedeutung des Wertes
isPublic	gibt an, ob der Link auch im öffentlichen Bereich sichtbar ist
isImage	gibt an, ob eine Grafik oder ein Textlink angezeigt werden soll
caption	Für Text-Links der anzuzeigende Text, für Grafik-Links die URL der anzuzeigenden Grafik
href	Ziel-URL des Links (HREF-Attribut des Links)
target	Ziel-Frame für den Inhalt des anzuzeigenden Dokuments.
requiredModules	Liste von Bezeichnern eines Moduls, die alle aktiv sein müssen, damit dieser Link angezeigt wird.

Tabelle 1: Konfigurierbare Angaben für einen Navigationslink

4.8 Implementierte Funktionsmodule

Im Rahmen des Komplexpraktikums wurde eine Untermenge aller Funktionen implementiert, die für die vollständige Funktionalität des Systems gefordert sind. Die aus

dieser Auswahl von Funktionen resultierenden Funktionsmodule bilden zum überwiegenden Teil die zwingend notwendige Infrastruktur für Verwaltungsfunktionen. Lediglich das *Document Management* (Abbildung 6) bietet „echte“ Funktionalität und könnte dadurch in der Systemkonfiguration abgeschaltet werden (vgl. Abschnitt 3.2, 4.7). Bei den übrigen Modulen ist dies per Festlegung verboten.

Das Modul *User Management* verwaltet die im System registrierten Nutzer, insbesondere deren Login-Name und Passwort sowie die ihnen zugeordneten Rollen. Diesen Rollen können dann konkrete Rechte zugeordnet werden, wofür das *Role Management* verantwortlich ist. Das Modul *Project Structure Management* bietet Funktionen zur Verwaltung und baumartigen Strukturierung von Projektstruktureinheiten, denen registrierte Nutzer als Mitarbeiter zugeordnet werden können. Das Modul *Document Management* schließlich baut auf all diesen Modulen auf und realisiert, wie durch den Namen angedeutet, eine Dokumentenverwaltung mit Suchfunktion.

4.9 Anwendungsbeispiel

Die in Abschnitt 4.8 erwähnte Suchfunktion des Dokumentenverwaltungsmoduls soll nun kurz erläutert werden.

Abbildung 9 zeigt im rechten Teil des Anzeigebereichs die Benutzungsoberfläche dieser Funktion (vgl. Abschnitt 4.5).

Die modulspezifischen Schaltflächen im oberen Teil des Hauptbereichs erlauben den schnellen Zugriff auf Funktionen zum Anzeigen, Suchen und Herunterladen von Dokumenten. Daneben ermöglicht ein Eingabefeld rechts die schnelle Durchführung von einfachen Suchanfragen. Diese Elemente sind in allen Teilen der Benutzungsschnittstelle für die Dokumentenverwaltung verfügbar. Komplexere Suchanfragen nach Dokumenten lassen sich mit der gesondert aufzurufenden Suchfunktion durchführen.

Die hierfür zur Verfügung stehende Eingabemaske ist horizontal in zwei Bereiche unterteilt. Im oberen Teil können durch Angabe von Suchbegriffen für jedes Attribut eines Dokuments einschränkende Bedingungen angegeben werden. Soll keine Einschränkung bezüglich eines bestimmten Attributs vorgenommen werden, ist das entsprechende Feld frei zu lassen. In Abbildung 9 wird beispielsweise nach einem Dokument mit dem Wort „Pflichtenheft“ in der Kurzbeschreibung gesucht.

Im unteren Teil wird durch Markieren von Auswahlfeldern die Menge der Attribute bestimmt, die in der Suchtrefferliste für jedes gefundene Dokument anzuzeigen sind. Als Standard ist das Auswahlfeld „all attributes“ markiert. Dadurch werden alle Attribute eines Dokuments in der Trefferliste angezeigt. Durch Betätigen der Schaltfläche „Search“ rechts unten wird die Suche gestartet.

5. Aktueller Stand und Ausblick

Im Rahmen des Komplexpraktikums wurde eine lauffähige Implementierung des ProjectWebs erstellt und beispielhaft für das Komplexpraktikum selbst sowie für das AQUILA-Projekt instanziiert.

Die Implementierung umfasst dabei grundlegende Anwendungsfälle, wie z.B. die Verwaltung von Projektrollen und -mitarbeitern, die Dokumentenverwaltung, die Suche in der Dokumentenverwaltung sowie den Abruf privater und öffentlicher Projektinformationen. Die in Abschnitt 3 beschriebenen Anwendungsfälle konnten auf Grund des zeitlichen Rahmens des Komplexpraktikums jedoch nicht vollständig umgesetzt werden. Dies betrifft z.B. Anwendungsfälle wie die Suche im gesamten Arbeitsbereich, die Snapshot-Funktion sowie die Verwaltung von E-Mail-Verteilerlisten. Da der Architektur des ProjectWeb ein modularer Aufbau zugrunde liegt und neue Dienste in Form von Funktionsmodulen über einen festgelegten Mechanismus in das ProjectWeb integriert werden können, ist die nachträgliche Realisierung dieser Anwendungsfälle jedoch unproblematisch.

Die Mitarbeiter des AQUILA-Projekts beabsichtigen eine Weiterführung und Vervollständigung des ProjectWeb in Rahmen von studentischen Arbeiten. Ein späterer praktischer Einsatz ist im AQUILA-Projekt sowie in ähnlich angelegten Projekten geplant.

6. Referenzen

- [1] BSCW-Server, <http://bscw.gmd.de/>
- [2] E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
- [3] E. Roman: Mastering Enterprise JavaBeans and the Java 2 Platform, Enterprise Edition; Wiley Computer Publishing, 1999
- [4] OMG: OMG Unified Modeling Language Specification (Draft), Version 1.4, Dokumentennummer: ad/01-02-13. 2001
- [5] K. Avedal et al.: Professional JSP, Wrox Press, 2000
- [6] AQUILA, <http://www-st.inf.tu-dresden.de/AQUILA>, Juli 2001
- [7] Komplexpraktikum, <http://www-st.inf.tu-dresden.de/kp/>
- [8] F. Buschmann, R. Meunier, H.Rohnert, P. Sommerlad, M. Stal: A system of patterns, John Wiley & Sons Ltd, 1996
- [9] Sun Microsystems: What is the Java Platform?, <http://java.sun.com/nav/whatis/>
- [10] Bill Shannon (Sun Microsystems): Java 2 Platform Enterprise Edition Specification, v1.2; Final Release, 1999, <http://java.sun.com/j2ee/>

- [11] V. Matena, M. Hapner (Sun Microsystems): Enterprise JavaBeans Specification, v1.1; Final Release, 1999, <http://java.sun.com/products/ejb/>
- [12] J. D. Davidson, D. Coward (Sun Microsystems): Java Servlet Specification, v2.2; Final Release, 1999, <http://java.sun.com/products/servlet/>
- [13] E. Pelegrí-Llopart, L. Cable (Sun Microsystems): JavaServer Pages Specification, v1.1; Final Release, 1999, <http://java.sun.com/products/jsp/>
- [14] World Wide Web Consortium (W3C): HTML 4.01 Specification, 1999, <http://www.w3.org/TR/html401/>
- [15] World Wide Web Consortium (W3C): Cascading Style Sheets, level 1, 1999, <http://www.w3.org/TR/REC-CSS1>
- [16] Netscape Corporation: SSL 3.0 Specification, 1996, <http://home.netscape.com/eng/ssl3/>
- [17] T. Dierks, C. Allen: RFC 2246: The TLS Protocol, Version 1.0, 1999, <ftp://ftp.isi.edu/in-notes/rfc2246.txt>
- [18] Java 2 Platform, Enterprise Edition Blueprints: J2EE Design Patterns, http://java.sun.com/j2ee/blueprints/design_patterns/catalog.html
- [19] The WebApp Framework, <http://www.webapp.de/>
- [20] G. E. Krasner, S. T Pope: A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80, Journal of Object-Oriented Programming 1(3), S. 26-49, August/September 1988
- [21] V. Turau, R. Pfeiffer: Java Server Pages, dpunkt, 2000, S. 129